



The **lwarp** package

L^AT_EX to HTML₅

v0.34 — 2017/08/08

© 2016–2017 Brian Dunn
bd@BDTechConcepts.com

Abstract

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

Documents may be produced by pdfL^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 a “beta” version of **lwarp, and some things may change in response to user feedback and further project development.**

License:

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

Contents

1	lwarp.sty	22
1	Updates	22
2	Introduction	26
2.1	Supported packages and features	27
3	Alternatives	32
3.1	Internet class	32
3.2	TeX4ht	32
3.3	Translators	32
3.4	AsciiDoc	33
3.5	Pandoc	33
3.6	Word processors	33
3.7	Commercial systems	34
3.8	Comparisons	34
4	Installation	35
4.1	Installing the lwarp package	35
4.2	Installing the lwarpmk utility	38
4.2.1	Using a local copy of lwarpmk	39
4.3	Installing additional utilities	40
5	Tutorial	43
5.1	Starting a new project	43
5.2	Compiling the print version with lwarpmk	47
5.3	Compiling the HTML version with lwarpmk	48
5.4	Generating the SVG images	49
5.5	Using MathJax for math	50
5.6	Changing the CSS style	51
5.7	Customizing the HTML output	52
5.8	Using latexmk	57
5.9	Using XeLaTeX or LuaLaTeX	58
5.10	Using a glossary	59
5.11	Cleaning auxiliary files	60
5.12	Cleaning auxiliary and output files	60
5.13	Processing multiple projects in the same directory	60
5.14	Using the make utility	60
6	Additional details	61
6.1	Font and UTF-8 support	61
6.1.1	Indexes and UTF-8	62
6.2	lwarp package loading and options	62
6.3	Selecting the operating system	64

6.4	Selecting actions for print or HTML output	64
6.5	Commands to be placed into the <code>warpprint</code> environment	65
6.6	Title page	65
6.7	HTML page meta descriptions	66
6.8	HTML page meta author	67
6.9	CSS	67
6.10	Modifying <code>xindy</code> index processing	68
6.11	Special cases and limitations	68
6.11.1	Page references	69
6.11.2	Text formatting	69
6.11.3	Horizontal space	69
6.11.4	Text alignment	69
6.11.5	Superscripts and other non-math uses of math mode	69
6.11.6	Empty <code>\item</code> followed by a new line of text or a nested list:	69
6.11.7	Filenames in lists	69
6.11.8	Cross-references	70
6.11.9	<code>cleveref</code> and <code>varioref</code> packages	70
6.11.10	Hyperlinks, <code>hyperref</code> , and <code>url</code>	70
6.11.11	Footnotes and page notes	70
6.11.12	Math	70
6.11.13	<code>ntheorem</code> package	72
6.11.14	Graphics	72
6.11.15	<code>xcolor</code> package	73
6.11.16	<code>Tabular</code>	73
6.11.17	<code>longtable</code> package	75
6.11.18	Save Boxes	75
6.11.19	Minipages	75
6.11.20	Side-by-side minipages	76
6.11.21	Framed minipages and other environments	76
6.11.22	<code>fancybox</code> package	77
6.11.23	<code>mdframed</code> package	79
6.11.24	<code>float</code> , <code>trivfloat</code> , and/or <code>algorithmicx</code> together	79
6.11.25	<code>caption</code> and <code>subcaption</code> packages	79
6.11.26	<code>subfig</code> package	79
6.11.27	<code>floatrow</code> package	80
6.11.28	<code>abstract</code> package	80
6.11.29	<code>verse</code> and <code>memoir</code>	80
6.11.30	<code>siunitx</code> package	81
6.11.31	<code>units</code> and <code>nicefrac</code> packages	81
6.11.32	<code>newclude</code> package	81
6.11.33	<code>newtxmath</code> package	82
6.11.34	<code>babel</code> package	82
6.11.35	<code>glossaries</code> package	82
7	EPUB conversion	84

8	Word-processor conversion	86
9	Modifying lwarp	87
9.1	Creating an lwarp version of a package	87
9.2	Testing lwarp	88
9.3	Modifying lwarpmk	88
10	Troubleshooting	89
10.1	Using the lwarp.sty package	89
10.1.1	Debug tracing output	91
10.2	Compiling the lwarp.dtx file	91
11	Implementation	92
12	Stack depths	92
13	Source Code	94
14	Detecting the T_EX Engine — pdf_latex, lua_latex, xel_latex	95
15	Unicode Input Characters	95
16	Early package requirements	96
17	Operating-System portability	97
17.1	Common portability code	97
17.2	Unix, Linux, and Mac OS	97
17.3	MS-Windows	97
18	Package options	98
19	Misplaced packages	101
20	Required packages	105
21	Loading packages	109
22	Copying a file	112
23	Debugging messages	112
24	HTML-conversion output modifications	113
25	Remembering original formatting macros	114
26	Configuration Files	116
26.1	project_html.tex	116
26.2	lwarpmk.conf	116
26.3	project.lwarpmkconf	117

26.4	lwarp.css	118
26.5	lwarp_sagebrush.css	143
26.6	lwarp_formal.css	148
26.7	sample_project.css	152
26.8	lwarp.xdy	152
26.9	lwarp_mathjax.txt	153
26.10	lwarpmk option	155
27	Stacks	166
27.1	Assigning depths	166
27.2	Closing actions	167
27.3	Closing depths	167
27.4	Pushing and popping the stack	168
28	Data arrays	169
29	HTML entities	170
30	HTML filename generation	171
31	Homepage link	173
32	\PrintStack diagnostic tool	174
33	Closing stack levels	175
34	PDF pages and styles	175
35	HTML tags, spans, divs, elements	176
35.1	Mapping L ^A T _E X Sections to HTML Sections	176
35.2	HTML tags	176
35.3	Block tags and comments	179
35.4	Div class and element class	180
35.5	Single-line elements	181
35.6	HTML5 semantic elements	182
35.7	High-level block and inline classes	182
35.8	Closing HTML tags	183
36	Paragraph handling	184
37	Paragraph start/stop handling	188
38	Page headers and footers	190
39	CSS	191
40	HTML meta description and author	192

41	Footnotes	193
41.1	Regular page footnotes	193
41.2	Minipage footnotes	193
41.3	Titlepage thanks	193
41.4	Regular page footnote implementation	193
42	Marginpars	195
43	Splitting HTML files	195
44	Sectioning	200
44.1	Book class commands	201
44.2	Sectioning support macros	201
44.3	\section and friends	206
45	Starting a new file	208
46	Starting HTML output	211
47	Ending HTML output	214
48	Titles and the titling package	215
48.1	Setting the title, etc.	217
48.2	Changes to HTML titlepage and titlingpage	219
48.3	Printing the title, etc. in HTML	220
48.4	Printing the title, etc. in print form	223
48.5	\maketitle for print output	224
48.6	\maketitle for HTML output	226
49	Abstract	229
50	Quote and verse	230
50.1	Citations and attributions	230
50.2	Quotes, quotations	230
50.3	Verse	231
51	Verbatim	232
52	Theorems	234
53	Lists	235
53.1	Itemize	235
53.2	Enumerate	236
53.3	Description	237
54	Tabular	238
54.1	Token lookahead	240
54.2	Booleans	240

54.3	Handling &, @, and !	241
54.3.1	Localizing & catcodes	242
54.3.2	Handling &	243
54.4	Handling \\	244
54.5	Variables	245
54.6	Parsing @, >, <, ! columns	246
54.7	Parsing 'l', 'c', or 'r' columns	249
54.8	Parsing 'p', 'm', or 'b' columns	249
54.9	Parsing 'D' columns	249
54.10	Parsing the column specifications	250
54.11	Starting a new row	254
54.12	Printing at or bang tags	254
54.13	Data opening tag	255
54.14	Midrules	256
54.15	Multicolumns	257
54.15.1	Parsing multicolumns	257
54.15.2	High-level multicolumn interface	260
54.15.3	Longtable captions	261
54.15.4	Counting HTML tabular columns	263
54.15.5	\tabledatamulticolumntag	265
54.16	Multirow	265
54.17	Utility macros inside a table	266
54.18	Checking for a new table cell	266
54.19	\mrowcell	268
54.20	New \tabular definition	268
54.21	Array	271
55	Cross-references	271
55.1	Setup	271
55.2	Zref setup	273
55.3	Labels	275
55.4	References	276
55.5	Hyper-references	278
56	Floats	281
56.1	Float captions	282
56.1.1	Caption inside a float environment	283
56.1.2	Caption and LOF linking and tracking	284
57	Table of Contents, LOF, LOT	287
57.1	Reading and printing the TOC	287
57.2	High-level TOC commands	290
57.3	Side TOC	290
57.4	Low-level TOC line formatting	291
58	Index and glossary	294

59	Math	296
59.1	Inline and display math	298
59.2	MathJax support	301
59.3	Equation environment	304
59.4	AMS Math environments	306
59.4.1	Support macros	306
59.4.2	Environment patches	307
60	Lateximages	314
61	center, flushleft, flushright	319
62	Siunitx	320
63	Graphics	321
63.1	<code>\graphicspath</code>	322
63.2	Length conversions and graphics options	322
63.3	<code>\includegraphics</code>	325
63.4	<code>\rotatebox</code> , <code>\scalebox</code> , <code>\reflectbox</code>	329
63.5	Null functions	331
64	Cleveref	332
65	Picture	333
66	Boxes and Minipages	334
66.1	Counters and lengths	334
66.2	Footnote handling	335
66.3	Minipage handling	335
66.4	Parbox, makebox, framebox, fbox, raisebox	338
67	Direct formatting	341
68	Skips, spaces, font sizes	346
69	<code>\phantomsection</code>	353
70	<code>\LaTeX</code> and other logos	353
70.1	HTML logos	353
70.2	Print logos	355
71	<code>\AtBeginDocument</code>, <code>\AtEndDocument</code>	356
72	Trademarks	357

2	lwarp-abstract.sty	358
73	Abstract	358
3	lwarp-afterpage.sty	360
74	Afterpage	360
4	lwarp-algorithmicx.sty	361
75	Algorithmicx	361
5	lwarp-alltt.sty	361
76	Alltt	362
6	lwarp-amsthm.sty	363
77	AMSthm	363
7	lwarp-balance.sty	366
78	Balance	366
8	lwarp-bookmark.sty	367
79	Bookmark	367
9	lwarp-booktabs.sty	367
80	Booktabs	367
10	lwarp-boxedminipage.sty	368
81	Boxedminipage	368
11	lwarp-boxedminipage2e.sty	368
82	Boxedminipage2e	368

12	lwarp-ccaption.sty	369
83	Ccaption	369
13	lwarp-changepage.sty	369
84	Changepage	369
14	lwarp-crop.sty	370
85	Crop	370
15	lwarp-cutwin.sty	371
86	Cutwin	371
16	lwarp-dcolumn.sty	371
87	Dcolumn	372
17	lwarp-draftwatermark.sty	373
88	Draftwatermark	373
18	lwarp-ellipsis.sty	373
89	Ellipsis	373
19	lwarp-emptypage.sty	374
90	Emptypage	374
20	lwarp-endnotes.sty	375
91	Endnotes	375
21	lwarp-enumerate.sty	376
92	Enumerate	376

22	lwarp-enumitem.sty	377
93	Enumitem	377
23	lwarp-epigraph.sty	378
94	Epigraph	378
24	lwarp-eso-pic.sty	379
95	Eso-pic	379
25	lwarp-everypage.sty	380
96	Everypage	380
26	lwarp-everyshi.sty	381
97	Everyshi	381
27	lwarp-extramarks.sty	381
98	Extramarks	381
28	lwarp-fancybox.sty	382
99	Fancybox	382
29	lwarp-fancyhdr.sty	385
100	Fancyhdr	385
30	lwarp-fancyvrb.sty	387
101	Fancyvrb	387
31	lwarp-figcaps.sty	393
102	Figcaps	393

32	lwarp-float.sty	394
103	Float and \newfloat	394
33	lwarp-floatflt.sty	396
104	Floatflt	396
34	lwarp-floatpag.sty	397
105	Floatpag	397
35	lwarp-floatrow.sty	398
106	Floatrow	398
36	lwarp-flushend.sty	404
107	Flushend	404
37	lwarp-fontenc.sty	405
108	Fontenc	405
38	lwarp-fontspec.sty	405
109	Fontspec	405
39	lwarp-footmisc.sty	406
110	Footmisc	406
40	lwarp-footnote.sty	408
111	Footnote	408
41	lwarp-footnotehyper.sty	409
112	Footnotehyper	409

42	lwarp-framed.sty	410
113	Framed	410
43	lwarp-ftnright.sty	413
114	Ftnright	413
44	lwarp-fullpage.sty	413
115	Fullpage	413
45	lwarp-geometry.sty	414
116	Geometry	414
46	lwarp-glossaries.sty	415
117	Glossaries	415
47	lwarp-graphics.sty	416
118	Graphics	416
48	lwarp-graphicx.sty	416
119	Graphicx	416
49	lwarp-hyperref.sty	417
120	Hyperref	417
50	lwarp-hyperxmp.sty	420
121	Hyperxmp	420
51	lwarp-idxlayout.sty	421
122	Idxlayout	421

52	lwarp-indentfirst.sty	422
123	Indentfirst	422
53	lwarp-inputenc.sty	422
124	Inputenc	422
54	lwarp-keyfloat.sty	423
125	Keyfloat	423
55	lwarp-layout.sty	425
126	Layout	425
56	lwarp-letterspace.sty	425
127	Letterspace	425
57	lwarp-lettrine.sty	426
128	Lettrine	426
58	lwarp-lips.sty	427
129	Lips	427
59	lwarp-listings.sty	428
130	Listings	428
60	lwarp-longtable.sty	432
131	Longtable	432
61	lwarp-lscape.sty	434
132	Lscape	434

62	lwarp-ltcaption.sty	434
133	Ltcaption	434
63	lwarp-marginfit.sty	435
134	Marginfit	435
64	lwarp-marginfix.sty	435
135	Marginfix	435
65	lwarp-marginnote.sty	436
136	Marginnote	436
66	lwarp-mcaption.sty	436
137	Mcaption	436
67	lwarp-mdframed.sty	437
138	Mdframed	437
68	lwarp-microtype.sty	446
139	Microtype	446
69	lwarp-moreverb.sty	447
140	Moreverb	447
70	lwarp-mparhack.sty	449
141	Mparhack	449
71	lwarp-multicol.sty	450
142	Multicol	450

72	lwarp-multirow.sty	452
143	Multirow	452
73	lwarp-nameref.sty	452
144	Nameref	452
74	lwarp-needspace.sty	453
145	Needspace	453
75	lwarp-newclude.sty	453
146	Newclude	453
76	lwarp-newunicodechar.sty	454
147	Newunicodechar	454
77	lwarp-nextpage.sty	454
148	Nextpage	454
78	lwarp-nowidow.sty	454
149	Nowidow	455
79	lwarp-ntheorem.sty	456
150	Ntheorem	456
80	lwarp-pagenote.sty	467
151	Pagenote	467
81	lwarp-paralist.sty	467
152	Paralist	467

82	lwarp-parskip.sty	468
153	Parskip	468
83	lwarp-pdfscape.sty	468
154	Pdfscape	468
84	lwarp-pdfsync.sty	469
155	Pdfsync	469
85	lwarp-placeins.sty	469
156	Placeins	469
86	lwarp-prelim2e.sty	470
157	Prelim2e	470
87	lwarp-ragged2e.sty	471
158	Ragged2e	471
88	lwarp-rotating.sty	472
159	Rotating	472
89	lwarp-rotfloat.sty	473
160	Rotfloat	473
90	lwarp-savetrees.sty	474
161	Savetrees	474
91	lwarp-setspace.sty	475
162	Setspace	475

92	lwarp-shadow.sty	476
163	Shadow	476
93	lwarp-showidx.sty	477
164	Showidx	477
94	lwarp-showkeys.sty	477
165	Showkeys	477
95	lwarp-sidecap.sty	478
166	Sidecap	478
96	lwarp-sidenotes.sty	479
167	Sidenotes	479
97	lwarp-soul.sty	481
168	Soul	481
98	lwarp-subfig.sty	483
169	Subfig	483
99	lwarp-syntonly.sty	489
170	Syntonly	489
100	lwarp-tabularx.sty	490
171	Tabularx	490
101	lwarp-tabulary.sty	491
172	Tabulary	491

102	lwarp-textpos.sty	492
173	Textpos	492
103	lwarp-theorem.sty	493
174	Theorem	493
104	lwarp-threeparttable.sty	497
175	Threeparttable	497
105	lwarp-tikz.sty	498
176	Tikz	498
106	lwarp-titleps.sty	500
177	Titleps	500
107	lwarp-titleref.sty	502
178	Titleref	502
108	lwarp-titlesec.sty	503
179	Titlesec	503
109	lwarp-titletoc.sty	504
180	Titletoc	504
110	lwarp-titling.sty	505
181	Titling	505
111	lwarp-tocloft.sty	506
182	Tocloft	506

112	lwarp-trivfloat.sty	511
183	Trivfloat	511
183.1	Combining <code>\newfloat</code> , <code>\trivfloat</code> , and <code>algorithmicx</code>	511
113	lwarp-ulem.sty	513
184	Ulem	513
114	lwarp-verse.sty	515
185	Verse	515
115	lwarp-wallpaper.sty	517
186	Wallpaper	517
116	lwarp-wrapfig.sty	518
187	Wrapfig	518
117	lwarp-xcolor.sty	520
188	Xcolor	520
118	lwarp-xfrac.sty	523
189	Xfrac	523
119	lwarp-xmpincl.sty	526
190	Xmpincl	526
	Change History and Index	527

List of Figures

1	tutorial.tex listing	44
---	--------------------------------	----

List of Tables

1	L ^A T _E X–HTML generation — lwarp package — Supported functions	27
2	Additional supported packages	31
3	Required software programs	36
4	Files created along with the print version	46
5	Package options	63
6	Section depths and HTML headings	93
7	Tabular baseline	250
8	Tabular HTML column conversions	251
9	Cross-referencing data structures	272
10	Float data structures	281

Package 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 [527](#).

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 L^AT_EX standards, and avoids problems with space between arguments.
- ⚠ **Optional arguments**
 - Likewise, `\InlineClass`'s optional argument now comes before the mandatory arguments:


```
\InlineClass[style]{class}{text}
```
- spans with minipages
 - Improved compatibility between spans, minipages, lists, frames, and math. Handles minipages and lists inside an HTML span, such as an `\fbox` containing a minipage, although with minimal HTML formatting. See section [6.11.19](#). `\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 [6.11.21](#). `\fbox` and minipages now often work in SVG math and `lateximages`. MathJax supports `\fbox`, but not `\fboxBlock` nor `fminipage`.
- framing minipages
 - Improved compatibility between `lateximage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, `booktabs`.
- lateximage, SVG math, tabular
 - Improved font control for `lateximages` and SVG math.
- eqnarray
 - Added the `eqnarray` environments.
- verbatim packages
 - `fancyvrb` is no longer required (preloaded), but is still supported.

framing packages
list packages

- Added `verbatim` and `moreverb`.
- Added `fancybox`, `boxedminipage2e` and `shadow`.
- `enumitem` is no longer required, but is still supported.
- Added `enumerate` and `paralist`.
- `titles` 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`.

babel-french

v0.33:

- Tabular `@` and `!` columns now have their own HTML columns.
- `&` catcode changes are localized, perhaps causing errors about the tab alignment character `&`, so any definitions of macros or environments which themselves contain `tabular` and `&` must be enclosed within `\StartDefiningTabulars` and `\EndDefiningTabulars`. See section 54.3.1. 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 HTML5, CSS3, and SVG are employed to allow a fairly feature-rich document without relying on the use of Javascript. Limited testing on older browsers show that these new features degrade gracefully, although the SVG format for math may not be available on small cell phones.

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

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

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

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

The `lwarp` package allows \LaTeX to directly generate HTML5 tags from a \LaTeX source document, with only minor intervention on the user’s part. A `texlua` program called `lwarpmk` is used to process either the print or HTML version of the document. A few external utility programs are used to finish the conversion from

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!

a L^AT_EX-generated PDF file which happens to have HTML5 tags, to a number of HTML5 plain-text files and accompanying images.

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

When requesting packages during HTML conversion, lwarp first looks to see if it has its own modified version to use instead of the usual L^AT_EX 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 27.)

Assistance is provide 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 L^AT_EX 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 L^AT_EX code.

Table 1: L^AT_EX–HTML generation — lwarp package — Supported functions

Category	Status
Engines:	pdfL ^A T _E X, X _Y L ^A T _E X, LuaL ^A T _E X
Classes:	<code>book</code> , <code>report</code> , or <code>article</code> . <code>memoir</code> is planned.

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> .
Page layout:	Emulates <code>fancyhdr</code> , <code>titleps</code> .
Table of contents, figures, tables:	Supported, with hyperlinks. Emulates <code>titletoc</code> and <code>tocloft</code> .
Title page:	<code>\maketitle</code> , <code>titlepage</code> , <code>titling</code> . Optional <code>titling</code> -based commands for published and subtitle.
<code>abstract</code> :	Supported
Cross-references:	Emulated, with hyperlinks.
<code>hyperref</code> :	Emulated. HTML hyperlinks are generated for TOC, LOF, LOT, <code>\nameref</code> , <code>\ref</code> , the <code>cleveref</code> commands, and index entries.
<code>url</code> :	Supported, without hyperlinks so far.
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> .
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, without hyperlinks so far.
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> .

lwarp Supported Functions — continued

Category	Status
Floats:	Appear where declared. <code>float</code> , <code>rotfloat</code> , <code>newfloat</code> , <code>caption</code> and <code>subcaption</code> , <code>subfig</code> , <code>capt-of</code> , <code>placeins</code> , <code>trivfloat</code> , <code>floatrow</code> , <code>keyfloat</code> , <code>wrapfig</code> , <code>cutwin</code> , <code>floatflt</code> .
<code>tabular</code> :	Emulated. <code>\multirow</code> and <code>\multicolumn</code> are available, but cannot be used at the same time. Nested tables are not supported.
<code>array</code> :	Supported inside math environments, emulated elsewhere.
<code>tabularx</code> , <code>tabulary</code> , <code>threeparttable</code> , <code>multirow</code> :	Emulated.
<code>longtable</code> :	Emulated. Converted to a <code>tabular</code> . Captions supported. Extra headings and <code>\kill</code> lines must be enclosed in <code>\warpprintonly{}</code>
<code>booktabs</code> :	Emulated. <code>\toprule</code> and <code>\bottomrule</code> form black rules, <code>\midrule</code> forms silver rules, as demonstrated on this table. <code>\cmidrule</code> , demonstrated at this line, does not use <code>width</code> or <code>trim</code> options.
<code>graphics</code> , <code>graphicx</code> :	Emulated. <code>\includegraphics</code> supports <code>width</code> , <code>height</code> , <code>origin</code> , <code>angle</code> , and <code>scale</code> tags, and adds <code>class</code> . References to PDF files are changed to SVG, other image types are accepted as well. <code>\rotatebox</code> and <code>\scalebox</code> are supported as well as HTML can handle.
<code>rotating</code> :	Emulated. All objects are displayed unrotated.
Lists:	Standard \LaTeX environments are supported, along with <code>enumitem</code> , <code>enumerate</code> , <code>paralist</code> . Spacing is still controlled by CSS.
Environments:	Standard \LaTeX environments are supported.
<code>picture</code> and <code>tikz</code> :	Converted to an SVG image.
<code>minipage</code> :	Supported with some HTML5-imposed limitations. Nested minipages are supported. Footnotes appear at the bottom of the HTML page.

lwarp Supported Functions — continued

Category	Status
Verbatim:	<code>verbatim</code> , <code>moreverb</code> , <code>fancyvrb</code> (except for <code>verbatim</code> footnotes).
Frames:	<code>framed</code> , <code>fancybox</code> , <code>mdframed</code> , <code>boxedminipage2e</code> , <code>shadow</code> .
<code>multicol</code> :	Emulated, with <code>CSS3</code> . Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported.
<code>siunitx</code> :	Supported except for <code>per-mode=fraction</code> .
<code>xfrac</code> :	Supported
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> , and <code>soul</code> are supported.
Ordinals:	<code>nth</code> , <code>fmtcount</code> , and <code>engord</code> are supported.
Text ligatures:	Ligatures for symbols are supported. Ligatures for <code>f</code> , <code>q</code> , <code>t</code> are intentionally turned off because many simpler browsers do not display them correctly. Modern full-featured browsers re-create these ligatures on-the-fly.
Horizontal space:	HTML output for <code>thin-unbreakable</code> , <code>unbreakable</code> , <code>\enskip</code> , <code>\quad</code> , <code>\qquad</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.
<code>xcolor</code> :	Supported . Full package color names, any color models, and <code>mixing</code> is converted to hex web colors via <code>\convertcolorspec</code> . Patched commands are <code>\textcolor</code> , <code>\colorbox</code> , and <code>\fcolorbox</code> . <code>\pagecolor</code> is not supported.
Where:	
Supported:	The existing \LaTeX package is used.

Table 2: Additional supported packages

babel, bm, calc, cleveref, csquotes, fancyvrb, fileerr, newtxmath, nicefrac siunitx, somedefs, tikz, trace, units, url, varioref, xspace

lwarp Supported Functions — continued

Category	Status
Emulated:	The \LaTeX package is not used, but some/all of its functions are emulated. Null functions, lengths, and counters are provided for source compatibility.

Supported packages include everything listed in the table of contents, plus the packages listed in table 2, and probably others which have not yet been tested. Many are simply nullfied during HTML output. Others are not affected by the output mode and thus work as-is.

These packages and features probably works with little or no change to the user's source code. Special environments are provided to mark blocks of code which are for print only, HTML only, or both, should it be necessary.

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 L^AT_EX 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 L^AT_EX 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 L^AT_EX-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 L^AT_EX 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 L^AT_EX-ish input syntax, unicode math fonts, and high-quality output, and also generate HTML with varying success.
Prog LibreOffice
Prog OpenOffice See recent developments in Microsoft® Word® and LibreOffice™ Writer.

3.7 Commercial systems

Prog Adobe Likewise, several professional systems exist whose abilities have been advancing in the areas of typesetting, cross-referencing, and HTML generation. See Adobe® FrameMaker®, Adobe® InDesign®, and Madcap Flare™.

Prog FrameMaker

Prog InDesign

Prog Flare

Prog Madcap

3.8 Comparisons

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

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

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

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

4 Installation

Table 3 shows the tools which are used for the L^AT_EX 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.

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

```
Enter ⇒ tlmgr install lwarp
```

MiK_TE_X: If using MiK_TE_X, 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 T_EX-related packages.

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

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

T_EX Live:

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

MiK_TE_X:

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

This should be something like:

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

4. Unpack the archive in the TDS local directory.

Table 3: Required software programs

Provided by your L^AT_EX distribution:

From T_EXLive: <http://tug.org/texlive/>.

L^AT_EX: pdf_latex, xelatex, or lualatex.

The lwarp package: This package.

The lwarpmk utility: Provided along with this package. This should be an operating-system executable in the same way that pdf_latex or latexmk is. It is possible to have the lwarp package generate a local copy of lwarpmk called lwarpmk.lua. See table 4.

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_TE_X 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 L^AT_EX code. On a MiK_TE_X system, Perl may need to be installed first.

pdfcrop: Used to pull images out of the L^AT_EX PDF.

Poppler PDF utilities:

pdftotext: Used to convert PDF to text.

pdfseparate: Used to pull images out of the L^AT_EX 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

5. Renew the cache:

Enter ⇒ `mktexlsr`

— or —

Enter ⇒ `texhash`

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

CTAN .dtx and .ins files: Another form of TeX 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 TeX Live local tree found in the previous CTAN TDS section, under the subdirectory:

`<texlocal>/tex/latex/local/lwarp`
8. Copy the documentation `lwarp.pdf` to a source directory in the local tree, such as:

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

Enter ⇒ `mktexlsr`

— or —

Enter ⇒ `texhash`

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

Just testing!

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.

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 lwrapmk
```

Windows T_EX Live: Create a new `lwrapmk.exe` file:

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

Windows MiK_TE_X: Create a new `lwrapmk.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 `lwrapmk.bat` containing:

```
texlua "C:\Program Files\MiKTeX 2.9\scripts\lwrap\lwrap.texlua" %*
```

This will call the copy of `lwrapmk.lua` which is in the `scripts` directory of the distribution.

4.2.1 Using a local copy of `lwrapmk`

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

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

```
\usepackage[lwrapmk]{lwrap}
```

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

Unix:

- (a) Make `lwrapmk.lua` executable:

```
Enter ⇒ chmod 0755 lwrapmk.lua
```

- (b) Compile documents with

```
Enter ⇒ ./lwrapmk.lua html
```

```
Enter ⇒ ./lwrapmk.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 3 on page 36.

```
Enter ⇒ luatex -version
```

```
Enter ⇒ xindy -version
```

```
Enter ⇒ latexmk -version
```

```
Enter ⇒ perl -version
```

```
Enter ⇒ pdfcrop -version
```

```
Enter ⇒ pdftotext -v
```

```
Enter ⇒ pdfseparate -version
```

```
Enter ⇒ pdftocairo -v
```

To install xindy, latexmk, and pdfcrop:

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

<http://ctan.org/pkg/xindy>
<http://ctan.org/pkg/latexmk>
<http://ctan.org/pkg/pdftocrop>

Prog pdftotext **To install the Poppler utilities to a Unix/Linux system:**
 Prog pdfseparate The tools from the POPPLER project should be provided by your operating
 Prog pdftocairo 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 3 on page 36.
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 4. 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, L^AT_EX program (`pdflatex`, `xelatex`, or `lualatex`), the filenames of the source code and HTML output, and whether the additional helper program `latexmk` will be used to compile the document.)

Figure 1: tutorial.tex listing

Note: There are two pages!

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

\documentclass{book}

\usepackage{iftex}

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

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

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

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

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

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

\begin{document}

\maketitle                % Or titlepage/titlingpage environment.

% An article abstract would go here.

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

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

\section{Some math}

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

\printindex

\end{document}
```

Table 4: Files created along with the print version

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

When the lwarpmk option is given to the lwarp package:

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

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

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

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

5.2 Compiling the print version with lwarpmk

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

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

1. Re-compile the print version:

Enter \Rightarrow `lwarpmk print`

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

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

Enter \Rightarrow `lwarpmk print`

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

4. Force a recompile to occur.

Enter \Rightarrow `lwarpmk again`

Enter \Rightarrow `lwarpmk print`

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

5. Process the index.⁴

Enter \Rightarrow `lwarpmk printindex`

6. Recompile again to include the index.

Enter \Rightarrow `lwarpmk print`

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 [6.11.35](#).

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`⁵, `Some-math.html`, and the document's index in `_Index.html`.⁶

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:⁷

```
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.

⁵`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`.

⁶`index.html` is commonly used as a homepage, so the document index is in `_Index.html`.

⁷A `lwarpmk htmlglossary` command is also available to process a glossary produced with the `glossaries` package. See section 6.11.35.

5.4 Generating the SVG images

math as SVG images By default `lwrap` 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 `lwrap` uses the same mechanism for `picture` and `Tikz` environments.

1. Create the SVG images:

```
Enter ⇒ lwrapmk limages
```

```
Enter ⇒ lwrapmk 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 `lwrapmk limages` to maintain the proper image file sequence numbers.

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

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

5.5 Using MathJax for math

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

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

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

2. Recompile

```
Enter ⇒ lwarpmk html
```

3. Reload the math page.

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

While using MathJax has many advantages, it may not be able to represent complex expressions or spacing adjustments as well as \LaTeX .

5.6 Changing the CSS style

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

The styles provided by `lwarp` include:

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

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

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

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

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

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

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

5.7 Customizing the HTML output

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

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

1. Clear all the HTML, PDF, and auxiliary files:
Enter \Rightarrow `lwarpmk clearall`
2. Recompile the print version in order to recreate the configuration files for `lwarpmk`:
Enter \Rightarrow `lwarpmk print`
3. Finally, recompile the HTML version with the new settings:
Enter \Rightarrow `lwarpmk html`

Options for the `lwarp` package:

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

Opt	<code>HomeHTMLFilename</code>	<p>HomeHTMLFilename: Filename of the homepage, without the “.html” suffix. Defaults to the <code>\BaseJobname</code>. A common setting is:</p> <pre>HomeHTMLFilename=index</pre> <p>causing the homepage to be the file <code>index.html</code>. Underscores are allowed in <code>HomeHTMLFilename</code> and <code>HTMLFilename</code> options, but may need to be escaped elsewhere, such as when appearing in a list:</p> <pre>\item [\href{file_name.pdf}{text}] \</pre>
	filename underscores	
Opt	<code>HTMLFilename</code>	<p>HTMLFilename: A filename prefix for the rest of the HTML web pages. Useful for numbered web pages with a common prefix. May be empty.</p>
Opt	<code>latexmk</code>	<p>latexmk: Controls whether <code>lwarp</code> uses <code>latexmk</code> to compile the document. This setting is written to <code>lwarpmk</code>'s configuration files. Defaults to false.</p>
Opt	<code>mathsvg</code>	<p>mathsvg: Selects SVG display for math output. (The default.)</p>
Opt	<code>mathjax</code>	<p>mathjax: Selects MathJax for math output.</p>

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

Ctr	<code>tocdepth</code>	<p>tocdepth: Sectioning depth of the table of contents. See section 12 for a list of L^AT_EX stack depths.</p>
-----	-----------------------	---

Ctrl **SideTOCDepth**

[sideTOC](#)

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

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



Ctrl **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`. (See `\SetHTMLFileNumber` to number grouped by chapter, for example.)

```
\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.

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

<code>\HTMLLanguage</code>	\HTMLLanguage: The HTML file's <code>html lang</code> tag. Defaults to <code>en-US</code> .
<code>\HTMLAuthor</code>	\HTMLAuthor: The HTML header's meta author. Defaults to <code>\theauthor</code> .
<code>\HTMLDescription</code>	\HTMLDescription: <code>{\description}</code> Sets the HTML <code>description</code> tag for the following files. May be changed before each each sectioning command which would cause a file split.
<code>\HTMLFirstPageTop</code>	\HTMLFirstPageTop: <code>{\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>	\HTMLPageTop: <code>{\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>	\HTMLPageBottom: <code>{\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:

<code>\tableofcontents</code> ⚠ TOC on the homepage!	\tableofcontents: Used to place a table of contents on the home page. This command must be used before the first file split, so that a way is available to navigate to other files from the homepage. Links to each chapter/section are provided, as selected by <code>tocdepth</code> .
--	--

Placed in the document wherever necessary:

Env <code>warpprint</code>	warpprint: An environment which is only used while generating print output. Place here anything which does not apply to HTML and which may cause problems with <code>lwarp</code> . If <code>lwarp</code> knows about and emulates or supports a package then its related macros, lengths, counters, etc. probably
----------------------------	---

won't have to be placed inside a `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: $\{\langle contents \rangle\}$ A macro version of the <code>warpprint</code> environment.
<code>\warpHTMLonly</code>	\warpHTMLonly: $\{\langle contents \rangle\}$ A macro version of the <code>warpHTML</code> environment.

5.8 Using latexmk

`latexmk` is a \LaTeX 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.

5.9 Using XeLaTeX or LuaLaTeX

X_YL^AT_EX or LuaL^AT_EX may be used instead of L^AT_EX.

1. Remove the auxiliary files for the project:

```
Enter ⇒ lwrapmk 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 `lwrapmk` are modified to remember which T_EX engine was used. X_YL^AT_EX or LuaL^AT_EX will be used for future runs of `lwrapmk`.

3. To recompile the document:

```
Enter ⇒ lwrapmk print
```

-and-

```
Enter ⇒ lwrapmk 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.



X_YLaTeX and LuaLaTeX users must use the `fontspec` package. Do NOT use `fontenc`!

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

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

- For X_YLaTeX or LuaLaTeX:

- `fontspec` and font choices

Pkg `fontspec`

`ligatures`

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

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

- For `pdflatex`:

- `lmodern` or other font-related packages

Pkg `lmodern`

- `fontenc`

Pkg `fontenc`

- `inputenc`

Pkg `inputenc`

- `newunicodechar`

Pkg `newunicodechar`

- `\input glyphtounicode.tex`

File `glyphtounicode`

- `\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 process indexes.

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

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

For an older distribution of L^AT_EX, it may be necessary to generate a local version of `lwarpmk.lua` and modify it to remove the `-C utf8` option from the `texindy` call. See section 9.3.

6.2 lwarp package loading and options

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

Pkg	lwarp	Load the lwarp package immediately after the font and UTF-8 setup commands.
Opt	warpprint	Select the warpprint option to generate print output (default), or the warpHTML option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual pdflatex, etc. When lwarp is loaded in print mode, it creates <code><project>_html.tex</code> , which sets the warpHTML 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 mathsvg (default), or mathjax. For more information about the math options, see section 6.11.12.
Opt	mathjax	

See table 5 for the full list of options.

Table 5: 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> .

6.3 Selecting the operating system

Prog	Unix	<code>lwarp</code> 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 <code>lwarp</code> option <code>OSWindows</code> .
Prog	Windows	When detected or specified, the operating-system path separator used by <code>lwarp</code> is modified, the boolean <code>usingOSWindows</code> is set true. This boolean may be tested by the user for later use.
Opt	OSWindows	

6.4 Selecting actions for print or HTML output

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

For most of built-in L^AT_EX 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	<code>warpHTML</code>	Anything which is to be done only for HTML5 output is surrounded by a <code>warpHTML</code> environment:
-----	-----------------------	--

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

Env	<code>warpprint</code>	Anything which is to be done only for print output is surrounded by a <code>warpprint</code> environment:
-----	------------------------	---

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

Env	<code>warpall</code>	Anything which is to be done for any output may be surrounded by a <code>warpall</code> environment. Doing so is optional.
-----	----------------------	--

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

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

```
\warpprintonly {<actions>}
```

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

```
\warpHTMLonly {<actions>}
```

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 10: [Troubleshooting](#).

6.6 Title page

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

```
\title{Document Title} % One line only
\subtitle{Optional Document Subtitle \\ with optional multiple lines}
\author{Author One\affiliation{Affiliation One} \and
Author Two\affiliation{Affiliation Two} }
\date{Optional date}
\published{Optional Journal Name \\ Optional multiple lines}
```

The title is used in the meta tags in the HTML files, and the rest are used in `\maketitle`.

- `\maketitle` Use `\maketitle` just after the `\begin{document}`, as this will establish the title of the homepage. Optionally, use a `titlepage` environment instead.
- Env `titlepage` The `titlepage` environment may be used to hold a custom title page. The titlepage will be set in a `<div>` class `titlepage`, and `\printtitle`, etc. may be used inside this environment.
- Env `titlingpage` Another form of custom title page, where `\maketitle` is allowed, and additional information may be included as well.
- `\title` `{\langle title \rangle}`
- ⚠ Avoid newlines in the `\title`; these will interfere with the file break and CSS detection. Use the `\subtitle` command instead. 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 `datedate`.
- `\subtitle` `{\langle subtitle \rangle}`
- A new command which sets a subtitle. Newlines are allowed. The default is empty. In HTML, this will appear in a `<div>` class `subtitle`.
- `\published` `{\langle published \rangle}`
- A new command which sets a publisher. The default is empty. In HTML, this will appear in a `<div>` class `published`.
- `\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 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` `{\author}` Sets the contents of the web page `<meta name="author">` element. Defaults to `\HTMLAuthor{\theauthor}`. May be set empty to cancel the meta author tag.

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 File `lwarp.xdy` `lwarp` whenever a print version of the document is processed.

To customize index processing:

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

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

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

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

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

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

6.11 Special cases and limitations

Also see section 10: [Troubleshooting](#).

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

6.11.1 Page references

The printed page does not translate to the HTML page, so references to page numbers are converted to parentheses containing `\pagerefPageFor`, which defaults to “see”, followed by a hyperlink to the appropriate object. Ex: “Sec. 1.23 on page (see sec. 1.23)”. `\pagerefPageFor` may be redefined to “page for”, empty, etc.

6.11.2 Text formatting

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

6.11.3 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`

6.11.4 Text alignment

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

6.11.5 Superscripts and other non-math uses of math mode

Use `x` instead of `x`

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

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

6.11.7 Filenames in lists

[filename underscore](#) Escape underscores in the filenames:
`\item[\href{file_name.pdf}{text}]`

6.11.8 Cross-references

`\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.

6.11.9 `cleveref` and `varioref` packages

`cleveref` and `varioref` are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used instead. See section 6.11 to redefine the message which is printed for page number references.

6.11.10 Hyperlinks, `hyperref`, and `url`

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

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

6.11.11 Footnotes and page notes

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

6.11.12 Math

 **Equation numbering** `ntheorem` has a bug with equation numbering in \mathcal{AMS} environments when the option `thref` is used. `lwrap` does not share this bug, so equations with `\split`, etc, are numbered correctly with `lwrap`'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 font⁸, 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

⁸See section 189 regarding fonts and fractions.

math may be typeset by L^AT_EX 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 L^AT_EX code which generated the math, allowing copy/paste of the L^AT_EX 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.

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 7 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 L^AT_EX PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of L^AT_EX, and
2. As detokenized printed L^AT_EX 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 L^AT_EX values, and the MathJax cross-referencing system is ignored in favor of the L^AT_EX internal system, seamlessly integrating with the rest of the L^AT_EX 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 L^AT_EX. `\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.

 `siunitx` inside an equation

<https://github.com/mathjax/MathJax-third-party-extensions/tree/master/siunitx>

\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 298.

6.11.13 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.

6.11.14 Graphics

 `graphics` vs. `graphicx`

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

 viewports

 `\graphicspath`

`\graphicspath` only works for a single directory; all graphics must be in this directory.

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`. Using 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.

 **image file types** For `\includegraphics` the user should provide both `.pdf` and `.svg` images, but always refer to `.pdf` images in the document source. All `\includegraphics` references to `.pdf` will automatically be changed to `.svg` for HTML output, and will be left as `.pdf` for print output. Images may also be `.jpg` and `.png`, and will be used as-is for either output.

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

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

6.11.15 `xcolor` package

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.

6.11.16 `Tabular`

 **misplaced alignment tab character &**

- When defining environments or macros which include `tabular` plus 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:

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

column types

- 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.
- \newcolumntype is ignored; unknown column types are set to 1.
- tabularx ignores the width, but X columns do produce paragraph columns or multicolumns.

 \multirow &
\multicolumn

- Multirow and multicolumn cannot be used at the same time. (No rectangular holes wider than one column or taller than one row.)

 \multirow

- For multirow, insert \mrowcell into any empty multi-row cells. This will be a null function for the print output, and is a placeholder for parsing the table for HTML output.

\multirow 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.

 paragraphs

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

\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.

longtable headings

- For longtable, place headings and footings which do not apply to HTML inside \warpprintonly{}

 \warpprintonly

- 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.

 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 \\}
```

6.11.17 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`.

6.11.18 Save Boxes

L^AT_EX 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.

6.11.19 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 L^AT_EX PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

6.11.20 Side-by-side minipages

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

6.11.21 Framed minipages and other environments

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

- `\fbox` For a framed object, options include:
`\fboxBlock`
 Env `fminipage`
- To remove the frame in HTML output:** Place the `\fbox` command and its closing brace inside `warpprint` environments. This will nullify the frame for HTML output.
- To frame the contents inline with some formatting losses in HTML:**
 For inline text: This is the default action of `\fbox` when enclosing a minipage. During HTML output, `\fbox` nullifies the HTML tags for `minipage`, `\parbox`, and lists. The contents are included as inline text inside the `\fbox`’s `` of class `framebox`. For lists, line breaks are converted to HTML breaks. The result is a plain-text inline version of the contents, framed inline with the surrounding text, but lacking any extra HTML markup.

To frame the contents on their own line with improved formatting inFor inline `minipage` and lists:

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

To create a framed minipage in both print and HTML:For display `tabular`, `minipages`, and lists:

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

To frame tables or verbatim environments:

△ Misplaced alignment tab character &

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 the `lwarp` documentation 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}

```

6.11.22 fancybox package

[framed equation example](#) `fancybox`'s documentation has an example `FramedEqn` environment which com-

bines `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`.

6.11.23 **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.

6.11.24 **float, trivfloat, and/or algorithmicx** together

 **package conflicts** If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 183.1.

6.11.25 **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`.

6.11.26 **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.

6.11.27 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.

6.11.28 abstract package

`abstract` is supported. 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.

6.11.29 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 `\HTMLleftmargin` 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 `\HTMLleftmargin` is if there is a wide `\flagverse` in use, such as the word “Chorus”, in which case the value of `\HTMLleftmargin` 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 `\HTMLleftmargin` or `\HTMLleftskip` the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

6.11.30 `siunitx` package

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

6.11.31 `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 `svgmath`.
 Pkg `nicefrac`

6.11.32 `newclue` package

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

Ex:

```

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

```

6.11.33 newtxmath package

Pkg `newtxmath` The proper load order is:

 loading sequence

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

6.11.34 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` patches `\FBcolonspace` and `\FBthinspace` to create fixed-width HTML spaces around punctuation:

```
\renewcommand*{\FBcolonspace}{~}
\renewcommand*{\FBthinspace}{\,}
```

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

6.11.35 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.

`lwarpmk` has the commands `printglossary` and `htmlglossary` to process the glossaries created by `glossaries` using `xindy`.

Opt **IndexLanguage** The package `lwarp` 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.

7 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 6. 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

Math 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.

8 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:

Bool `HTMLMarkFloats`

Default true. Adds `=== table begin` or `=== figure begin`, and `=== end` around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames.

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

9 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 L^AT_EX output. See section 65 for an example of the `picture` environment.

To create a custom HTML block or inline CSS class, see section 35.7.

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

An `lwarp- \langle packagename \rangle .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 150 (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.

9.2 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.

9.3 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`”.

10 Troubleshooting

10.1 Using the `lwarp.sty` package

Also see:

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

Section 6.11: [Special cases and limitations](#)

Section 6.11: [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.

Misplaced alignment character `&`:

- If using `tabular` inside a `\fbox` or other macro, use `\StartDefiningTabulars` before the definition, and `\EndDefiningTabulars` after. See section 6.11.16

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 instead. See section 6.11 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}` instead of `[warpHTML]`.

HTML document contains a single unformatted print document:

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

Images are appearing in strange places:

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

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

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 $\$x\$$ 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:

⚠ Adding/removing

⚠ HTML instead of images

⚠ Lots of files!

- 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.

10.1.1 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.

10.2 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}`

11 Implementation

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

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

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

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

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

There is still room to improve the factoring of the code, and doing so will become important if support for other output formats is added. Rather than wait until the code is pristine, the author felt it best to publish early and accept input before pushing on towards a perhaps less-than-ideal solution.

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

12 Stack depths

Stacks are created to track depth inside the \LaTeX document structure. This depth is translated to HTML headings as shown in table 6. “Depth” here is not depth in

Table 6: 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

the traditional computer-science stack-usage sense, but rather a representation of the nesting depth inside the L^AT_EX document structure.

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

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

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

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

13 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 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.

lwarp source code begins on the following page.

14 Detecting the T_EX Engine — pdf_lat_ex, lua_lat_ex, x_el_ate_x

```

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

```

15 Unicode Input Characters

for HTML & PRINT: If using pdf_lat_ex, 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

```

16 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 `comment` Provides conditional code blocks.
- ```
25 \RequirePackage{comment}
26 \excludecomment{testing}
```
- Pkg `letltxmacro` Used to redefine `\textbf` and friends.
- ```
27 \RequirePackage{letltxmacro}
```

17 Operating-System portability

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

17.1 Common portability code

Bool **usingOSWindows** Set if the OSWindows option is used.

```
28 \newbool{usingOSWindows}
29 \boolfalse{usingOSWindows}
```

17.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

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

17.3 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.

```
31 \newcommand*{\LWR@setOSWindows}
32 {
33 \booltrue{usingOSWindows}
34 \renewcommand*{\OSPathSymbol}{\@backslashchar}
35 }
```

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

```

36 \ifwindows
37 \LWR@setOSWindows
38 \fi

```

18 Package options

Pkg `kvoptions` Allows key/value package options.

```

39 \RequirePackage{kvoptions}
40 \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:

```

41 \newbool{warpingprint}
42 \newbool{warpingHTML}
43 \newbool{mathjax}

```

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

Only process the contents if producing printed output.

```

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

```

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

Only process the contents if producing HTML output.

```

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

```

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

```

46 \includecomment{warpall}

```

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

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

```

47 \DeclareVoidOption{warpprint}{%
48 \PackageInfo{lwrap}{Using option 'warpprint'}
49 \includecomment{warpprint}%

```

```

50 \excludecomment{warpHTML}%
51 \booltrue{warpingprint}%
52 \boolfalse{warpingHTML}%
53 }

```

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

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

```

54 \DeclareVoidOption{warpHTML}{%
55 \PackageInfo{lwrap}{Using option 'warpHTML'}
56 \excludecomment{warpprint}%
57 \includecomment{warpHTML}%
58 \booltrue{warpingHTML}%
59 \boolfalse{warpingprint}%
60 }

```

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.

```

61 \DeclareVoidOption{mathsvg}{%
62 \PackageInfo{lwrap}{Using option 'mathsvg'}
63 \boolfalse{mathjax}%
64 }

```

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

```

65 \DeclareVoidOption{mathjax}{%
66 \PackageInfo{lwrap}{Using option 'mathjax'}
67 \booltrue{mathjax}%
68 }

```

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.

```

69 \DeclareStringOption[\jobname]{BaseJobname}

```

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

```

70 \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

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

```
71 \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`:

```
72 \DeclareVoidOption[lwarpmk]{
73 \PackageInfo[lwarp]{Using option 'lwarpmk'}
74 \includecomment{LWR@createlwarpmk}
75 }
```

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.

```
76 \DeclareVoidOption{OSWindows}{
77 \PackageInfo[lwarp]{Using option 'OSWindows'}
78 \LWR@setOSWindows
79 }
```

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.

```
80 \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.

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

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

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

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

```

83 \includecomment{warpprint}%
84 \excludecomment{warpHTML}%
85 \booltrue{warpingprint}%
86 \boolfalse{warpingHTML}%
87 \boolfalse{mathjax}%

```

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

```
88 \excludecomment{LWR@createlwarpmk}
```

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

```
89 \ProcessKeyvalOptions*\relax
```

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

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

Defaults unless already over-ridden by the user:

```

91 \ifcsemt{LWR@HomeHTMLFilename}{
92 \newcommand*{\HomeHTMLFilename}{\BaseJobname}
93 }{
94 \csedef{HomeHTMLFilename}{\LWR@HomeHTMLFilename}
95 }
96
97 \csedef{HTMLFilename}{\LWR@HTMLFilename}

```

19 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`.

```

98 \newcommand*{\LWR@loadafter}[1]{%
99 \@ifpackageloaded{#1}
100 {
101 \PackageError{lwarp}
102 {Package #1, or one which uses #1, must be loaded after lwarp}

```

```

103 {Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{lwarp}.
104 Package #1 may also be loaded by something else, which must also be moved
105 after lwarp.}
106 }
107 {}
108 }

```

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

```

109 \newcommand*\LWR@loadbefore[1]{%
110 \@ifpackageloaded{#1}
111 {}
112 {
113 \PackageError{lwarp}
114 {Package #1 must be loaded before lwarp}
115 {Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{lwarp}.}
116 }
117 }

```

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

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

```

118 \newcommand*\LWR@loadnever[2]{%
119 \PackageError{lwarp}
120 {Package #1 is not supported by lwarp's HTML conversion.
121 Package(s) #2 may be useful instead}
122 {Package #1 might conflict with lwarp in some way,
123 or is superceded by another package.
124 For a possible alternative, see package(s) #2.}
125 }

```

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:

```

126 \LWR@loadafter{abstract}
127 \LWR@loadafter{afterpage}
128 \LWR@loadafter{algorithmicx}
129 \LWR@loadafter{alltt}
130 \LWR@loadafter{amsthm}
131 \LWR@loadafter{array}
132 \LWR@loadafter{balance}
133 \LWR@loadafter{bookmark}
134 \LWR@loadafter{booktabs}
135 \LWR@loadafter{boxedminipage}
136 \LWR@loadafter{boxedminipage2e}

```

137 \LWR@loadafter{ccaption}
138 \LWR@loadafter{change page}
139 \LWR@loadafter{crop}
140 \LWR@loadafter{cutwin}
141 \LWR@loadafter{dcolumn}
142 \LWR@loadafter{draftwatermark}
143 \LWR@loadafter{ellipsis}
144 \LWR@loadafter{emptypage}
145 \LWR@loadafter{enumerate}
146 \LWR@loadafter{enumitem}
147 \LWR@loadafter{epigraph}
148 \LWR@loadafter{eso-pic}
149 \LWR@loadafter{every page}
150 \LWR@loadafter{everyshi}
151 \LWR@loadafter{extramarks}
152 \LWR@loadafter{fancybox}
153 \LWR@loadafter{fancyhdr}
154 \LWR@loadafter{fancyvrb}
155 \LWR@loadafter{figcaps}
156 \LWR@loadafter{float}
157 \LWR@loadafter{floatflt}
158 \LWR@loadafter{floatpag}
159 \LWR@loadafter{floatrow}
160 \LWR@loadafter{flushend}
161 \LWR@loadafter{ftnright}
162 \LWR@loadafter{fullpage}
163 \LWR@loadafter{geometry}
164 \LWR@loadafter{glossaries}
165 % \LWR@loadafter{graphics}% pre-loaded by xunicode
166 % \LWR@loadafter{graphicx}% pre-loaded by xunicode
167 \LWR@loadafter{hyperref}
168 \LWR@loadafter{hyperxmp}
169 \LWR@loadafter{idxlayout}
170 \LWR@loadafter{indentfirst}
171 \LWR@loadafter{keyfloat}
172 \LWR@loadafter{layout}
173 \LWR@loadafter{letterspace}
174 \LWR@loadafter{letrine}
175 \LWR@loadafter{lips}
176 \LWR@loadafter{listings}
177 \LWR@loadafter{longtable}
178 \LWR@loadafter{lscap}
179 \LWR@loadafter{ltcaption}
180 \LWR@loadafter{marginfit}
181 \LWR@loadafter{marginfix}
182 \LWR@loadafter{marginnote}
183 \LWR@loadafter{mcaption}
184 \LWR@loadafter{mdframed}
185 \LWR@loadafter{microtype}
186 \LWR@loadafter{moreverb}

```
187 \LWR@loadafter{mparhack}
188 %\LWR@loadafter{multicol}% loaded by ltxdoc
189 \LWR@loadafter{multirow}
190 \LWR@loadafter{nameref}
191 \LWR@loadafter{needspace}
192 \LWR@loadafter{newtxmath}
193 \LWR@loadafter{nextpage}
194 \LWR@loadafter{nowidow}
195 \LWR@loadafter{intheorem}
196 \LWR@loadafter{pagenote}
197 \LWR@loadafter{paralist}
198 \LWR@loadafter{parskip}
199 \LWR@loadafter{pdfscape}
200 \LWR@loadafter{pdfsync}
201 \LWR@loadafter{placeins}
202 \LWR@loadafter{prelim2e}
203 \LWR@loadafter{ragged2e}
204 \LWR@loadafter{rotating}
205 \LWR@loadafter{rotfloat}
206 \LWR@loadafter{savetrees}
207 \LWR@loadafter{setspace}
208 \LWR@loadafter{shadow}
209 \LWR@loadafter{showidx}
210 \LWR@loadafter{showkeys}
211 \LWR@loadafter{sidecap}
212 \LWR@loadafter{sidenotes}
213 \LWR@loadafter{soul}
214 \LWR@loadafter{subfig}
215 \LWR@loadafter{tabularx}
216 \LWR@loadafter{tabulary}
217 \LWR@loadafter{textpos}
218 \LWR@loadafter{theorem}
219 \LWR@loadafter{threeparttable}
220 \LWR@loadafter{tikz}
221 \LWR@loadafter{titleps}
222 \LWR@loadafter{titlesec}
223 \LWR@loadafter{titletoc}
224 \LWR@loadafter{tocloft}
225 \LWR@loadafter{trivfloat}
226 \LWR@loadafter{ulem}
227 \LWR@loadafter{varioref}
228 \LWR@loadafter{verse}
229 \LWR@loadafter{wallpaper}
230 \LWR@loadafter{wrapfig}
231 \LWR@loadafter{xcolor}
232 \LWR@loadafter{xfrac}
233 \LWR@loadafter{xmpincl}
```

20 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: 234 `\begin{warpall}`

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

Detects $X_{\text{T}}\text{T}_{\text{E}}\text{X}$ and $\text{LuaL}^{\text{A}}\text{T}_{\text{E}}\text{X}$:

```

235 \RequirePackage{iftex}
236 \newif\ifxetexorluatex
237 \ifXeTeX
238   \xetexorluatextrue
239 \else
240   \ifLuaTeX
241     \xetexorluatextrue
242   \else
243     \xetexorluatexfalse
244   \fi
245 \fi

246 \end{warpall}

```

for HTML output: 247 `\begin{warpHTML}`

```

248 \ifxetexorluatex
249 % ^^A \usepackage[no-math]{fontspec}

```

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

```

250 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
251 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
252 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
253 \else

```

pdf_latex only: Only pre-loaded if `pdflatex` is being used.

Pkg `microtype`

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

```

254 \RequirePackage {microtype}
255
256 \microtypesetup{
257 protrusion=false,
258 expansion=false,
259 tracking=false,
260 kerning=false,
261 spacing=false}
262
263 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}

264 \fi

265 \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: 266 \begin{warpHTML}
267 \RequirePackage[paperheight=190in,paperwidth=20in,%
268 left=2in,right=12in,%
269 top=1in,bottom=1in,%
270 ]{geometry}
271 \@twosidefalse
272 \@mparswitchfalse
273 \end{warpHTML}

```

```

for HTML & PRINT: 274 \begin{warpall}

```

Pkg **xparse**

L^AT_EX3 command argument parsing

```

275 \RequirePackage{xparse}

276 \end{warpall}

```

```
for HTML output: 277 \begin{warpHTML}

    Pkg  expl3

           LATEX3 programming

    278 \RequirePackage{expl3}

Pkg  gettitlestring

           Used to emulate \nameref.

    279 \RequirePackage{gettitlestring}

    Pkg  everyhook

           everyhook is used to patch paragraph handling.

    280 \RequirePackage{everyhook}
    281 \end{warpHTML}

for HTML & PRINT: 282 \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

    283 \RequirePackage{filecontents}
    284
    285 \@ifpackagelater{filecontents}{2011/10/09}%
    286 {}
    287 {
    288 \newwrite\fcwrite
    289 \let\LWR@origfilec@ntents\filec@ntents
    290 \def\filec@ntents{\def\chardef##1\write{\let\reserved@c\fcwrite}\LWR@origfilec@ntents}
    291 }

    292 \end{warpall}

for HTML output: 293 \begin{warpHTML}

    Pkg  xifthen

    294 \RequirePackage{xifthen}
```

Pkg **xstring**

295 `\RequirePackage{xstring}`

Pkg **xstring**

296 `\RequirePackage{verbatim}`

Pkg **makeidx**

297 `\RequirePackage{makeidx}`

298 `\makeindex`

Pkg **calc**

299 `\RequirePackage{calc}`

Pkg **refcount**

300 `\RequirePackage{refcount}`

Pkg **newfloat**

301 `\RequirePackage{newfloat}`

Pkg **caption**

302 `\RequirePackage{caption}`

303 `\end{warpHTML}`

for HTML & PRINT: 304 `\begin{warpall}`

Pkg **titling**

Used for `\maketitle` and the title page. See section [48](#).

305 `\RequirePackage{titling}`

Pkg **environ**

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

306 `\RequirePackage{environ}`

307 `\end{warpall}`

for HTML output: 308 `\begin{warpHTML}`

Pkg **zref**

Used for cross-references.

```
309 \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.

```
310 \PassOptionsToPackage{leqno}{amsmath}
311 \RequirePackage{amsmath}
```

Pkg **xfrac**

Patched for HTML use. See section 189.

```
312 \RequirePackage{xfrac}
```

Used to convert lengths for image width/height options.

```
313 \RequirePackage{printlen}
```

```
314 \end{warpHTML}
```

21 Loading packages

for HTML output: 315 `\begin{warpHTML}`

Remember the original `\RequirePackage`:

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

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

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

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

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

```

318 \newcommand*\LWR@findword[3][,]{%
319   \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
320 }

```

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

```

321 \newcommand*\LWR@lookforpackagename}[1]{%

```

Find the n 'th package name from the list:

```

322 \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`.

```

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

```

See if the package name was found:

```

324 \IfStrEq{\LWR@strresulttwo}{}%
325 {}% no filename
326 {}% yes filename

```

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

```

327 \IfFileExists{lwarp-\LWR@strresulttwo.sty}%
328 {}% lwarp-* file found
329 \StrSubstitute%
330 {\LWR@requirepackagenames}%
331 {\LWR@strresult}%
332 {lwarp-\LWR@strresulttwo}[\LWR@requirepackagenames]%
333 }%
334 {}% no lwarp-* file
335 }% yes filename
336 }

```

`\RequirePackage` [$\langle 1: options \rangle$] [$\langle 2: package names \rangle$] [$\langle 3: version \rangle$]

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

```

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

```

Redirect up to nine names:

```

338 \renewcommand*\LWR@requirepackagenames}{#2}

```

```

339 \LWR@lookforpackagename{1}
340 \LWR@lookforpackagename{2}
341 \LWR@lookforpackagename{3}
342 \LWR@lookforpackagename{4}
343 \LWR@lookforpackagename{5}
344 \LWR@lookforpackagename{6}
345 \LWR@lookforpackagename{7}
346 \LWR@lookforpackagename{8}
347 \LWR@lookforpackagename{9}

```

\RequirePackage depending on the options and version:

```

348 \IfValueTF{#1}
349 {% options given
350 \IfValueTF{#3}% version given?
351 {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}[#3]}
352 {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}
353 }
354 {% no options given
355 \IfValueTF{#3}% version given?
356 {\LWR@origRequirePackage{\LWR@requirepackagenames}[#3]}
357 {\LWR@origRequirePackage{\LWR@requirepackagenames}}
358 }
359 }
360 \LetLtxMacro{\usepackage}{\RequirePackage}

```

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

Uses the original package, including options.

```

361 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
362 \PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
363 \IfValueTF{#2}
364 {\ProvidesPackage{lwarp-#1}[#2]}
365 {\ProvidesPackage{lwarp-#1}}
366 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
367 \ProcessOptions\relax
368
369 \IfValueTF{#2}
370 {\LWR@origRequirePackage{#1}[#2]}
371 {\LWR@origRequirePackage{#1}}
372 }

```

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

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

```

373 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
374 \PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
375 \IfValueTF{#2}
376 {\ProvidesPackage{lwarp-#1}[#2]}
377 {\ProvidesPackage{lwarp-#1}}
378 \DeclareOption*{}
379 \ProcessOptions\relax
380 }

381 \end{warpHTML}

```

22 Copying a file

for HTML output: 382 \begin{warpHTML}

`\LWR@copyfile` $\{ \langle source\ filename \rangle \} \{ \langle destination\ filename \rangle \}$

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

```

383 \newcommand*{\LWR@copyfile}[2]{%
384 \newwrite\copyfile % open the file to write to
385 \immediate\openout\copyfile=#2
386 \newread\file % open the file to read from
387 \openin\file=#1
388 \begingroup\endlinechar=-1
389 \makeatletter
390 \loop\unless\ifeof\file
391 \read\file to\fileline % Read one line and store it into \fileline
392 % \fileline\par % print the content into the pdf
393 % print the content:
394 \immediate\write\copyfile{\unexpanded\expandafter{\fileline}}%
395 \repeat
396 \closeout\copyfile
397 \endgroup
398 }

399 \end{warpHTML}

```

23 Debugging messages

400 \begin{warpall}

```

Bool LWR@tracinglwarp True if tracing is turned on.
401 \newbool{LWR@tracinglwarp}

\tracinglwarp Turns on the debug tracing messages.
402 \newcommand{\tracinglwarp}{\booltrue{LWR@tracinglwarp}}

\LWR@traceinfo {{text}} If tracing is turned on, writes the text to the .log file.
403 \newcommand{\LWR@traceinfo}[1]{%
404 \ifbool{LWR@tracinglwarp}%
405 {%
406 \typeout{*** lwarp: #1}%
407 % \PackageInfo{lwarp}{#1 : }%
408 }%
409 {}%
410 }

Bool HTMLDebugComments Default false. Add comments in HTML about closing <div>s, sections, etc.
411 \newbool{HTMLDebugComments}
412 \boolfalse{HTMLDebugComments}

If \tracinglwarp, show where preamble hooks occur:
413 \AfterEndPreamble{
414 \LWR@traceinfo{AfterEndPreamble}
415 }
416
417 \AtBeginDocument{
418 \LWR@traceinfo{AtBeginDocument}
419 }

```

24 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 chap-
ter/section.
420 \newbool{FormatEPUB}
421 \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.

```
422 \newbool{FormatWordProcessor}
423 \boolfalse{FormatWordProcessor}
```

Bool `HTMLMarkFloats` Default true. Adds `=== table begin` or `=== figure begin`, and `=== end` around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames. (Perhaps some day word processors will have HTML import options for identifying `<div>` classes for figures and tables.)

```
424 \newbool{HTMLMarkFloats}
425 \booltrue{HTMLMarkFloats}
```

```
426 \end{warpall}
```

25 Remembering original formatting macros

for HTML output: 427 `\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.

```
428 \LetLtxMacro{\LWR@origtextrm}{\textrm}
429 \LetLtxMacro{\LWR@origtextsf}{\textsf}
430 \LetLtxMacro{\LWR@origtexttt}{\texttt}
431 \LetLtxMacro{\LWR@origtextnormal}{\textnormal}
432 \LetLtxMacro{\LWR@origtextbf}{\textbf}
433 \LetLtxMacro{\LWR@origtextmd}{\textmd}
434 \LetLtxMacro{\LWR@origtextit}{\textit}
435 \LetLtxMacro{\LWR@origtextsl}{\textsl}
436 \LetLtxMacro{\LWR@origtextsc}{\textsc}
437 \LetLtxMacro{\LWR@origtextup}{\textup}
438 \LetLtxMacro{\LWR@origemph}{\emph}
439
440 \LetLtxMacro{\LWR@origrmfamily}{\rmfamily}
441 \LetLtxMacro{\LWR@origsfamily}{\sffamily}
442 \LetLtxMacro{\LWR@origttfamily}{\ttfamily}
443 \LetLtxMacro{\LWR@origbfseries}{\bfseries}
444 \LetLtxMacro{\LWR@origmdseries}{\mdseries}
445 \LetLtxMacro{\LWR@origupshape}{\upshape}
446 \LetLtxMacro{\LWR@origslshape}{\slshape}
```

```
447 \LetLtxMacro{\LWR@origscshape}{\scshape}
448 \LetLtxMacro{\LWR@origitshape}{\itshape}
449 \LetLtxMacro{\LWR@origem}{\em}
450 \LetLtxMacro{\LWR@orignormalfont}{\normalfont}
451
452 \let\LWR@origraggedright\raggedright
453 \let\LWR@origonecolumn\onecolumn
454
455 \let\LWR@origtextsuperscript\textsuperscript
456 \let\LWR@origtextsubscript\textsubscript
457
458 \let\LWR@origscriptsize\scriptsize
459
460 \let\LWR@orignewpage\newpage
461
462 \let\LWR@origpagestyle\pagestyle
463 \let\LWR@origthispagestyle\thispagestyle
464 \let\LWR@origpagenumbering\pagenumbering
465
466 \LetLtxMacro{\LWR@origminipage}{\minipage}
467 \let\LWR@origendminipage\endminipage
468 \LetLtxMacro{\LWR@origparbox}{\parbox}
469
470 \let\LWR@orignewline\newline
471
472 \LetLtxMacro{\LWR@origitem}{\item}
473
474 \LetLtxMacro{\LWR@origitemize}{\itemize}
475 \LetLtxMacro{\LWR@endorigitemize}{\enditemize}
476 \LetLtxMacro{\LWR@origenumerate}{\enumerate}
477 \LetLtxMacro{\LWR@endorigenumerate}{\endenumerate}
478 \LetLtxMacro{\LWR@origdescription}{\description}
479 \LetLtxMacro{\LWR@endorigdescription}{\enddescription}
480
481
482 \let\LWR@origpar\par
483
484
485 \LetLtxMacro{\LWR@origfootnote}{\footnote}
486 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
487
488 \let\LWR@origclearpage\clearpage
489 \let\clearpage\relax
490 \let\cleardoublepage\relax
491 \end{warpHTML}
```

26 Configuration Files

```
492 \begin{warpprint}
493 \typeout{lwarp: generating configuration files}
494 \end{warpprint}
```

26.1 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.

```
495 \begin{warpprint}
496 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
497 \immediate\openout{LWR@file}=\jobname_html.tex
498 \immediate\write{LWR@file}{%
499 \detokenize{\PassOptionsToPackage}%
500 {warpHTML,BaseJobname=\jobname}{lwarp}%
501 }
502 \immediate\write{LWR@file}{%
503 \detokenize{\input}\string{\jobname.tex}\string }%
504 }
505 \immediate\closeout{LWR@file}
506 \end{warpprint}
```

26.2 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:

```
507 \begin{warpprint}
508 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
509 \immediate\openout{LWR@file}=lwarpmk.conf
510 \ifbool{usingOSWindows}{
```

```

511 \immediate\write\LWR@file{opsystem = "Windows"}
512 }{
513 \immediate\write\LWR@file{opsystem = "Unix"}
514 }
515 \ifPDFTeX
516 \immediate\write\LWR@file{latexname = "pdflatex"}
517 \fi
518 \ifXeTeX
519 \immediate\write\LWR@file{latexname = "xelatex"}
520 \fi
521 \ifLuaTeX
522 \immediate\write\LWR@file{latexname = "lualatex"}
523 \fi
524 \immediate\write\LWR@file{sourcename = "\jobname"}
525 \immediate\write\LWR@file{%
526 homehtmlfilename = "\HomeHTMLFilename"%
527 }
528 \immediate\write\LWR@file{htmlfilename = "\HTMLFilename"}
529 \immediate\write\LWR@file{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
530 \immediate\write\LWR@file{language = "\LWR@IndexLanguage"}
531 \immediate\write\LWR@file{xdyfile = "\LWR@xdyFilename"}
532 \immediate\closeout\LWR@file
533 \end{warpprint}

```

26.3 project.lwarpmkconf

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

```

534 \begin{warpprint}
535 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
536 \immediate\openout\LWR@file=\jobname.lwarpmkconf
537 \ifbool{usingOSWindows}{
538 \immediate\write\LWR@file{opsystem = "Windows"}
539 }{
540 \immediate\write\LWR@file{opsystem = "Unix"}
541 }
542 \ifPDFTeX
543 \immediate\write\LWR@file{latexname = "pdflatex"}
544 \fi
545 \ifXeTeX
546 \immediate\write\LWR@file{latexname = "xelatex"}
547 \fi
548 \ifLuaTeX
549 \immediate\write\LWR@file{latexname = "lualatex"}
550 \fi
551 \immediate\write\LWR@file{sourcename = "\jobname"}
552 \immediate\write\LWR@file{%
553 homehtmlfilename = "\HomeHTMLFilename"%

```

```
554 }
555 \immediate\write\LWR@file{htmlfilename = "\HTMLFilename"}
556 \immediate\write\LWR@file{latexmk = "\ifbool{LWR@latexmk}{true}{false}}
557 \immediate\write\LWR@file{language = "\LWR@IndexLanguage"}
558 \immediate\write\LWR@file{xdyfile = "\LWR@xdyFilename"}
559 \immediate\closeout\LWR@file
560 \end{warpprint}
```

26.4 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.

```
561 \begin{warpprint}
562 \begin{filecontents*}{lwarp.css}
563 /*
564  CSS stylesheet for the LaTeX lwarp package
565  Copyright 2016-2017 Brian Dunn -- BD Tech Concepts LLC
566 */
567
568
569 /* a fix for older browsers: */
570 header, section, footer, aside, nav, main,
571     article, figure { display: block; }
572
573
574 A:link {color:#000080 ; text-decoration: none ; }
575 A:visited {color:#800000 ; }
576 A:hover {color:#000080 ; text-decoration: underline ;}
577 A:active {color:#800000 ; }
578
579 a.tocpart {display: inline-block ; margin-left: 0em ;
580     font-weight: bold ;}
581 a.tocchapter {display: inline-block ; margin-left: 0em ;
582     font-weight: bold ;}
583 a.tocsection {display: inline-block ; margin-left: 1em ;
584     text-indent: -.5em ; font-weight: bold ; }
585 a.tocsubsection {display: inline-block ; margin-left: 2em ;
586     text-indent: -.5em ; }
587 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
588     text-indent: -.5em ; }
589 a.tocparagraph {display: inline-block ; margin-left: 4em ;
590     text-indent: -.5em ; }
591 a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
592     text-indent: -.5em ; }
```

```
593 a.tocfigure {margin-left: 0em}
594 a.tocsubfigure {margin-left: 2em}
595 a.toctable {margin-left: 0em}
596 a.tocsubtable {margin-left: 2em}
597 a.toctheorem {margin-left: 0em}
598 a.toclstlisting {margin-left: 0em}
599
600
601 body {
602     font-family: "DejaVu Serif", "Bitstream Vera Serif",
603         "Lucida Bright", Georgia, serif;
604     background: #FAF7F4 ;
605     color: black ;
606     margin:0em ;
607     padding:0em ;
608     font-size: 100% ;
609     line-height: 1.2 ;
610 }
611
612 p {margin: 1.5ex 0em 1.5ex 0em ;}
613
614 /* Holds a section number to add space between it and the name */
615 span.sectionnumber { margin-right: .6em }
616
617 /* Inserted in front of index lines */
618 span.indexitem {margin-left: 0em}
619 span.indexsubitem {margin-left: 2em}
620 span.indexsubsubitem {margin-left: 4em}
621
622 div.hidden, span.hidden { display: none ; }
623
624 kbd {
625     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
626         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
627         "Courier New", monospace;
628     font-size: 100% ;
629 }
630
631 pre { padding: 3pt ; }
632
633 span.strong { font-weight: bold; }
634
635 span.textmd { font-weight: normal; }
636
637 span.textsc { font-variant: small-caps; }
638
639 span.textsl { font-style: oblique; }
640
641 span.textup { font-variant: normal; }
642
```

```
643 span.textrm {
644     font-family: "DejaVu Serif", "Bitstream Vera Serif",
645     "Lucida Bright", Georgia, serif;
646 }
647
648 span.textsf {
649     font-family: "DejaVu Sans", "Bitstream Vera Sans",
650     Geneva, Verdana, sans-serif ;
651 }
652
653 span.attribution {
654     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
655 }
656
657 span.citetitle {
658     margin-left: 1em ; font-size: 80% ; font-style: oblique;
659 }
660
661 span.poemtitle {
662     font-size: 120% ; font-weight: bold;
663 }
664
665 blockquote {
666     margin-left: 0px ;
667     margin-right: 0px ;
668 }
669
670 blockquote p {
671     line-height: 1.5;
672     text-align: left ;
673     font-size: .85em ;
674     margin-left: 3em ;
675     margin-right: 3em ;
676 }
677
678 blockquotation {
679     margin-left: 0px ;
680     margin-right: 0px ;
681 }
682
683 blockquotation p {
684     line-height: 1.5;
685     text-align: left ;
686     font-size: .85em ;
687     margin-left: 3em ;
688     margin-right: 3em ;
689 }
690
691 div.epigraph {
692     line-height: 1.2;
```

```
693     text-align: left ;
694     padding: 3ex 1em 0ex 1em ;
695 /*     margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
696     margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
697 /*     margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
698     font-size: .85em ;
699     max-width: 27em ;
700 }
701
702
703
704 div.epigraphsource{
705     text-align:right ;
706     margin-left:auto ;
707 /*     max-width: 50% ; */
708     border-top: 1px solid #A0A0A0 ;
709     padding-bottom: 3ex ;
710     line-height: 1.2;
711 }
712
713 div.epigraph p { padding: .5ex ; margin: 0ex ;}
714 div.epigraphsource p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
715
716
717 /* lettrine package: */
718 span.lettrine { font-size: 3ex ; float: left ; }
719 span.lettrinetext { font-variant: small-caps ; }
720
721 /* ulem and soul packages: */
722 span.uline {
723     text-decoration: underline ;
724     text-decoration-skip ;
725 }
726
727 span.uuline {
728     text-decoration: underline ;
729     text-decoration-skip ;
730     text-decoration-style: double ;
731 }
732
733 span.uwave {
734     text-decoration: underline ;
735     text-decoration-skip ;
736     text-decoration-style: wavy ;
737 }
738
739 span.sout {
740     text-decoration: line-through ;
741 }
742
```

```
743 span.xout {
744     text-decoration: line-through ;
745 }
746
747 span.dashuline {
748     text-decoration: underline ;
749     text-decoration-skip ;
750     text-decoration-style: dashed ;
751 }
752
753 span.dotuline {
754     text-decoration: underline ;
755     text-decoration-skip ;
756     text-decoration-style: dotted ;
757 }
758
759 span.letterspacing { letter-spacing: .2ex ; }
760
761 span.capsspacing {
762     font-variant: small-caps ;
763     letter-spacing: .1ex ;
764 }
765
766 span.highlight { background: #F8E800 ; }
767
768
769
770
771 html body {
772     margin: 0 ;
773     line-height: 1.2;
774 }
775
776
777 body div {
778     margin: 0ex;
779 }
780
781
782 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
783 {
784     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
785         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
786         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
787         "Times New Roman", serif;
788     font-style: normal ;
789     font-weight: bold ;
790     text-align: left ;
791 }
792
```

```
793 h1 { /* title of the entire website, used on each page */
794     text-align: center ;
795     font-size: 2.5em ;
796     padding: .4ex 0em 0ex 0em ;
797 }
798 h2 { font-size: 2.25em }
799 h3 { font-size: 2em }
800 h4 { font-size: 1.75em }
801 h5 { font-size: 1.5em }
802 h6 { font-size: 1.25em }
803 span.paragraph {font-size: 1em ; font-variant: normal ;
804     margin-right: 1em ; }
805 span.subparagraph {font-size: 1em ; font-variant: normal ;
806     margin-right: 1em ; }
807
808
809
810 /* Title of the file */
811 h1 {
812     margin: 0ex 0em 0ex 0em ;
813     line-height: 1.3;
814     text-align: center ;
815 }
816
817 /* Part */
818 h2 {
819     margin: 1ex 0em 1ex 0em ;
820     line-height: 1.3;
821     text-align: center ;
822 }
823
824 /* Chapter */
825 h3 {
826     margin: 3ex 0em 1ex 0em ;
827     line-height: 1.3;
828 }
829
830 /* Section */
831 h4 {
832     margin: 3ex 0em 1ex 0em ;
833     line-height: 1.3;
834 }
835
836 /* Sub-Section */
837 h5 {
838     margin: 3ex 0em 1ex 0em ;
839     line-height: 1.3;
840 }
841
842 /* Sub-Sub-Section */
```

```
843 h6 {
844   margin: 3ex 0em 1ex 0em ;
845   line-height: 1.3;
846 }
847
848
849 div.titlepage {
850   text-align: center ;
851 }
852
853 .footnotes {
854   font-size: .85em ;
855   margin: 3ex 1em 0ex 1em ;
856   padding-bottom: 1ex ;
857   border-top: 1px solid silver ;
858 }
859
860 .marginpar {
861   max-width:50%;
862   float:right;
863   text-align:left;
864   margin: 1ex 0.5em 1ex 1em ;
865   padding: 1ex 0.5em 1ex 0.5em ;
866   font-size: 85% ;
867   border-top: 1px solid silver ;
868   border-bottom: 1px solid silver ;
869   overflow-x: auto;
870 }
871
872 .marginpar br { margin-bottom: 2ex ; }
873
874 div.marginblock {
875   max-width:50%;
876   float:right;
877   text-align:left;
878   margin: 1ex 0.5em 1ex 1em ;
879   padding: 1ex 0.5em 1ex 0.5em ;
880   overflow-x: auto;
881 }
882
883 div.marginblock div.minipage {
884   display: block ;
885   margin: 0pt auto 0pt auto ;
886 }
887
888 div.marginblock div.minipage p { font-size: 85%;}
889
890 div.marginblock br { margin-bottom: 2ex ; }
891
892
```

```
893 section.textbody div.footnotes{
894     margin: 3ex 0em 0em 0em ;
895     border-bottom: 2px solid silver ;
896 }
897
898 .footnoteheader {
899     border-top: 2px solid silver ;
900     margin-top: 3ex ;
901     padding-top: 1ex ;
902     font-weight: bold ;
903 }
904
905 .mpfootnotes {
906     text-align: left ;
907     font-size: .85em ;
908     margin-left: 1em ;
909     border-top: 1px solid silver ;
910 }
911
912 /* Remove footnote top border in the title page. */
913 div.titlepage div.mpfootnotes {
914     border-top: none ;
915 }
916
917
918
919 ol {
920     margin: 1ex 1em 1ex 0em;
921     line-height: 1.2;
922 }
923
924 ul, body dir, body menu {
925     margin: 1ex 1em 1ex 0em;
926     line-height: 1.2;
927 }
928
929 li { margin: 0em 0em 1ex 0em; }
930
931 html {
932     margin: 0;
933     padding: 0;
934 }
935
936 .programlisting {
937     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
938         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
939         "Courier New", monospace;
940     margin: 1ex 0em 1ex 0em ;
941     padding: .5ex 0pt .5ex 0pt ;
942     overflow-x: auto;
```

```
943 }
944
945 section.textbody>pre.programlisting {
946 border-top: 1px solid silver ;
947 border-bottom: 1px solid silver ;
948 }
949
950
951 .inlineprogramlisting {
952 font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
953             "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
954             "Courier New", monospace;
955 overflow-x: auto;
956 }
957
958 span.listinglabel {
959 display: inline-block ;
960 font-size: 70% ;
961 width: 4em ;
962 text-align: right ;
963 margin-right: 2em ;
964 }
965
966 div.abstract {
967 margin: 2em 5% 2em 5% ;
968 padding: 1ex 1em 1ex 1em ;
969 /* font-weight: bold ; */
970 font-size: 90% ;
971 }
972
973 div.abstract dl {line-height:1.5;}
974 div.abstract dt {color:#304070;}
975
976 div.abstracttitle{
977 font-family: "URW Classico", Optima, "Linux Biolinum 0",
978             "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
979             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
980 font-weight:bold;
981 font-size:1.25em;
982 text-align: center ;
983 }
984
985 span.abstracruntintitle{
986 font-family: "URW Classico", Optima, "Linux Biolinum 0",
987             "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
988             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
989 font-weight:bold;
990 }
991
992
```

```
993 .verbatim {
994     overflow-x: auto ;
995 }
996
997 .alltt {
998     overflow-x: auto ;
999 }
1000
1001
1002 .bverbatim {
1003     margin: 1ex Opt 1ex Opt ;
1004     padding: .5ex Opt .5ex Opt ;
1005     overflow-x: auto ;
1006 }
1007
1008 .lverbatim {
1009     margin: 1ex Opt 1ex Opt ;
1010     padding: .5ex Opt .5ex Opt ;
1011     overflow-x: auto ;
1012 }
1013
1014 .fancyvrb {
1015     font-size:.85em ;
1016     margin: 3ex Opt 3ex Opt
1017 }
1018
1019 .fancyvrblabel {
1020     font-weight:bold;
1021     text-align: center ;
1022 }
1023
1024
1025 .verse {
1026     font-family: "Linux Libertine Mono O", "Lucida Console",
1027                 "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1028                 "Liberation Mono", "FreeMono", "Andale Mono",
1029                 "Nimbus Mono L", "Courier New", monospace;
1030     margin-left: 1em ;
1031 }
1032
1033
1034 div.singlespace { line-height: 1.2 ; }
1035 div.onehalfspace { line-height: 1.5 ; }
1036 div.doublespace { line-height: 2 ; }
1037
1038
1039
1040
1041
1042 /* Minipage environments, vertically aligned to top, center, bottom: */
```

```
1043 .minipage {
1044     /* display: inline-block ; */
1045     /* Mini pages which follow each other will be tiled. */
1046     margin: .25em .25em .25em .25em;
1047     padding: .25em .25em .25em .25em;
1048     display: inline-flex;
1049     flex-direction: column ;
1050     overflow: auto;
1051 }
1052
1053 /* Paragraphs in the flexbox did not collapse their margins. */
1054 /* Have not yet researched this. */
1055 .minipage p {margin: .75ex 0em .75ex 0em ;}
1056
1057 .shadowbox, .shabox {
1058     border: 1px solid black;
1059     box-shadow: 3px 3px 3px #808080 ;
1060     border-radius: 0px ;
1061     padding: .4ex .3em .4ex .3em ;
1062     margin: 0pt .3ex 0pt .3ex ;
1063     display: inline-block ;
1064 }
1065
1066 .doublebox {
1067     border: 3px double black;
1068     border-radius: 0px ;
1069     padding: .4ex .3em .4ex .3em ;
1070     margin: 0pt .3ex 0pt .3ex ;
1071     display: inline-block ;
1072 }
1073
1074 .ovalbox, .Ovalbox {
1075     border: 1px solid black;
1076     border-radius: 1ex ;
1077     padding: .4ex .3em .4ex .3em ;
1078     margin: 0pt .3ex 0pt .3ex ;
1079     display: inline-block ;
1080 }
1081
1082 .Ovalbox { border-width: 2px ; }
1083
1084
1085
1086 .framebox {
1087     border: 1px solid black;
1088     border-radius: 0px ;
1089     padding: .3ex .2em 0ex .2em ;
1090     margin: 0pt .1ex 0pt .1ex ;
1091     display: inline-block ;
1092 }
```

```
1093
1094
1095 .mdframed {
1096 /* padding: 0ex ; */
1097 /* border: 1px solid black; */
1098 /* border-radius: 0px ; */
1099 padding: 0ex ;
1100 margin: 3ex 5% 3ex 5% ;
1101 /* display: inline-block ; */
1102 }
1103
1104 .mdframed p { padding: 0ex .5em 0ex .5em ; }
1105
1106 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
1107
1108 .mdframedtitle {
1109 padding: .5em ;
1110 display: block ;
1111 font-size: 130%
1112 }
1113
1114 .mdframedsubtitle {
1115 padding: 0ex .5em 0ex .5em ;
1116 display: block ;
1117 font-size: 115% ;
1118 }
1119
1120 .mdframedsubsubtitle {
1121 padding: 0ex .5em 0ex .5em ;
1122 display: block ;
1123 }
1124
1125 .mdtheorem {
1126 padding: 0ex .5em 0ex .5em ;
1127 margin: 3ex 5% 3ex 5% ;
1128 /* display: inline-block ; */
1129 }
1130
1131
1132 /* framed package */
1133 .framed, pre.boxedverbatim {
1134 margin: 3ex 0em 3ex 0em ;
1135 border: 1px solid black;
1136 border-radius: 0px ;
1137 padding: .3ex 1em 0ex 1em ;
1138 display: block ;
1139 }
1140
1141 .snugframed {
1142 margin: 3ex 0em 3ex 0em ;
```

```
1143     border: 1px solid black;
1144     border-radius: 0px ;
1145     display: block ;
1146 }
1147
1148 .framedleftbar {
1149     margin: 3ex 0em 3ex 0em ;
1150     border-left: 3pt solid black;
1151     border-radius: 0px ;
1152     padding: .3ex .2em .3ex 1em ;
1153     display: block ;
1154 }
1155
1156 .framedtitle {
1157 margin: 0em ;
1158 padding: 0em ;
1159     font-size: 130%
1160 }
1161
1162 .framedtitle p { padding: .3em }
1163
1164
1165
1166 dl {
1167     margin: 1ex 2em 1ex 0em;
1168     line-height: 1.3;
1169 }
1170
1171 dl dt {
1172 margin-top: 1ex;
1173 margin-left: 1em ;
1174     font-weight: bold;
1175 }
1176
1177 dl dd p { margin-top: 0em; }
1178
1179
1180 nav.toc, nav.lof, nav.lot, nav.lol, nav.lothm {
1181     font-family: "URW Classico", Optima, "Linux Biolinum O",
1182         "DejaVu Sans", "Bitstream Vera Sans",
1183         Geneva, Verdana, sans-serif ;
1184     margin-bottom: 4ex ;
1185 }
1186
1187 nav.toc p, nav.lof p, nav.lot p, nav.lol p, nav.lothm p {
1188     line-height: 1.2 ;
1189     margin-top:.5ex ;
1190     margin-bottom:.5ex;
1191     font-size: .9em ;
1192 }
```

```
1193
1194
1195
1196 img, img.hyperimage, img.borderimage {
1197     max-width: 600px;
1198     border: 1px solid silver;
1199     box-shadow: 3px 3px 3px #808080 ;
1200     padding: .5% ;
1201     margin: .5% ;
1202     background: none ;
1203 }
1204
1205 img.inlineimage{
1206     padding: 0px ;
1207     box-shadow: none ;
1208     border: none ;
1209     background: none ;
1210     margin: 0px ;
1211     display: inline-block ;
1212     border-radius: 0px ;
1213 }
1214
1215 img.logoimage{
1216     max-width: 300px ;
1217     box-shadow: 3px 3px 3px #808080 ;
1218     border: 1px solid black ;
1219     background:none ;
1220     padding:0 ;
1221     margin:.5ex ;
1222     border-radius: 10px ;
1223 }
1224
1225
1226 .section {
1227 /*
1228     To have each section float relative to each other:
1229 */
1230 /*
1231     display: block ;
1232     float: left ;
1233     position: relative ;
1234     background: white ;
1235     border: 1px solid silver ;
1236     padding: .5em ;
1237 */
1238     margin: 0ex .5em 0ex .5em ;
1239     padding: 0 ;
1240 }
1241
1242
```

```
1243 figure {
1244     margin: 3ex auto 3ex auto ;
1245     padding: 1ex 1em 1ex 1em ;
1246     overflow-x: auto ;
1247 }
1248
1249
1250 /* To automatically center images in figures: */
1251 /*
1252 figure img.inlineimage {
1253     margin: 0ex auto 0ex auto ;
1254     display: block ;
1255 }
1256 */
1257
1258 /* To automatically center minipages in figures: */
1259 /*
1260 figure div.minipage, figure div.minipage div.minipage {
1261     margin: 1ex auto 1ex auto ;
1262     display: block ;
1263 }
1264 */
1265
1266 figure div.minipage p { font-size: 85% ; }
1267
1268 figure.subfigure, figure.subtable {
1269     display: inline-block ; margin: 3ex 1em 3ex 1em ;
1270 }
1271
1272 figcaption .minipage { margin:0 ; padding: 0 }
1273
1274 div.floatrow { text-align: center; }
1275
1276 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
1277
1278 div.floatfoot { font-size: .85em ;
1279     border-top: 1px solid silver ; line-height: 1.2 ; }
1280
1281 figcaption , .lstlistingtitle {
1282     font-size: .85em ;
1283     text-align: center ;
1284     font-weight: bold ;
1285     margin-top: 1ex ;
1286     margin-bottom: 1ex ;
1287 }
1288
1289 figure.subfigure figcaption, figure.subtable figcaption {
1290     border-bottom: none ; background: none ;
1291 }
1292
```

```
1293 div.nonfloatcaption {
1294     margin: 1ex auto 1ex auto ;
1295     font-size: .85em ;
1296     text-align: center ;
1297     font-weight: bold ;
1298 }
1299
1300 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
1301 figure div.floatrow div.minipage figcaption {
1302 border: none ;
1303 background: none ;
1304 }
1305
1306
1307 table {
1308     margin: 1ex auto 1ex auto ;
1309     border-collapse: collapse ;
1310     border-spacing: 0px ;
1311     line-height: 1.3 ;
1312 }
1313
1314 tr.hline {border-top: 1px solid silver ; margin-top: 0ex ;
1315     margin-bottom: 0ex ; } /* for \hline */
1316
1317 tr.tbrule {border-top: 1px solid black ; margin-top: 0ex ;
1318     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
1319
1320 td {padding: 1ex .5em 1ex .5em ;}
1321
1322 table td.tdl { text-align: left ; vertical-align: middle ; }
1323 table td.tdc { text-align: center ; vertical-align: middle ; }
1324 table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
1325 table td.tdbang { text-align: center ; vertical-align: middle ; }
1326 table td.tdr { text-align: right ; vertical-align: middle ; }
1327 table td.tdp { text-align: left ; vertical-align: bottom ; }
1328 table td.tdm { text-align: left ; vertical-align: middle ; }
1329 table td.tdb { text-align: left ; vertical-align: top ; }
1330 table td.tdP { text-align: center ; vertical-align: bottom ; }
1331 table td.tdM { text-align: center ; vertical-align: middle ; }
1332 table td.tdB { text-align: center ; vertical-align: top ; }
1333 table td.tdlrule { text-align: left ; border-top: 1px solid silver ;
1334     vertical-align: middle ; } /* for cmidrule */
1335 table td.tdcrule { text-align: center ; border-top: 1px solid silver ;
1336     vertical-align: middle ; }
1337 table td.tdatrule { text-align: center ; border-top: 1px solid silver ;
1338     vertical-align: middle ; padding: 0px ; margin: 0px ; }
1339 table td.tdbangrule { text-align: center ; border-top: 1px solid silver ;
1340     vertical-align: middle ; } /* for cmidrule */
1341 table td.tdrrule { text-align: right ; border-top: 1px solid silver ;
1342     vertical-align: middle ; }
```

```
1343 table td.tdprule { text-align: left ; border-top: 1px solid silver ;
1344     vertical-align: bottom ; }
1345 table td.tdmrule { text-align: left ; border-top: 1px solid silver ;
1346     vertical-align: middle ; }
1347 table td.tdbrule { text-align: left ; border-top: 1px solid silver ;
1348     vertical-align: top ; }
1349 table td.tdPrule { text-align: center ; border-top: 1px solid silver ;
1350     vertical-align: bottom ; }
1351 table td.tdMrule { text-align: center ; border-top: 1px solid silver ;
1352     vertical-align: middle ; }
1353 table td.tdBrule { text-align: center ; border-top: 1px solid silver ;
1354     vertical-align: top ; }
1355
1356 /* Margins of paragraphs inside table cells: */
1357 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
1358     padding-bottom: 1ex ; margin: 0ex ; }
1359 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
1360     padding-bottom: 1ex ; margin: 0ex ; }
1361 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
1362     padding-bottom: 1ex ; margin: 0ex ; }
1363
1364 td.tdp , td.tdprule , td.tdP , td.tdPrule
1365     { padding: 0ex .5em 0ex .5em ; }
1366 td.tdm , td.tdmrule , td.tdM , td.tdMrule
1367     { padding: 0ex .5em 0ex .5em ; }
1368 td.tdb , td.tdbrule , td.tdB , td.tdBrule
1369     { padding: 0ex .5em 0ex .5em ; }
1370
1371
1372 /* table notes: */
1373 .tnotes {
1374     margin: 0ex 5% 1ex 5% ;
1375     padding: 0.5ex 1em 0.5ex 1em;
1376     font-size:.85em;
1377     text-align: left ;
1378 }
1379
1380 .tnotes dl dt p {margin-bottom:0px;}
1381
1382 .tnoteitemheader {margin-right: 1em;}
1383
1384
1385
1386 /* center, flushleft, flushright environments */
1387 div.center{text-align:center;}
1388 div.center table {margin-left:auto;margin-right:auto;}
1389 div.flushleft{text-align:left;}
1390 div.flushleft table {margin-left:0em ; margin-right:auto;}
1391 div.flushright{text-align:right;}
1392 div.flushright table {margin-left:auto ; margin-right: 0em ;}
```

```
1393
1394
1395 /* Fancybox */
1396 div.Btrivlist table tr td { padding: .2ex 0em ; }
1397
1398
1399 /* program listing callouts: */
1400 span.callout {
1401     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1402         Geneva, Verdana, sans-serif ;
1403     border-radius: .5em;
1404     background-color:black;
1405     color:white;
1406     padding:0px .25em 0px .25em;
1407 margin: 0 ;
1408     font-weight: bold;
1409     font-size:.72em ;
1410 }
1411
1412 div.programlisting pre.verbatim span.callout{
1413 font-size: .85em ;
1414 }
1415
1416
1417
1418
1419
1420 div.published
1421 {
1422     text-align: center ;
1423     font-variant: normal ;
1424     font-style: italic ;
1425     font-size: 1em ;
1426     margin: 3ex 0em 3ex 0em ;
1427 }
1428
1429 div.subtitle
1430 {
1431     text-align: center ;
1432     font-variant: normal ;
1433     font-style: italic ;
1434     font-size: 1.25em ;
1435     margin: 3ex 0em 3ex 0em ;
1436 }
1437
1438 div.subtitle p { margin: 1ex ; }
1439
1440 div.author
1441 {
1442     font-variant: normal ;
```

```
1443     font-style: normal ;
1444     font-size: 1em ;
1445     margin: 3ex 0em 3ex 0em ;
1446 }
1447
1448 div.author table {
1449     margin: 3ex auto 0ex auto ;
1450     background: none ;
1451 }
1452
1453 div.author table tbody tr td { padding: .25ex ; }
1454
1455 span.affiliation {font-size: .85em ; font-variant: small-caps; }
1456
1457 div.titledate {
1458     text-align: center ;
1459     font-size: .85em ;
1460     font-style: italic;
1461     margin: 6ex 0em 6ex 0em ;
1462 }
1463
1464
1465 nav.topnavigation{
1466     text-align: left ;
1467     padding: 0.5ex 1em 0.5ex 1em ;
1468 /*     margin: 2ex 0em 3ex 0em ; */
1469     margin: 0 ;
1470     border-bottom: 1px solid silver ;
1471     border-top: 1px solid silver ;
1472     clear:right ;
1473 }
1474
1475 nav.botnavigation{
1476     text-align: left ;
1477     padding: 0.5ex 1em 0.5ex 1em ;
1478 /*     margin: 3ex 0em 2ex 0em ; */
1479     margin: 0 ;
1480     border-top: 1px solid silver ;
1481     border-bottom: 1px solid silver ;
1482     clear:right ;
1483 }
1484
1485
1486 header{
1487     line-height: 1.2 ;
1488     font-size: 1em ;
1489 /*     border-bottom: 2px solid silver ; */
1490     margin: 0px ;
1491     padding: 0ex 1em 0ex 1em ;
1492     text-align:center ;
```

```
1493 }
1494
1495 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
1496
1497
1498 footer{
1499     font-size: .85em ;
1500     line-height: 1.2 ;
1501     margin-top: 1ex ;
1502     border-top: 2px solid silver ;
1503     padding: 2ex 1em 2ex 1em ;
1504     clear:right ;
1505     text-align:left ;
1506 }
1507
1508
1509 a.linkhome { font-weight:bold ; font-size: 1em ;}
1510
1511
1512 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
1513
1514 img.lateximage{
1515     padding: 0px 0px 0px 0px ;
1516     box-shadow: none ;
1517     border: none ;
1518     background: none ;
1519     margin: 0px 0px -.15ex 0px ;
1520     /* pdfcrop leaves a slight margin, adjust to baseline */
1521     max-width: 100% ;
1522     border-radius: 0ex ;
1523     border: none ;
1524 }
1525
1526
1527
1528 nav.sidetoc {
1529     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1530         "Lucida Bright", Georgia, serif;
1531     float:right ;
1532     width: 20%;
1533     border-left: 1px solid silver;
1534     border-top: 1px solid silver;
1535     border-bottom: 1px solid silver;
1536 /*     border-top: 2px solid #808080 ; */
1537     background: #FAF7F4 ;
1538     padding: 2ex 0em 2ex 1em ;
1539     margin: 0ex 0em 2ex 1em ;
1540     font-size:.9em ;
1541     border-radius: 20px 0px 0px 20px ;
1542 }
```

```
1543
1544 div.sidetoccontents {
1545 /*    border-top: 1px solid silver ; */
1546     overflow-y: auto ;
1547     width: 100% ;
1548     text-align: left ;
1549 }
1550
1551 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
1552     text-indent: 0 ; }
1553 nav.sidetoc p a {color:black ; font-size: .7em ;}
1554 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
1555     border-bottom: 1px solid silver ;    }
1556 nav.sidetoc a:hover {text-decoration: underline ; }
1557
1558
1559
1560 section.textbody { margin: 0ex 1em 0ex 1em ;}
1561
1562
1563 div.multicolsheading { -webkit-column-span: all;
1564     -moz-column-span: all; column-span: all; }
1565 div.multicols { -webkit-columns: 3 380px ;
1566     -moz-columns: 3 380px ; columns: 3 380px ; }
1567 div.multicols p {margin-top: 0ex}
1568
1569
1570
1571 /* Used to support algorithmicx: */
1572 span.floatright { float: right ; }
1573
1574
1575
1576
1577 /* Native LaTeX theorems: */
1578
1579 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
1580 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
1581
1582
1583 /* theorem, amsthm, and ntheorem packages */
1584
1585 span.theoremheader,
1586 span.theoremheaderplain,
1587 span.theoremheaderdefinition,
1588 span.theoremheaderbreak,
1589 span.theoremheadermarginbreak,
1590 span.theoremheaderchangebreak,
1591 span.theoremheaderchange,
1592 span.theoremheadermargin
```

```
1593 {
1594 font-style:normal ; font-weight: bold ; margin-right: 1em ;
1595 }
1596
1597 span.amsthmnameplain,
1598 span.amsthmnamedefinition,
1599 span.amsthmnumberplain,
1600 span.amsthmnumberdefinition
1601 {
1602 font-style:normal ; font-weight: bold ;
1603 }
1604
1605
1606 span.amsthmnameremark,
1607 span.amsthmnumberremark
1608 {font-style:italic ; font-weight: normal ; }
1609
1610
1611 span.amsthmnoteplain,
1612 span.amsthmnotedefinition
1613 {font-style:normal ;}
1614
1615
1616 span.theoremheaderremark,
1617 span.theoremheaderproof,
1618 span.amsthmproofname
1619 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
1620
1621 span.theoremheadersc
1622 {
1623 font-style:normal ;
1624 font-variant: small-caps ;
1625 font-weight: normal ;
1626 margin-right: 1em ;
1627 }
1628
1629 .theoremendmark {float:right}
1630
1631 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
1632 div.theorembodybreak, div.theorembodynonumberbreak,
1633 div.theorembodymarginbreak,
1634 div.theorembodychangebreak,
1635 div.theorembodychange,
1636 div.theorembodymargin
1637 {
1638 font-style:italic;
1639 margin-top: 3ex ; margin-bottom: 3ex ;
1640 }
1641
1642 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
```

```
1643 div.theorembodyplainupright, nonumberplainuprightsc,
1644 div.amsthmbodydefinition, div.amsthmbodyremark,
1645 div.amsthmproof
1646 {
1647 font-style: normal ;
1648 margin-top: 3ex ; margin-bottom: 3ex ;
1649 }
1650
1651 span.amsthmnoteremark {}
1652
1653
1654
1655 /*
1656 For CSS LaTeX and related logos:
1657 Based on:
1658 http://edward.oconnor.cx/2007/08/tex-poshlet
1659 http://nitens.org/taraborelli/texlogo
1660 */
1661
1662 .latexlogofont {
1663     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1664         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1665     font-variant: normal ;
1666 }
1667
1668 .latexlogo {
1669     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1670         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1671     letter-spacing: .03em ;
1672     font-size: 1.1em;
1673 }
1674
1675 .latexlogo sup {
1676     text-transform: uppercase;
1677     letter-spacing: .03em ;
1678     font-size: 0.85em;
1679     vertical-align: 0.15em;
1680     margin-left: -0.36em;
1681     margin-right: -0.15em;
1682 }
1683
1684 .latexlogo sub {
1685     text-transform: uppercase;
1686     vertical-align: -0.5ex;
1687     margin-left: -0.1667em;
1688     margin-right: -0.125em;
1689     font-size: 1em;
1690 }
1691
1692 .xetexlogo {
```

```
1693     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1694             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1695     letter-spacing: .03em ;
1696     font-size: 1.1em;
1697 }
1698
1699 /* A smaller gap between Xe and Tex v.s. LaTeX: */
1700 .xetexlogo sub {
1701     text-transform: uppercase;
1702     vertical-align: -0.5ex;
1703     margin-left: -0.0667em;
1704     margin-right: -0.2em;
1705     font-size: 1em;
1706     letter-spacing: .03em ;
1707 }
1708
1709 /* A large gap between Xe and LaTeX v.s. TeX: */
1710 .xelatexlogo sub {
1711     text-transform: uppercase;
1712     vertical-align: -0.5ex;
1713     margin-left: -0.0667em;
1714     margin-right: -.05em;
1715     font-size: 1em;
1716     letter-spacing: .03em ;
1717 }
1718
1719 .amslogo {
1720     font-family: "TeXGyreChorus","URW Chancery L",
1721             "Apple Chancery","ITC Zapf Chancery","Monotype Corsiva",
1722             "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
1723             "Hoefler Text", Times, "Times New Roman", serif;
1724     font-style: italic;
1725 }
1726
1727 .lyxlogo {
1728     font-family: "URW Classico", Optima, "Linux Biolinum O",
1729             "DejaVu Sans", "Bitstream Vera Sans", Geneva,
1730     Verdana, sans-serif ;
1731 }
1732
1733
1734
1735
1736 /* Only display top and bottom navigation if a small screen: */
1737 /* Hide the sidetoc if a small screen: */
1738 nav.topnavigation { display:none; }
1739 nav.botnavigation { display:none; }
1740
1741 @media screen and (max-width: 45em) {
1742 /*     nav.sidetoc {display:none;} */
```

```
1743     nav.sidetoc {
1744         float: none ;
1745         width: 100% ;
1746         margin: 5ex 0px 5ex 0px ;
1747         padding: 0 ;
1748         border-radius: 0 ;
1749         border-bottom: 1px solid black ;
1750         border-top: 1px solid black ;
1751         box-shadow: none ;
1752     }
1753 /*     nav.topnavigation { display:block } */
1754     nav.botnavigation { display:block }
1755     .marginpar {
1756         max-width: 100%;
1757         float: none;
1758         display:block ;
1759         margin: 1ex 1em 1ex 1em ;
1760     }
1761 }
1762
1763 @media print {
1764     body {
1765         font-family: "Linux Libertine O",
1766         "DejaVu Serif", "Bitstream Vera Serif",
1767         "Liberation Serif", "Nimbus Roman No 9 L",
1768         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1769     }
1770     nav.sidetoc { display:none; }
1771     nav.topnavigation { display: none; }
1772     nav.botnavigation { display: none; }
1773 }
1774
1775 @media handheld {
1776     nav.sidetoc { display:none; }
1777     nav.topnavigation { display:block }
1778     nav.botnavigation { display:block }
1779 }
1780
1781 @media projection {
1782     nav.sidetoc { display:none; }
1783     nav.topnavigation { display:block }
1784     nav.botnavigation { display:block }
1785 }
1786 \end{filecontents*}
1787 % \end{Verbatim}% for syntax highlighting
1788 \end{warpprint}
```

26.5 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.

```
1789 \begin{warpprint}
1790 \begin{filecontents*}{lwarp_sagebrush.css}
1791 @import url("lwarp.css") ;
1792
1793
1794 A:link {color:#105030 ; text-decoration: none ; }
1795 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
1796 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
1797 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
1798
1799
1800
1801 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1802 {
1803     font-family: "URW Classico", Optima, "Linux Biolinum O",
1804         "Linux Libertine O", "Liberation Serif",
1805         "Nimbus Roman No 9 L", "FreeSerif",
1806         "Hoefler Text", Times, "Times New Roman", serif;
1807     font-variant: small-caps ;
1808     font-weight: normal ;
1809     color: #304070 ;
1810     text-shadow: 2px 2px 3px #808080;
1811 }
1812
1813 h1 { /* title of the entire website, used on each page */
1814     font-variant: small-caps ;
1815     color: #304070 ;
1816     text-shadow: 2px 2px 3px #808080;
1817     background-color: #F7F7F0 ;
1818     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
1819 }
1820
1821 h1 {
1822     border-bottom: 1px solid #304070;
1823     border-top: 2px solid #304070;
1824 }
1825
1826 h2 {
1827     border-bottom: 1px solid #304070;
1828     border-top: 2px solid #304070;
1829     background-color: #F7F7F0 ;
1830     background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
```

```
1831 }
1832
1833
1834
1835 div.abstract {
1836     background: #f5f5eb ;
1837     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1838
1839     border: 1px solid silver;
1840     border-radius: 1em ;
1841 }
1842
1843 div.abstract dl {line-height:1.5;}
1844 div.abstract dt {color:#304070;}
1845
1846 div.abstracttitle{
1847     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1848         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
1849         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1850     font-weight:bold;
1851     font-variant: small-caps ;
1852     font-size:1.5em;
1853     border-bottom: 1px solid silver ;
1854     color: #304070 ;
1855     text-align: center ;
1856     text-shadow: 1px 1px 2px #808080;
1857 }
1858
1859 span.abstracrunintitle{
1860     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1861         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
1862         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1863     font-weight:bold;
1864 }
1865
1866
1867 div.epigraph {
1868     background: #f5f5eb ;
1869     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1870
1871     border: 1px solid silver ;
1872     border-radius: 1ex ;
1873     box-shadow: 3px 3px 3px #808080 ;
1874 }
1875
1876
1877 .example {
1878     background-color: #f5f5eb ;
1879     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1880
```

```
1881 }
1882
1883 div.exampletitle{
1884     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1885         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
1886         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1887     font-weight:bold;
1888     font-variant: small-caps ;
1889     border-bottom: 1px solid silver ;
1890     color: #304070 ;
1891     text-align: center ;
1892     text-shadow: 1px 1px 2px #808080;
1893 }
1894
1895
1896 .sidebar {
1897     background-color: #f5f5eb ;
1898     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1899
1900 }
1901
1902 div.sidebartitle{
1903     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1904         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
1905         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1906     font-weight:bold;
1907     font-variant: small-caps ;
1908     border-bottom: 1px solid silver ;
1909     color: #304070 ;
1910     text-align: center ;
1911     text-shadow: 1px 1px 2px #808080;
1912 }
1913
1914
1915 .fancyvrblabel {
1916     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1917         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
1918         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1919     font-weight:bold;
1920     font-variant: small-caps ;
1921     font-size: 1.5em ;
1922     color: #304070 ;
1923     text-align: center ;
1924     text-shadow: 1px 1px 2px #808080;
1925 }
1926
1927
1928
1929 .minipage {
1930     background-color: #eeeeee7 ;
```

```
1931     border: 1px solid silver ;
1932 border-radius: 1ex ;
1933 }
1934
1935 .framed .minipage , .framedleftbar .minipage {
1936 border: none ;
1937 background: none ;
1938 padding: 0ex ;
1939 margin: 0ex ;
1940 }
1941
1942 figure.figure .minipage, figcaption .minipage { border: none; }
1943
1944 div.marginblock div.minipage { border: none; }
1945
1946 figure , div.marginblock {
1947     background-color: #eeeeee7 ;
1948     border: 1px solid silver ;
1949     border-radius: 1ex ;
1950     box-shadow: 3px 3px 3px #808080 ;
1951 }
1952
1953 figure figure {
1954     border: 1px solid silver ;
1955     margin: 0em ;
1956 box-shadow: none ;
1957 }
1958
1959 /*
1960 figcaption {
1961     border-top: 1px solid silver ;
1962     border-bottom: 1px solid silver ;
1963     background-color: #e8e8e8 ;
1964 }
1965 */
1966
1967
1968 div.table {
1969     box-shadow: 3px 3px 3px #808080 ;
1970 }
1971
1972 /*
1973 .tnotes {
1974     background: #e8e8e8;
1975     border: 1px solid silver;
1976 }
1977 */
1978
1979
1980 nav.topnavigation{
```

```
1981     background-color: #b0b8b0 ;
1982     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
1983 }
1984
1985 nav.botnavigation{
1986     background-color: #b0b8b0 ;
1987     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
1988 }
1989
1990
1991
1992 header{
1993     background-color: #F7F7F0 ;
1994     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
1995 }
1996
1997 footer{
1998     background-color: #F7F7F0 ;
1999     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
2000 }
2001
2002
2003
2004 nav.sidetoc {
2005     background-color: #F7F7F0 ;
2006     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
2007     box-shadow: 3px 3px 3px #808080 ;
2008     border-radius: 0px 0px 0px 20px ;
2009 }
2010
2011 div.sidetocitle {color: #304070 ; }
2012
2013 nav.sidetoc a:hover {
2014     color:#006000 ;
2015     text-decoration: none ;
2016     text-shadow:0px 0px 2px #a0a0a0;
2017 }
2018
2019
2020 @media screen and (max-width: 45em) {
2021     nav.sidetoc { border-radius: 0 ; }
2022 }
2023
2024
2025 \end{filecontents*}
2026 % \end{Verbatim}% for syntax highlighting
2027 \end{warpprint}
```

26.6 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.

```
2028 \begin{warpprint}
2029 \begin{filecontents*}{lwarp_formal.css}
2030 @import url("lwarp.css") ;
2031
2032
2033
2034 A:link {color:#802020 ; text-decoration:none; }
2035 A:visited {color:#802020 ; text-shadow:none ;}
2036 A:hover {color:#400000 ; text-shadow:none ;}
2037 A:active {color:#C00000 ; text-shadow:none ;}
2038
2039
2040 body {
2041     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2042         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2043         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2044         "Times New Roman", serif;
2045     background: #fffcf5;
2046 }
2047
2048 span.textrm {
2049     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2050         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2051         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2052         "Times New Roman", serif;
2053 }
2054
2055 span.textsf {
2056     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2057         Geneva, Verdana, sans-serif ;
2058 }
2059
2060
2061
2062 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2063 {
2064     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2065         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2066         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2067         "Times New Roman", serif;
2068     color: #800000 ;
2069     text-shadow: none ;
```

```
2070 }
2071
2072 h1, h2 {
2073     background-color: #fffcf5 ;
2074     background-image: none ;
2075     border-bottom: 1px solid #808080;
2076     border-top: 2px solid #808080;
2077 }
2078
2079 div.abstracttitle {
2080     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2081         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2082         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2083         "Times New Roman", serif;
2084     color: black ;
2085     text-shadow: none ;
2086 }
2087
2088 span.abstracrunintitle {
2089     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2090         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2091         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2092         "Times New Roman", serif;
2093     color: black ;
2094     text-shadow: none ;
2095 }
2096
2097 div.abstract { font-size: 100% }
2098
2099 .sidebar {
2100     background: #fffcf5;
2101     background-image: none ;
2102     margin: 2em 5% 2em 5%;
2103     padding: 0.5em 1em;
2104     border: none ;
2105     border-top : 1px solid silver;
2106     border-bottom : 1px solid silver;
2107     font-size: 90% ;
2108 }
2109
2110 div.sidebartitle{
2111     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2112         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2113         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2114         "Times New Roman", serif;
2115     color: #800000 ;
2116     text-shadow: none ;
2117     border: none ;
2118 }
2119
```

```
2120 .example {
2121     background: #fffcf5;
2122     background-image: none ;
2123     margin: 2em 5% 2em 5%;
2124     padding: 0.5em 1em;
2125     border: none ;
2126     border-top : 1px solid silver;
2127     border-bottom : 1px solid silver;
2128 }
2129
2130 div.exampletitle{
2131     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2132         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2133         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2134         "Times New Roman", serif;
2135     color: #800000 ;
2136     text-shadow: none ;
2137     border: none ;
2138 }
2139
2140 div.fancyvrblabel{
2141     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2142         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2143         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2144         "Times New Roman", serif;
2145     color: #800000 ;
2146     text-shadow: none ;
2147     border: none ;
2148 }
2149
2150
2151
2152 .verse {
2153     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2154         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2155         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2156         "Times New Roman", serif;
2157 }
2158
2159
2160 figure {
2161     margin: 3ex 5% 3ex 5% ;
2162     padding: 1ex 1em 1ex 1em ;
2163     background-color: #fffcf5 ;
2164     overflow-x: auto ;
2165     border: none ;
2166 /*     border-top: 1px solid silver; */
2167 /*     border-bottom: 1px solid silver; */
2168 }
2169
```

```
2170
2171 figcaption , .lstlisting {
2172     border: none ;
2173 /*     border-top: 1px solid silver ; */
2174 /*     border-bottom: 1px solid silver ; */
2175     background-color: #fffcf5 ;
2176 }
2177
2178 .tnotes {
2179     background: #fffcf5 ;
2180 }
2181
2182 .theorem {
2183     background: none ;
2184 }
2185
2186 .minipage {
2187     background-color: #fffcf5 ;
2188     border: none ;
2189 }
2190
2191 div.floatrow figure { border: none ; }
2192
2193 figure figure { border: none ; }
2194
2195
2196 nav.toc, nav.lof, nav.lot, nav.lol {
2197     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2198         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2199         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2200         "Times New Roman", serif;
2201 }
2202
2203 nav.sidetoc {
2204     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2205         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2206         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2207         "Times New Roman", serif;
2208     background-image: linear-gradient(to bottom, #fffcf5, #COCOC0);
2209     border-radius: 0px 0px 0px 20px ;
2210 }
2211
2212 div.sidetoc{
2213     color: #800000 ;
2214 }
2215
2216 header{
2217     background-color: #e0e0e0 ;
2218     background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
2219     text-align:center ;
```

```

2220 }
2221
2222 footer{
2223     background-color: #e0e0e0 ;
2224     background-image: linear-gradient(to bottom, #ffcf5, #b0b0b0);
2225     padding: 2ex 1em 2ex 1em ;
2226     clear:right ;
2227     text-align:left ;
2228 }
2229
2230 nav.botnavigation {
2231     background: #dedcd5 ;
2232     border-top: 1px solid black ;
2233 }
2234 \end{filecontents*}
2235 % \end{Verbatim}% for syntax highlighting
2236 \end{warpprint}

```

26.7 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.

```

2237 \begin{warpprint}
2238 \begin{filecontents*}{sample_project.css}
2239 /* ( --- Start of project.css --- ) */
2240 /* A sample project-specific CSS file for lwarp --- ) */
2241
2242 /* Load default lwarp settings: */
2243 @import url("lwarp.css") ;
2244 /* or lwarp_formal.css, lwarp_sagebrush.css */
2245
2246 /* Project-specific CSS setting follow here. */
2247 /* . . . */
2248
2249 /* ( --- End of project.css --- ) */
2250 \end{filecontents*}
2251 % \end{Verbatim}% for syntax highlighting
2252 \end{warpprint}

```

26.8 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.

```

2253 \begin{warpprint}
2254 \begin{filecontents*}{lwrap.xdy}
2255 (require "tex/inputenc/latin.xdy")
2256 (merge-rule "\\PS *" "Postscript")
2257 (require "texindy.xdy")
2258 (require "page-ranges.xdy")
2259 (require "book-order.xdy")
2260 (markup-locref :open "\hyperindexref{" :close "}")
2261 \end{filecontents*}
2262 % \end{Verbatim}% for syntax highlighting
2263 \end{warpprint}

```

26.9 lwrap_mathjax.txt

File `lwrap_mathjax.txt` Used by lwrap when using MathJax.

This must be present when compiling the project, but does not need to be present when distributing the resulting HTML files.

```

2264 \begin{warpprint}
2265 \begin{filecontents*}{lwrap_mathjax.txt}
2266 <!-- https://groups.google.com/forum/#!topic/
2267             mathjax-users/jUteWUcE2bY -->
2268 <script type="text/x-mathjax-config">
2269 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
2270     var seteqsectionDefault = {name: "", num: 0};
2271     var seteqsections = {}, seteqsection = seteqsectionDefault;
2272     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
2273     var AMS = MathJax.Extension["TeX/AMSmath"];
2274     TEX.Definitions.Add({
2275     macros: {
2276         seteqsection: "mySection",
2277         seteqnumber: "mySetEqNumber"
2278     }
2279     });
2280
2281     PARSE.Augment({
2282     mySection: function (name) {
2283         seteqsection.num = AMS.number;
2284         var n = this.GetArgument(name);
2285         if (n === "") {
2286             seteqsection = seteqsectionDefault;
2287         } else {
2288             if (!seteqsections["_"+n])

```

```

2289         seteqsections["_"+n] = {name:n, num:0};
2290     seteqsection = seteqsections["_"+n];
2291     }
2292     AMS.number = seteqsection.num;
2293 },
2294 mySetEqNumber: function (name) {
2295     var n = this.GetArgument(name);
2296     if (!n || !n.match(/^ *[0-9]+ *$/))
2297         n = ""; else n = parseInt(n)-1;
2298     <!-- $ syntax highlighting -->
2299     if (n === "" || n < 1)
2300         TEX.Error
2301             ("Argument to "+name+" should be a positive integer");
2302     AMS.number = n;
2303 }
2304 });
2305 MathJax.Hub.Config({
2306   TeX: {
2307     equationNumbers: {
2308       formatTag: function (n)
2309         {return "("+(seteqsection.name+"."+n).replace(/\./,"")+"}"},
2310       formatID: function (n) {
2311         n = (seteqsection.name+'.'+n).replace
2312           (/[:"<>&]/g,"").replace(/\./,"");
2313         return 'mjx-eqn-' + n;
2314       }
2315     }
2316   }
2317 });
2318 });
2319 </script>
2320
2321 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
2322 <script type="text/x-mathjax-config">
2323   MathJax.Ajax.config.path["Contrib"] =
2324     "https://cdn.mathjax.org/mathjax/contrib";
2325 </script>
2326
2327 <!-- https://github.com/mathjax/MathJax-third-party-extensions/
2328         tree/master/siunitx -->
2329 <script type="text/x-mathjax-config">
2330   MathJax.Hub.Config({
2331     extensions: ["tex2jax.js", "[Contrib]/siunitx/siunitx.js"],
2332     jax: ["input/TeX", "output/HTML-CSS"],
2333     tex2jax: {inlineMath: [["$","$"],["\\(","\\)"]]},
2334     TeX: {extensions: ["AMSMath.js", "AMSsymbols.js", "sinuitx.js"]}
2335   });
2336 </script>
2337
2338 <script type="text/x-mathjax-config">

```

```

2339 MathJax.Hub.Config({
2340   TeX: {
2341     equationNumbers: {
2342       autoNumber: "AMS"
2343     }
2344   }
2345 });
2346 </script>
2347
2348 <!-- Alternative CDN provider: -->
2349 <script type="text/javascript" async
2350   src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?config=TeX-AMS_HTML-full">
2351 </script>
2352
2353 <!-- No longer supported after April 30, 2017: -->
2354 <!--
2355 <script
2356   src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
2357 </script>
2358 -->
2359
2360 \end{filecontents*}
2361 % \end{Verbatim}% for syntax highlighting
2362 \end{warpprint}

```

26.10 lwarpmk option

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

```

2363 \begin{LWR@createlwarpmk}

```

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

```

2364 \begin{filecontents*}{lwarpmk.lua}
2365 #!/usr/bin/env texlua
2366
2367 -- Copyright 2016-2017 Brian Dunn
2368
2369 -- Print the usage of the lwarpmk command:
2370
2371 printversion = "v0.34"
2372
2373 function printhelp ()
2374 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
2375 end
2376
2377 function printusage ()

```

```
2378 print ( [[
2379
2380 lwarpmk print [project]: Compile a print version.
2381 lwarpmk printindex [project]: Process the index for the print version.
2382 lwarpmk printglossary [project]: Process the glossary for the print version.
2383 lwarpmk html [project]: Compile an HTML version.
2384 lwarpmk htmlindex [project]: Process the index for the html version.
2385 lwarpmk htmlglossary [project]: Process the glossary for the html version.
2386 lwarpmk again [project]: Touch the source code to trigger recompiles.
2387 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
2388 lwarpmk pdftohtml [project]:
2389     For use with latexmk or a Makefile:
2390     Convert project_html.pdf to project_html.html and
2391     individual HTML files.
2392 lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, .gl*
2393 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
2394 lwarpmk -h: Print this help message.
2395 lwarpmk --help: Print this help message.
2396
2397 ]] )
2398 printconf ()
2399 end
2400
2401 -- Print the format of the configuration file lwarpmk.conf:
2402
2403 function printconf ()
2404 print ( [[
2405 An example lwarpmk.conf or <project>.lwarpmkconf project file:
2406 --
2407 opsystem = "Unix"    (or "Windows")
2408 latexname = "pdflatex" (or "lualatex", or "xelatex")
2409 sourcename = "projectname" (the source-code filename w/o .tex)
2410 homehtmlfilename = "index" (or perhaps the project name)
2411 htmlfilename = "" (or "projectname" - filename prefix)
2412 latexmk = "false" (or "true" to use latexmk to build PDFs)
2413 language = "english" (use a language supported by xindy)
2414 xdyfile = "lwarp.xdy" (or a custom file based on lwarp.xdy)
2415 --
2416 Filenames must contain only letters, numbers, underscore, or dash.
2417 Values must be in "quotes".
2418
2419 ]] ) ;
2420 end
2421
2422
2423 -- Split one large sourcefile into a number of files,
2424 -- starting with destfile.
2425 -- The file is split at each occurrence of <!--|Start file|newfilename|*
2426
2427 function splitfile (destfile,sourcefile)
```

```
2428 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
2429 local sfile = io.open(sourcefile)
2430 io.output(destfile)
2431 for line in sfile:lines() do
2432 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*|)");
2433 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
2434 io.output(newfilename) ;
2435 else -- not a splitpoint
2436 io.write (line .. "\n") ;
2437 end
2438 end -- do
2439 io.close(sfile)
2440 end -- function
2441
2442 -- Incorrect value, so print an error and exit.
2443
2444 function cvalueerror ( line, linenum , cvalue )
2445     print ( linenum .. " : " .. line ) ;
2446     print ("lwarpmk: incorrect variable value \"" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
2447     printconf () ;
2448     os.exit(1) ;
2449 end
2450
2451 -- Load settings from the project's "lwarpmk.conf" file:
2452
2453 function loadconf ()
2454 -- Default configuration filename:
2455 local conffile = "lwarpmk.conf"
2456 -- Optional configuration filename:
2457 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
2458 -- Default language:
2459 language = "english"
2460 -- Default xdyfile:
2461 xdyfile = "lwarp.xdy"
2462 -- Verify the file exists:
2463 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
2464 print("lwarpmk: " .. conffile .. " does not exist.")
2465 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
2466 printhelp () ;
2467 os.exit(1) -- exit the entire lwarpmk script
2468 else -- file exists
2469 -- Read the file:
2470 print ("lwarpmk: Reading " .. conffile .. ".")
2471 local cfile = io.open(conffile)
2472 -- Scan each line:
2473 local linenum = 0
2474 for line in cfile:lines() do -- scan lines
2475 linenum = linenum + 1
2476 i,j,cvarname,cvalue = string.find (line,"([%w-_*])%s*=%s*".."([%w%-_*])\(") ;
2477 -- Error if incorrect enclosing characters:
```

```
2478 if ( i == nil ) then
2479 print ( linenum .. " : " .. line ) ;
2480 print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
2481 printconf ( ) ;
2482 os.exit(1) ;
2483 end
2484 if ( cvarname == "opssystem" ) then
2485   -- Verify choice of opssystem:
2486   if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
2487     opssystem = cvalue
2488   else
2489     cvalueerror ( line, linenum , cvalue )
2490   end
2491 elseif ( cvarname == "latexname" ) then
2492   -- Verify choice of LaTeX compiler:
2493   if (
2494     (cvalue == "pdflatex") or
2495     (cvalue == "xelatex") or
2496     (cvalue == "lualatex")
2497   ) then
2498     latexname = cvalue
2499   else
2500     cvalueerror ( line, linenum , cvalue )
2501   end
2502 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
2503 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
2504 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
2505 elseif ( cvarname == "latexmk" ) then latexmk = cvalue
2506 elseif ( cvarname == "language" ) then language = cvalue
2507 elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue
2508 else
2509 print ( linenum .. " : " .. line ) ;
2510 print ( "lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. conffile .. ".\n" ) ;
2511 printconf ( ) ;
2512 os.exit(1) ;
2513 end
2514 end -- do scan lines
2515 io.close(cfile)
2516 end -- file exists
2517 -- Select some operating-system commands:
2518 if opssystem=="Unix" then -- For Unix / Linux / Mac OS:
2519 rmname = "rm"
2520 mvname = "mv"
2521 touchnamepre = "touch"
2522 touchnamepost = ""
2523 dirslash = "/"
2524 opquote= "\"'"
2525 elseif opssystem=="Windows" then -- For Windows
2526 rmname = "DEL"
2527 mvname = "MOVE"
```

```
2528 touchnamepre = "COPY /b"
2529 touchnamepost = "+,,"
2530 dirslash = "\\"
2531 opquote= "\""
2532 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
2533 end --- for Windows
2534
2535 -- set xindycmd according to pdflatex vs xelatex/lualatex:
2536 if ( latexname == "pdflatex" ) then
2537 xindycmd = "texindy -C utf8"
2538 glossarycmd = "xindy -C utf8"
2539 else
2540 xindycmd = "xindy -M texindy -C utf8"
2541 glossarycmd = "xindy -C utf8"
2542 end
2543
2544 end -- loadconf
2545
2546
2547 function refreshdate ()
2548 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
2549 end
2550
2551
2552 -- Scan the LaTeX log file for the phrase "Rerun to get",
2553 -- indicating that the file should be compiled again.
2554 -- Return true if found.
2555
2556 function reruntoget (filesource)
2557 local fsource = io.open(filesource)
2558 for line in fsource:lines() do
2559 if ( string.find(line,"Rerun to get") ~= nil ) then
2560 io.close(fsource)
2561 return true
2562 end
2563 end
2564 io.close(fsource)
2565 return false
2566 end
2567
2568
2569 -- Compile one time, return true if should compile again.
2570 -- fsuffix is "" for print, "_html" for HTML output.
2571
2572 function onetime (fsuffix)
2573 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
2574 err = os.execute(
2575 -- "echo " ..
2576 latexname .. " " .. sourcename..fsuffix )
2577 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
```

```
2578 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
2579 end
2580
2581
2582 -- Compile up to five times.
2583 -- fsuffix is "" for print, "_html" for HTML output
2584
2585 function manytimes (fsuffix)
2586 if onetime(fsuffix) == true then
2587 if onetime(fsuffix) == true then
2588 if onetime(fsuffix) == true then
2589 if onetime(fsuffix) == true then
2590 if onetime(fsuffix) == true then
2591 end end end end end
2592 end
2593
2594 -- Exit if the given file does not exist.
2595
2596 function verifyfileexists (filename)
2597 if (lfs.attributes ( filename , "modification" ) == nil ) then
2598 print ( "lwarpmk: " .. filename .. " not found." ) ;
2599 os.exit (1) ;
2600 end
2601 end
2602
2603
2604 -- Convert <project>_html.pdf into HTML files:
2605
2606 function pdftohtml ()
2607     -- Convert to text:
2608     print ("lwarpmk: Converting " .. sourcename
2609         .. "_html.pdf to " .. sourcename .. "_html.html")
2610     os.execute("pdftotext -enc UTF-8 -nopgbrk -layout "
2611         .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
2612     -- Split the result into individual HTML files:
2613     splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
2614 end
2615
2616
2617 -- Remove auxiliary files:
2618
2619 function removeaux ()
2620     os.execute ( rmname .. " " ..
2621         sourcename .. ".aux " .. sourcename .. "_html.aux " ..
2622         sourcename .. ".toc " .. sourcename .. "_html.toc " ..
2623         sourcename .. ".lof " .. sourcename .. "_html.lof " ..
2624         sourcename .. ".lot " .. sourcename .. "_html.lot " ..
2625         sourcename .. ".idx " .. sourcename .. "_html.idx " ..
2626         sourcename .. ".ind " .. sourcename .. "_html.ind " ..
2627         sourcename .. ".log " .. sourcename .. "_html.log " ..
```

```
2628     sourcename ..".gl*" " .. sourcename .. "_html.gl*" "  
2629     )  
2630 end  
2631  
2632  
2633  
2634 -- Create lateximages based on lateximages.txt:  
2635 function createlateximages ()  
2636 print ("lwrapmk: Creating lateximages.")  
2637 local limagesfile = io.open("lateximages.txt")  
2638 -- Create the lateximages directory, ignore error if already exists  
2639 err = os.execute("mkdir lateximages")  
2640 -- Scan lateximages.txt  
2641 for line in limagesfile:lines() do  
2642 -- lwimgpage is the page number in the PDF which has the image  
2643 -- lwimgnum is the sequential lateximage number to assign for the image  
2644 i,j,lwimgpage,lwimgnum = string.find (line,"|(.)|(.)|")  
2645 -- For each entry:  
2646 if ( i~=nil ) then  
2647 -- Separate out the image into its own single-page pdf:  
2648 err = os.execute(  
2649 "pdfseparate -f " .. lwimgpage .. " -1 " ..  
2650 lwimgpage .. " " .. sourcename .."_html.pdf lateximagetemp-%d.pdf")  
2651 -- Crop the image:  
2652 err = os.execute(  
2653 "pdfcrop --hires lateximagetemp-" .. lwimgpage ..".pdf lateximage-" .. lwimgnum ..".pdf")  
2654 if ( err ~= 0 ) then print ( "lwrapmk: File error." ) ; os.exit(1) ; end  
2655 -- Convert the image to svg:  
2656 err = os.execute(  
2657 "pdftocairo -svg lateximage-" .. lwimgnum ..".pdf lateximage-" .. lwimgnum ..".svg")  
2658 if ( err ~= 0 ) then print ( "lwrapmk: File error." ) ; os.exit(1) ; end  
2659 -- Move the result into lateximages/:  
2660 err = os.execute(  
2661 mvname .. " lateximage-" .. lwimgnum ..".svg lateximages" .. dirslash )  
2662 if ( err ~= 0 ) then print ( "lwrapmk: File error." ) ; os.exit(1) ; end  
2663 -- Remove the temporary files:  
2664 err = os.execute(  
2665 rmname .. " lateximage-" .. lwimgnum ..".pdf lateximagetemp-" .. lwimgpage ..".pdf")  
2666 if ( err ~= 0 ) then print ( "lwrapmk: File error." ) ; os.exit(1) ; end  
2667 end  
2668 end -- do  
2669 io.close(limagesfile)  
2670 end -- function  
2671  
2672  
2673 -- Use latexmk to compile source and index:  
2674 -- fsuffix is "" for print, or "_html" for HTML  
2675 function compilelatexmk ( fsuffix )  
2676     -- The recorder option is required to detect changes in <project>.tex  
2677     -- while we are loading <project>_html.tex.
```

```
2678     err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
2679         .. "-e "
2680         .. opquote
2681         .. "$makeindex = q/" -- $
2682         .. xindycmd
2683         .. " -M " .. xdyfile
2684         .. " -L " .. language .. " /"
2685         .. opquote
2686         .. "-pdflatex=\\"" .. latexname .. "%O %S\" "
2687         .. sourcename..fsuffix ..".tex" ) ;
2688     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2689 end
2690
2691
2692
2693 -- lwarpmk --version :
2694
2695 if ( arg[1] == "--version" ) then
2696 print ( "lwarpmk: " .. printversion )
2697
2698 else -- not -- version
2699
2700 -- print intro:
2701
2702 print ( "lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package." )
2703
2704 -- lwarpmk print:
2705
2706 if arg[1] == "print" then
2707 loadconf ( )
2708 if ( latexmk == "true" ) then
2709     compilelatexmk ( "" )
2710     print ( "lwarpmk: Done." )
2711 else -- not latexmk
2712     verifyfileexists ( sourcename .. ".tex" ) ;
2713     -- See if up to date:
2714     if (
2715         ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
2716         (
2717             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2718             lfs.attributes ( sourcename .. ".pdf" , "modification" )
2719         )
2720     ) then
2721         -- Recompile if not yet up to date:
2722         manytimes( "" )
2723         print ( "lwarpmk: Done." ) ;
2724     else
2725         print ( "lwarpmk: " .. sourcename .. ".pdf is up to date." ) ;
2726     end
2727 end -- not latexmk
```

```
2728
2729 -- lwarp printindex:
2730 -- Compile the index then touch the source
2731 -- to trigger a recompile of the document:
2732
2733 elseif arg[1] == "printindex" then
2734 loadconf ()
2735 print ("lwarpmk: Processing the index.")
2736 os.execute(
2737     xindycmd
2738     .. " -M " .. xdyfile
2739     .. " -L " .. language
2740     .. " " .. sourcename .. ".idx")
2741 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2742 refreshdate ()
2743 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2744 print ("lwarpmk: Done.")
2745
2746 -- lwarp printglossary:
2747 -- Compile the glossary then touch the source
2748 -- to trigger a recompile of the document:
2749
2750 elseif arg[1] == "printglossary" then
2751 loadconf ()
2752 print ("lwarpmk: Processing the glossary.")
2753
2754 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
2755     " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
2756     .. sourcename .. ".glo")
2757 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2758 refreshdate ()
2759 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2760 print ("lwarpmk: Done.")
2761
2762 -- lwarpmk html:
2763
2764 elseif arg[1] == "html" then
2765 loadconf ()
2766 if ( latexmk == "true" ) then
2767     compilelatexmk ("_html")
2768     pdftohtml ()
2769     print ("lwarpmk: Done.")
2770 else -- not latexmk
2771     verifyfileexists ( sourcename .. ".tex" ) ;
2772     -- See if exists and is up to date:
2773     if (
2774         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
2775         (
2776             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2777             lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
```

```
2778     )
2779   ) then
2780     -- Recompile if not yet up to date:
2781     manytimes("_html")
2782     pdftohtml ()
2783     print ("lwarpmk: Done.")
2784   else
2785     print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
2786   end
2787 end -- not latexmk
2788
2789 elseif arg[1] == "pdftohtml" then
2790   loadconf ()
2791   pdftohtml ()
2792
2793 -- lwarpmk htmlindex:
2794 -- Compile the index then touch the source
2795 -- to trigger a recompile of the document:
2796
2797 elseif arg[1] == "htmlindex" then
2798 loadconf ()
2799 print ("lwarpmk: Processing the index.")
2800 os.execute(
2801   xindycmd
2802   .. " -M " .. xdyfile
2803   .. " -L " .. language
2804   .. " " .. sourcename .. "_html.idx"
2805 )
2806 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2807 refreshdate ()
2808 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2809 print ("lwarpmk: Done.")
2810
2811 -- lwarpmk htmlglossary:
2812 -- Compile the glossary then touch the source
2813 -- to trigger a recompile of the document:
2814
2815 elseif arg[1] == "htmlglossary" then
2816 loadconf ()
2817 print ("lwarpmk: Processing the glossary.")
2818
2819 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
2820   "_html -t " .. sourcename .. "_html.glg -o " .. sourcename ..
2821   "_html.gls " .. sourcename .. "_html.glo")
2822
2823 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2824 refreshdate ()
2825 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2826 print ("lwarpmk: Done.")
2827
```

```
2828 -- lwarpmk limages:
2829 -- Scan the lateximages.txt file to create lateximages,
2830 -- then touch the source to trigger a recompile.
2831
2832 elseif arg[1] == "limages" then
2833 loadconf ()
2834 print ("lwarpmk: Processing images.")
2835 createlateximages ()
2836 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2837 refreshdate ()
2838 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2839 print ("lwarpmk: Done.")
2840
2841 -- lwarpmk again:
2842 -- Touch the source to trigger a recompile.
2843
2844 elseif arg[1] == "again" then
2845 loadconf ()
2846 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2847 refreshdate ()
2848 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2849 print ("lwarpmk: Done.")
2850
2851 -- lwarpmk clean:
2852 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, .gl*
2853
2854 elseif arg[1] == "clean" then
2855 loadconf ()
2856 removeaux ()
2857 print ("lwarpmk: Done.")
2858
2859 -- lwarpmk cleanall
2860 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, .gl*
2861 -- and also project.pdf, *.html
2862
2863 elseif arg[1] == "cleanall" then
2864 loadconf ()
2865 removeaux ()
2866 os.execute ( rmname .. " " ..
2867     sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
2868     "*.html"
2869     )
2870 print ("lwarpmk: Done.")
2871
2872 -- lwarpmk with no argument :
2873
2874 elseif (arg[1] == nil) then
2875 printhelp ()
2876
2877 -- lwarpmk -h or lwarpmk --help :
```

```

2878
2879 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
2880 printusage ()
2881
2882 else
2883 print ("lwarpmk: Unknown command \""..arg[1].."\".\n")
2884 printhelp ()
2885 end
2886
2887 end -- not --version
2888 \end{filecontents*}
2889 % \end{Verbatim}% for syntax highlighting

2890 \end{LWR@createlwarpmk}

```

27 Stacks

for HTML output: 2891 \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 (unnested) 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.

27.1 Assigning depths

initial depths for empty stack entries:

```
2892 \newcommand*{\LWR@depthnone}{-5}
```

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

```

2893 \newcommand*{\LWR@depthfinished}{-4}
2894 \newcommand*{\LWR@depthpart}{-1}
2895 \newcommand*{\LWR@depthchapter}{0}
2896 \newcommand*{\LWR@depthsection}{1}
2897 \newcommand*{\LWR@depthsubsection}{2}

```

```
2898 \newcommand*\LWR@depthsubsubsection}{3}
2899 \newcommand*\LWR@depthparagraph}{4}
2900 \newcommand*\LWR@depthsubparagraph}{5}
```

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

```
2901 \newcommand*\LWR@depthlist}{6}
```

used by `\item`:

```
2902 \newcommand*\LWR@depthlistitem}{7}
```

27.2 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.

```
2903 \newcommand*\LWR@closeone}{% top of the stack
2904 \newcommand*\LWR@closetwo}{
2905 \newcommand*\LWR@closethree}{
2906 \newcommand*\LWR@closefour}{
2907 \newcommand*\LWR@closefive}{
2908 \newcommand*\LWR@closesix}{
2909 \newcommand*\LWR@closeseven}{
2910 \newcommand*\LWR@closeeight}{
2911 \newcommand*\LWR@closenine}{
2912 \newcommand*\LWR@closeten}{
2913 \newcommand*\LWR@closeeleven}{
2914 \newcommand*\LWR@closetwelve}{
```

27.3 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}
```

```
2915 \newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
2916 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
2917 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
2918 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
2919 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
2920 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
2921 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
2922 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
2923 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
2924 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}
2925 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
2926 \newcommand*{\LWR@closedephtwelve}{\LWR@depthnone}
```

27.4 Pushing and popping the stack

`\pushclose` $\{(action)\} \{(depth)\}$

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

```
2927 \NewDocumentCommand{\pushclose}{m m}
2928 {
2929 \let\LWR@closetwelve\LWR@closeeleven
2930 \let\LWR@closeeleven\LWR@closeten
2931 \let\LWR@closeten\LWR@closenine
2932 \let\LWR@closenine\LWR@closeeight
2933 \let\LWR@closeeight\LWR@closeseven
2934 \let\LWR@closeseven\LWR@closesix
2935 \let\LWR@closesix\LWR@closefive
2936 \let\LWR@closefive\LWR@closefour
2937 \let\LWR@closefour\LWR@closethree
2938 \let\LWR@closethree\LWR@closetwo
2939 \let\LWR@closetwo\LWR@closeone
2940 \let\LWR@closeone#1
2941 \let\LWR@closedephtwelve\LWR@closedeptheleven
2942 \let\LWR@closedeptheleven\LWR@closedepthten
2943 \let\LWR@closedepthten\LWR@closedepthnine
```

```

2944 \let\LWR@closedepthnine\LWR@closedeptheight
2945 \let\LWR@closedeptheight\LWR@closedepthseven
2946 \let\LWR@closedepthseven\LWR@closedepthsix
2947 \let\LWR@closedepthsix\LWR@closedepthfive
2948 \let\LWR@closedepthfive\LWR@closedepthfour
2949 \let\LWR@closedepthfour\LWR@closedepththree
2950 \let\LWR@closedepththree\LWR@closedepthtwo
2951 \let\LWR@closedepthtwo\LWR@closedepthone
2952 \let\LWR@closedepthone#2
2953 }

```

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

```

2954 \newcommand*{\popclose}
2955 {
2956 \let\LWR@closeone\LWR@closetwo
2957 \let\LWR@closetwo\LWR@closethree
2958 \let\LWR@closethree\LWR@closefour
2959 \let\LWR@closefour\LWR@closefive
2960 \let\LWR@closefive\LWR@closesix
2961 \let\LWR@closesix\LWR@closeseven
2962 \let\LWR@closeseven\LWR@closeeight
2963 \let\LWR@closeeight\LWR@closenine
2964 \let\LWR@closenine\LWR@closeten
2965 \let\LWR@closeten\LWR@closeeleven
2966 \let\LWR@closeeleven\LWR@closetwelve
2967 \let\LWR@closedepthone\LWR@closedepthtwo
2968 \let\LWR@closedepthtwo\LWR@closedepththree
2969 \let\LWR@closedepththree\LWR@closedepthfour
2970 \let\LWR@closedepthfour\LWR@closedepthfive
2971 \let\LWR@closedepthfive\LWR@closedepthsix
2972 \let\LWR@closedepthsix\LWR@closedepthseven
2973 \let\LWR@closedepthseven\LWR@closedeptheight
2974 \let\LWR@closedeptheight\LWR@closedepthnine
2975 \let\LWR@closedepthnine\LWR@closedepthten
2976 \let\LWR@closedepthten\LWR@closedeptheleven
2977 \let\LWR@closedeptheleven\LWR@closedepthtwelve
2978 }

2979 \end{warpHTML}

```

28 Data arrays

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

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

To assign an empty value:

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

for HTML output: 2980 \begin{warpHTML}

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

```
2981 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
2982 \ifthenelse{\isempty{#3}}{%
2983 {\csdef{#1#2}{}}
2984 {\expandafter\edef\csname #1#2\endcsname{\expandonce#3}}%
2985 }
```

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

```
2986 \newcommand*{\LWR@getexparray}[2]{\csuse{#1#2}}
2987 \end{warpHTML}
```

29 HTML entities

for HTML output: 2988 \begin{warpHTML}

HTML entites and HTML Unicode entities:

```
2989 \let\LWR@origampersand\&
```

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

```
2990 \newcommand*{\HTMLentity}[1]{%
2991 % \LWR@traceinfo{HTMLentity \detokenize{#1}}%
2992 \begingroup%
2993 \LWR@FBcancel%
2994 \LWR@origampersand#1;%
2995 \endgroup
2996 % \LWR@traceinfo{HTMLentity done}%
2997 }
```

```
\HTMLunicode {<hex_unicode>}
```

```
2998 \newcommand*{\HTMLunicode}[1]{\HTMLentity{\#x#1}}
```

```

\&

2999 \renewcommand*{\&}{\HTMLentity{amp}}

\textless
\textgreater
3000 \let\LWR@origtextless\textless
3001 \renewcommand*{\textless}{\HTMLentity{lt}}
3002
3003 \let\LWR@origtextgreater\textgreater
3004 \renewcommand*{\textgreater}{\HTMLentity{gt}}

3005 \end{warpHTML}

```

30 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: 3006 `\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.

```
3007 \providecommand*{\BaseJobname}{\jobname}
```

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

```
3008 \providecommand*{\HTMLFilename}{}
```

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

```
3009 \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.

```
3010 \newcommand*\SetHTMLFileNumber}[1]{%
3011 \setcounter{LWR@htmlfilenumber}{#1}%
3012 }
```

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

Defaults to use section names in the filenames.

```
3013 \newbool{FileSectionNames}
3014 \booltrue{FileSectionNames}
```

```
3015 \end{warpall}
```

for HTML output: 3016 \begin{warpHTML}

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

```
3017 \newcounter{LWR@htmlfilenumber}
3018 \setcounter{LWR@htmlfilenumber}{0}
```

`\LWR@htmlsectionfilename` $\{(\textit{htmlfilenumber or name})\}$

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

```
3019 \newcommand*\LWR@htmlsectionfilename}[1]{%
3020 \LWR@traceinfo{LWR@htmlsectionfilename A}%
```

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

```
3021 \LWR@traceinfo{about to assign temp}%
3022 \edef\LWR@tempone{#1}%
3023 \LWR@traceinfo{about to compare with ??}%
3024 \ifthenelse{\equal{\LWR@tempone}{??}}{%
3025 {%
3026 \LWR@traceinfo{found ??}%
3027 }%
3028 \LWR@traceinfo{not found ??}%
3029 }%
3030 \LWR@traceinfo{about to compare with zero or empty}%
3031 \ifthenelse{%
3032 \equal{\LWR@tempone}{0}}%
3033 \OR \equal{\LWR@tempone}{}%
3034 \OR \equal{\LWR@tempone}{??}}%
```

```

3035 }%
3036 {%
3037 \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
3038 \HomeHTMLFilename.html%
3039 }%

```

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

```

3040 {%
3041 \LWR@traceinfo{LWR@htmlsectionfilename C \LWR@tempone}%
3042 \ifthenelse{%
3043 \equal{\HTMLFilename}{ } \AND \equal{\LWR@tempone}{Index} \OR \equal{\LWR@tempone}{index}%
3044 }%
3045 {%
3046 \LWR@traceinfo{prefixing the index name with an underscore.}%
3047 \_#1.html}%

```

Otherwise, create a filename with the chosen prefix:

```

3048 {\HTMLFilename#1.html}%
3049 }%
3050 \LWR@traceinfo{LWR@htmlsectionfilename Z}%
3051 }

```

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

Prints the filename for the given label

```

3052 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
3053 \LWR@traceinfo{LWR@htmlrefsectionfilename A: !#1!}%
3054 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
3055 \LWR@traceinfo{LWR@htmlrefsectionfilename B}%
3056 }

3057 \end{warpHTML}

```

31 Homepage link

for HTML output: 3058 `\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.

```

3059 \newcommand*{\LinkHome}{%

```

```

3060 \LWR@subhyperrefclass{%
3061 \HomeHTMLFilename.html}%
3062 {Home}{linkhome}%
3063 }

```

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

```

3064 \newcommand*{\LWR@topnavigation}{
3065 \LWR@htmlElementclassline{nav}{topnavigation}{\LinkHome}
3066 }

```

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

```

3067 \newcommand*{\LWR@botnavigation}{
3068 \LWR@htmlElementclassline{nav}{botnavigation}{\LinkHome}
3069 }

```

```

3070 \end{warpHTML}

```

32 `\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: 3071 `\begin{warpHTML}`

`\PrintStack` Prints the closedepth stack.

```

3072 \newcommand*{\PrintStack}{
3073 \LWR@startpars
3074 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
3075 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
3076 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
3077 \LWR@closedepthten{} \LWR@closedeptheleven{} \LWR@closedepthtwelve{}
3078 }

```

```

3079 \end{warpHTML}

```

33 Closing stack levels

for HTML output: 3080 `\begin{warpHTML}`

Close one nested level:

```
3081 \newcommand*{\LWR@closeoneprevious}{%
3082
3083 \LWR@closeone{
3084
3085 \popclose{
3086 }
```

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

```
3087 \newcommand*{\LWR@closeprevious}[1]{
```

Close any pending paragraph:

```
3088 \LWR@stoppars
```

Close anything nested deeper than the desired depth:

```
3089 \whiledo{\not\(\LWR@closedepthone<#1\)}{\LWR@closeoneprevious}
3090 }
```

```
3091 \end{warpHTML}
```

34 PDF pages and styles

for HTML output: 3092 `\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.

```
3093 \newcommand{\LWR@forcenewpage}{%
3094 \LWR@stoppars\LWR@orignewpage\LWR@startpars%
3095 }
```

`\pagestyle` and `\thispagestyle` are nullified for HTML output.

`\pagestyle` $\langle style \rangle$

3096 `\renewcommand*\pagestyle}[1]{}`

`\thispagestyle` $\langle style \rangle$

3097 `\renewcommand*\thispagestyle}[1]{}`

`\pagenumbering` $\langle commands \rangle$

3098 `\renewcommand*\pagenumbering}[1]{}`

3099 `\end{warpHTML}`

35 HTML tags, spans, divs, elements

for HTML output: 3100 `\begin{warpHTML}`

35.1 Mapping L^AT_EX Sections to HTML Sections

```

3101 \newcommand*\LWR@tagpart}{h2}
3102 \newcommand*\LWR@tagpartend}{/h2}
3103 \newcommand*\LWR@tagchapter}{h3}
3104 \newcommand*\LWR@tagchapterend}{/h3}
3105 \newcommand*\LWR@tagsection}{h4}
3106 \newcommand*\LWR@tagsectionend}{/h4}
3107 \newcommand*\LWR@tagsubsection}{h5}
3108 \newcommand*\LWR@tagsubsectionend}{/h5}
3109 \newcommand*\LWR@tagsubsubsection}{h6}
3110 \newcommand*\LWR@tagsubsubsectionend}{/h6}
3111 \newcommand*\LWR@tagparagraph}{span class="paragraph"{} }
3112 \newcommand*\LWR@tagparagraphend}{/span}
3113 \newcommand*\LWR@tagsubparagraph}{span class="subparagraph"{} }
3114 \newcommand*\LWR@tagsubparagraphend}{/span}
3115
3116 \newcommand*\LWR@tagregularparagraph}{p}

```

35.2 HTML tags

Adjust babel-french for HTML spaces:

```

3117 \@ifpackagewith{babel}{french}{

```

```

3118 \renewcommand*{\FBcolonspace}{~}
3119 \renewcommand*{\FBthinspace}{\,}
3120 }{}

```

`\LWR@FBcancel` Cancels HTML spacing for babel-french

```

3121 \@ifpackagewith{babel}{french}{%
3122 \newcommand{\LWR@FBcancel}{%
3123 \csdef{FBcolonspace}{}%
3124 \csdef{FBthinspace}{}%
3125 }
3126 }
3127 {% not french:
3128 \newcommand{\LWR@FBcancel}{%
3129 }%

```

`\LWR@htmltagc` `{<tag>}` Break ligatures and use upright apostrophes in HTML tags.

`\protect` is in case the tag appears in TOC, LOF, LOT.

```

3130 \newcommand*{\LWR@htmltagc}[1]{%
3131 {%
3132 % \LWR@traceinfo{\LWR@htmltagc \detokenize{#1}}%
3133 \begingroup%
3134 \LWR@FBcancel%
3135 \ifmmode\else\protect\LWR@origttfamily\fi%
3136 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
3137 \endgroup%
3138 }%
3139 }

```

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.

```

3140 \newcommand*{\LWR@nestspanitem}{%
3141 \if@newlist\else{\LWR@htmltagc{br /}}\fi%
3142 \LWR@origitem%
3143 }
3144
3145 \newenvironment*{LWR@nestspan}
3146 {%
3147 \LWR@traceinfo{LWR@nestspan}%
3148 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%

```

```

3149 {%
3150 \LWR@traceinfo{LWR@nestspan: inside a lateximage}%
3151 }%
3152 {% not in a lateximage
3153 \LWR@traceinfo{LWR@nestspan: NOT inside a lateximage}%
3154 \addtocounter{LWR@spandepth}{1}%
3155 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{-}{-}%
3156 \RenewDocumentEnvironment{BlockClass}{o m}{-}{-}%
3157 \renewcommand{\BlockClassSingle}[2]{##2}%
3158 \renewcommand{\LWR@forcenewpage}{-}%
3159 \renewcommand{\LWR@itemizestart}{-}%
3160 \let\item\LWR@nestspanitem%
3161 % \global\boolfalse{LWR@allownestspanitembreak}%
3162 }%
3163 \renewcommand{\LWR@itemizeend}{-}%
3164 \renewcommand{\LWR@enumeratestart}{-}%
3165 \let\item\LWR@nestspanitem%
3166 % \global\boolfalse{LWR@allownestspanitembreak}%
3167 }%
3168 \renewcommand{\LWR@enumerateend}{-}%
3169 \renewcommand{\LWR@descriptionstart}{-}%
3170 \let\item\LWR@nestspanitem%
3171 % \global\boolfalse{LWR@allownestspanitembreak}%
3172 }%
3173 \renewcommand{\LWR@descriptionend}{-}%
3174 }% not in a lateximage
3175 }%
3176 {%
3177 \ifthenelse{\cinttest{\value{LWR@lateximagedepth}}{>}{0}}%
3178 {}%
3179 {\addtocounter{LWR@spandepth}{-1}}%
3180 \LWR@traceinfo{LWR@nestspan: done}%
3181 }
3182
3183 \AfterEndEnvironment{LWR@nestspan}{\global\let\par\LWR@closeparagraph}

```

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



`\LWR@spandepth` is used to ensure that paragraph tags are not generated inside a span. The exact sequence of when to add and subtract the counter is important to correctly handle the paragraph tags before and after the span.

```

3184 \NewDocumentCommand{\LWR@htmlspan}{m +m}{-}%
3185 \LWR@ensuredoingapar%
3186 \LWR@htmltagc{#1}%
3187 \begin{LWR@nestspan}%
3188 #2%
3189 \LWR@htmltagc{/#1}%
3190 \end{LWR@nestspan}%

```

3191 }

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

```
3192 \NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{%
3193 \LWR@ensuredoingapar%
3194 \LWR@subhtmelementclass{span}{#1}{#2}%
3195 \begin{LWR@nestspan}%
3196 #3%
3197 \LWR@htmltagc{/span}%
3198 \end{LWR@nestspan}%
3199 }
```

`\LWR@htmltag` {*tag*}

Print an HTML tag: <tag>

```
3200 \newcommand*{\LWR@htmltagb}[1]{%
3201 \LWR@htmltagc{#1}%
3202 \endgroup%
3203 }
3204
3205 \newcommand*{\LWR@htmltag}{%
3206 \begingroup\catcode'\_ =12
3207 \LWR@htmltagb%
3208 }
```

35.3 Block tags and comments

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

`\LWR@htmlopencomment`

`\LWR@htmlclosecomment`

```
3209 \newcommand*{\LWR@htmlopencomment}{%
3210 {%
3211 % \LWR@traceinfo{\LWR@htmlopencomment}%
3212 \begingroup%
3213 \LWR@FBcancel%
3214 \ifmode\else\protect\LWR@origttfamily\fi%
3215 \LWR@origtextless{}!{-}{-}%
3216 \endgroup%
3217 }%
3218 }
3219
```

```

3220 \newcommand*{\LWR@htmlclosecomment}{%
3221 {%
3222 % \LWR@traceinfo{\LWR@htmlclosecomment}%
3223 \begingroup%
3224 \LWR@FBcancel%
3225 \ifmode\else\protect\LWR@origttfamily\fi%
3226 {-}{-}\LWR@origtextgreater{}%
3227 \endgroup%
3228 }%
3229 }

```

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

```

3230 \newcommand{\LWR@htmlcomment}[1]{%
3231 \LWR@htmlopencomment{}%
3232 {%
3233 \LWR@origttfamily% break ligatures
3234 #1%
3235 }%
3236 \LWR@htmlclosecomment{}}

```

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

```

3237 \newcommand{\LWR@htmlblockcommentb}[1]
3238 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
3239
3240 \newcommand{\LWR@htmlblockcomment}
3241 {%
3242 \begingroup\catcode'\_ =12%
3243 \LWR@htmlblockcommentb%
3244 }

```

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

```

3245 \newcommand*{\LWR@htmlblocktag}[1]{%
3246 \LWR@stoppars%
3247 \LWR@htmltag{#1}%
3248 \LWR@startpars%
3249 }

```

35.4 Div class and element class

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

Factored and reused in several places.

```

3250 \NewDocumentCommand{\LWR@subhtmlclass}{m o m}{%
3251 \IfValueTF{#2}%
3252 {% option
3253 \ifthenelse{\equal{#2}{}}%
3254 {\LWR@htmltag{#1 class="#3"}}% empty option
3255 {\LWR@htmltag{#1 class="#3" style="#2"}}% non-empty option
3256 }% option
3257 {\LWR@htmltag{#1 class="#3"}}% no option
3258 }

```

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

```

3259 \NewDocumentCommand{\LWR@htmlclass}{m o m}{%
3260 \LWR@stoppars%
3261 \LWR@subhtmlclass{#1}[#2]{#3}%
3262 \LWR@startpars%
3263 }

```

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

```

3264 \newcommand*{\LWR@htmlclassend}[2]{%
3265 \LWR@stoppars%
3266 \LWR@htmltag{/#1}%
3267 \ifbool{HTMLDebugComments}{%
3268 \LWR@htmlcomment{End of #1 ‘#2’}%
3269 }{}%
3270 \LWR@startpars%
3271 }

```

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

```

3272 \NewDocumentCommand{\LWR@htmldivclass}{o m}{%
3273 \LWR@htmlclass{div}[#1]{#2}%
3274 }

```

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

```

3275 \newcommand*{\LWR@htmldivclassend}[1]{%
3276 \LWR@htmlclassend{div}{#1}%
3277 }

```

35.5 Single-line elements

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

```

\LWR@htmlclassline  {<element>} [(<style>)] {<class>} {<text>}

3278 \NewDocumentCommand{\LWR@htmlclassline}{m o m +m}{%
3279 \LWR@stoppars
3280 \LWR@subhtmlclass{#1}{#2}{#3}%
3281 #4%
3282 \LWR@htmltag{/#1}
3283 \LWR@startpars
3284 }

```

35.6 HTML5 semantic elements

```

\LWR@htmllement  {<element>}

3285 \newcommand*{\LWR@htmllement}[1]{%
3286 \LWR@htmlblocktag{#1}
3287 }

```

```

\LWR@htmllementend  {<element>}

3288 \newcommand*{\LWR@htmllementend}[1]{%
3289 \LWR@stoppars
3290 \LWR@htmltag{/#1}
3291 \LWR@startpars
3292 }
3293
3294 \end{wrapHTML}

```

35.7 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.

For other direct-formatting commands, see section [67](#).

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

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

for HTML output:

```

3295 \begin{wrapHTML}
3296 \NewDocumentEnvironment{BlockClass}{o m}%
3297 {
3298 \LWR@htmldivclass[#1]{#2}

```

```

3299 }
3300 {
3301 \LWR@htmldivclassend{#2}
3302 }
3303 \end{warpHTML}

```

for PRINT output: 3304 \begin{warpprint}
3305 \NewDocumentEnvironment{BlockClass}{o m}{}{}%
3306 \end{warpprint}

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

for HTML output: 3307 \begin{warpHTML}
3308 \newcommand{\BlockClassSingle}[2]{%
3309 \LWR@htmlclassline{div}{#1}{#2}%
3310 }
3311 \end{warpHTML}

for PRINT output: 3312 \begin{warpprint}
3313 \newcommand{\BlockClassSingle}[2]{#2}
3314 \end{warpprint}

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

for HTML output: 3315 \begin{warpHTML}
3316 \NewDocumentCommand{\InlineClass}{o m +m}{%
3317 \LWR@htmlspanclass[#1]{#2}{#3}%
3318 }
3319 \end{warpHTML}

for PRINT output: 3320 \begin{warpprint}
3321 \NewDocumentCommand{\InlineClass}{o m +m}{#3}%
3322 \end{warpprint}

35.8 Closing HTML tags

for HTML output: 3323 \begin{warpHTML}

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

```

3324 \newcommand*\LWR@printclosepart}
3325     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
3326 \newcommand*\LWR@printclosechapter}
3327     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}

```

```

3328 \newcommand*\LWR@printclosesection}
3329     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
3330 \newcommand*\LWR@printclosesubsection}
3331     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
3332 \newcommand*\LWR@printclosesubsubsection}
3333     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
3334 \newcommand*\LWR@printcloseparagraph}
3335     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
3336 \newcommand*\LWR@printclose subparagraph}
3337     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}

```

Lists require closing HTML tags:

```

3338 \newcommand*\LWR@printcloselistitem}
3339     {\LWR@htmltag{/li}}
3340 \newcommand*\LWR@printclosedescitem}
3341     {\LWR@htmltag{/dd}}
3342 \newcommand*\LWR@printcloseitemize}
3343     {\LWR@htmltag{/ul}}
3344 \newcommand*\LWR@printcloseenumerate}
3345     {\LWR@htmltag{/ol}}
3346 \newcommand*\LWR@printclosedescription}
3347     {\LWR@htmltag{/dl}}

3348 \end{warpHTML}

```

36 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 37 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: 3349 `\begin{warpHTML}`

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

```

3350 \newcounter{LWR@spandepth}
3351 \setcounter{LWR@spandepth}{0}

```

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

```
3352 \newbool{LWR@doingstartpars}
3353 \boolfalse{LWR@doingstartpars}
```

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

```
3354 \newbool{LWR@doingapar}
3355 \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:

```
3356 \newcommand*{\LWR@ensuredoingapar}{%
3357 \ifbool{LWR@doingstartpars}%
3358 {\global\booltrue{LWR@doingapar}}%
3359 {}%
3360 }
```

\LWR@openparagraph

```
3361 \newcommand*{\LWR@openparagraph}
3362 {%
```

See if paragraph handling is enabled:

```
3363 \ifbool{LWR@doingstartpars}%
3364 {% handling pars
```

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

```
3365 \ifthenelse{%
3366 \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3367 \cnttest{\value{LWR@spandepth}}{>}{0}%
3368 }% 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 .

```
3369 {}% no nested par tags
```

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

```
3370 {% yes nest par tags
3371 \LWR@htmltagc{\LWR@tagregularparagraph}%
```

Manually indent item list labels to avoid left margin intrusion:

See if are nested inside an item list:

```
3372 \ifnumcomp{\@listdepth}{>}{0}%
3373 {%
```

If so, leave some horizontal room in the L^AT_EX PDF output for list labels:

```
3374 \LWR@orighspace{1in}%
3375 }{}%
```

Now have started a paragraph.

```
3376 \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.)

```
3377 \let\par\LWR@closeparagraph%
3378 }% end of yes nest par tags
3379 }% end of handling pars
3380 {}% not handling pars
3381 }
```

\LWR@closeparagraph

```
3382 \newcommand*{\LWR@closeparagraph}
3383 {%
```

See if paragraph handling is enabled:

```
3384 \ifbool{LWR@doingapar}%
```

If currently in paragraph mode:

```
3385 {% handling pars
```

See if already started a lateximage or a :

```
3386 \ifthenelse{%
3387 \cinttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3388 \cinttest{\value{LWR@spandepth}}{>}{0}%
3389 }%
```

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

```

3390 {% no nested par tags
3391 \ifthenelse{%
3392 \cnttest{\value{LWR@spandepth}}{>}{0}%
3393 \AND%
3394 \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3395 }%
3396 {%
3397 \ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}%
3398 }%
3399 {}%
3400 }% no nested par tags

    If have not already started a lateximage or a <span>:

3401 {% yes nest par tags

    Print a closing tag:

3402 \unskip%
3403 \LWR@htmltagc{/LWR@tagregularparagraph}%

    No longer doing a paragraph:

3404 \global\boolfalse{LWR@doingapar}%
3405 % Disable the special \env{minipage} \& \cs{hspace} interaction
3406 % until a new minipage is found:
3407 %   \begin{macrocode}
3408 \global\boolfalse{LWR@minipagethispar}%
3409 }% end of yes nest par tags
3410 }% end of handling pars

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

3411 {% not handling pars
3412 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
3413 {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
3414 {}%
3415 }% not handling pars

    Finish with regular paragraph processing

3416 \LWR@origpar%
3417 }

3418 \end{warpHTML}

```

37 Paragraph start/stop handling

These commands allow/disallow the generation of HTML paragraph tags.

Section 36 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: 3419 `\begin{warphTML}`

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

```
3420 \newcommand*{\LWR@startpars}%
3421 {%
```

Ignore if inside a span:

```
3422 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
3423 {}%
3424 {%
```

See if currently handling HTML paragraphs:

```
3425 \ifbool{LWR@doingstartpars}%
```

If already in paragraph mode, do nothing.

```
3426 {}%
```

If not currently in paragraph mode:

```
3427 {%
```

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

```
3428 \PushPreHook{par}{\LWR@openparagraph}%
```

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

```
3429 \let\par\LWR@closeparagraph
3430
3431 }% an intentionally blank line
```

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

```
3432 \global\setbool{LWR@doingstartpars}{true}%
```

No <par> tag yet to undo:

```
3433 \global\boolfalse{LWR@doingapar}%
3434 }% nestspan
3435 }
```

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

```
3436 \newcommand*{\LWR@stoppars}%
3437 {%
```

Ignore if inside a span:

```
3438 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
3439 {}%
3440 {%
```

See if currently handling HTML paragraphs:

```
3441 \ifbool{LWR@doingapar}%
```

if currently in an HTML paragraph:

```
3442 {%
```

Print a closing tag:

```
3443 \unskip%
3444 \LWR@htmltagc{/\LWR@tagregularparagraph}%
```

No longer have an open HTML paragraph:

```
3445 \global\boolfalse{LWR@doingapar}%
```

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

```
3446 \global\boolfalse{LWR@minipagethispar}
3447
3448 }% an intentionally blank line
```

If was not in an HTML paragraph:

```
3449 {}%
```

See if currently allowing HTML paragraphs:

```
3450 \ifbool{LWR@doingstartpars}%
```

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

```

3451 {%
3452 \ClearPreHook{par}%
3453 }%

Else: do nothing

3454 {}%

no longer in paragraph mode

3455 \global\setbool{LWR@doingstartpars}{false}%

no <p> tag to undo:

3456 \global\boolfalse{LWR@doingapar}%
3457 }% nestspan
3458 }

3459 \end{warpHTML}

```

38 Page headers and footers

for HTML & PRINT: 3460 \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.

```

3461 \newcommand{\LWR@firstpagetop}{} % for the home page alone
3462 \newcommand{\LWR@pagetop}{} % for all other pages
3463 \newcommand{\LWR@pagebottom}{}
3464
3465 \newcommand{\LWR@setfirstpagetop}[1]{%
3466 \renewcommand{\LWR@firstpagetop}{#1}
3467 \catcode'\_ =8
3468 }

```

\HTMLFirstPageTop *{(text and logos)}*

```

3469 \newcommand{\HTMLFirstPageTop}{%
3470 \catcode'\_ =12
3471 \LWR@setfirstpagetop
3472 }

```

```

3473 \newcommand{\LWR@setpagetop}[1]{%
3474 \renewcommand{\LWR@pagetop}{#1}
3475 \catcode'\_ =8
3476 }

```

`\HTMLPageTop` $\{ \langle \textit>text and logos \rangle \}$

```

3477 \newcommand{\HTMLPageTop}{%
3478 \catcode'\_ =12
3479 \LWR@setpagetop
3480 }

```

```

3481 \newcommand{\LWR@setpagebottom}[1]{%
3482 \renewcommand{\LWR@pagebottom}{#1}
3483 \catcode'\_ =8
3484 }

```

`\HTMLPageBottom` $\{ \langle \textit>text and logos \rangle \}$

```

3485 \newcommand{\HTMLPageBottom}{%
3486 \catcode'\_ =12
3487 \LWR@setpagebottom
3488 }

```

```

3489 \end{warpall}

```

39 CSS

for HTML output: 3490 `\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.

```

3491 \newcommand*{\LWR@currentcss}{lwarp.css}

```

`\CSSFilename` $\{ \langle \textit>new-css-filename.css \rangle \}$ Assigns the CSS file to be used by the following HTML pages.

```

3492 \newcommand*{\LWR@newcssb}[1]{%
3493 \renewcommand*{\LWR@currentcss}{#1}
3494 \catcode'\_ =8
3495 }
3496

```

```

3497 \newcommand*{\CSSFilename}{
3498 \catcode'\_ =12
3499 \LWR@newcssb
3500 }
3501 \end{warpHTML}

```

for PRINT output:

```

3502 \begin{warpprint}
3503 \newcommand*{\CSSFilename}[1]{}
3504 \end{warpprint}

```

40 HTML meta description and author

for HTML & PRINT: 3505 \begin{warpall}

`\HTMLAuthor` $\{(\textit{authorname})\}$ The author to place into an HTML meta tag.

```

3506 \providecommand{\theauthor}{}
3507 \newcommand{\theHTMLAuthor}{\theauthor}
3508
3509 \newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}
3510 \end{warpall}

```

for HTML & PRINT: 3511 \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` $\{(\textit{New HTML meta description.})\}$ Assigns the HTML file's description meta tag.

```

3512 \newcommand{\LWR@currentHTMLDescription}{}
3513
3514 \newcommand{\HTMLDescription}[1]{%
3515 \renewcommand{\LWR@currentHTMLDescription}{#1}
3516 }
3517
3518 \end{warpall}

```

41 Footnotes

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

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.

41.1 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 [41.4](#) for the implementation.

41.2 Minipage footnotes

See section [66.2](#) for minipage footnotes.

41.3 Titlepage thanks

See section [48.6](#) for titlepage footnotes.

41.4 Regular page footnote implementation

for HTML output: 3519 `\begin{warpHTML}`

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

```
3520 \newbox\LWR@footnotes
```

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

```
3521 \long\def\@makefntext#1{\textsuperscript{\@thefnmark} #1}
3522
3523 \def\@makefnmark{\hbox{\textsuperscript{\@thefnmark}}}
```

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

```

3524 \long\def\@footnotetext#1{%
3525 \global\setbox\LWR@footnotes=\vbox{\unvbox\LWR@footnotes%
3526 \protected@edef\@currentlabel{%
3527 \csname p@footnote\endcsname\@thefnmark%
3528 }% \@currentlabel
3529 \color@begingroup%
3530 \ifbool{LWR@doingstartpars}{\LWR@htmltagc{\LWR@tagregularparagraph}}%
3531 \@makefntext{#1}%
3532 \ifbool{LWR@doingstartpars}{\par}{\LWR@htmltagc{/\LWR@tagregularparagraph}}%
3533 \color@endgroup%
3534 }% vbox
3535 }%
3536
3537 \long\def\@mpfootnotetext#1{%
3538 \global\setbox\@mpfootins\vbox{%
3539 \unvbox\@mpfootins
3540 \reset@font\footnotesize
3541 \hsize\columnwidth
3542 \@parboxrestore
3543 \protected@edef\@currentlabel
3544 {\csname p@mpfootnote\endcsname\@thefnmark}%
3545 \color@begingroup
3546 \@makefntext{%
3547 \ignorespaces#1%
3548 }%

```

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

```

3549 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
3550 {}%
3551 {\LWR@htmltagc{/\LWR@tagregularparagraph}}%
3552 \color@endgroup}%
3553 }

```

Enclose the footnotes in a class, print, then clear:

```

3554 \newcommand*\LWR@printpendingfootnotes}{%
3555 \ifvoid\LWR@footnotes\else
3556 \LWR@forcenewpage
3557 \begin{BlockClass}{footnotes}
3558 \LWR@origmedskip
3559 \unvbox\LWR@footnotes
3560 \setbox\LWR@footnotes=\vbox{}
3561 \end{BlockClass}
3562 \fi

```

```
3563 }
```

Used to print footnotes before sections only if formatting for an EPUB or word processor:

```
3564 \newcommand*\LWR@epubprintpendingfootnotes}{%
3565 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}{%
3566 {\LWR@printpendingfootnotes}%
3567 {}}%
3568 }

3569 \end{warpHTML}
```

42 Marginpars

for HTML output: 3570 \begin{warpHTML}

```
\marginpar [left] {right}
```

```
3571 \renewcommand{\marginpar}[2] []{%
3572 \LWR@htmlspanclass{marginpar}{#2}%
3573 }

3574 \end{warpHTML}
```

43 Splitting HTML files

- Files are split according to `FileDepth` and `CombineHigherDepths`.
- Filenames are sanitized by `\LWR@filenameno blanks`.
- `\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: 3575 \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.

```
3576 \newcounter{FileDepth}
3577 \setcounter{FileDepth}{-5}
```

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

```
3578 \newbool{CombineHigherDepths}
3579 \booltrue{CombineHigherDepths}

3580 \end{warpall}
```

for HTML output: 3581 `\begin{warpHTML}`

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

```
3582 \newcommand*{\LWR@thisfilename}{}
```

`\LWR@thisnewfilename` The filename being sanitized.

```
3583 \newcommand*{\LWR@thisnewfilename}{}
```

`\LWR@filenamoblanks` `{\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` L^AT_EX commands which appear in section names and TOC captions.

```
3584 \newcommand*{\LWR@filenamoblanks}[1]{%
3585 \begingroup
```

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

```
3586 \LWR@nullfonts
3587 \renewcommand*{\LWR@htmltagc}[1]{%
```

Replaces common symbols and short words with hyphens:

```
3588 \edef\LWR@thisnewfilename{#1}
3589 \fullexpandarg
```

Convert spaces into hyphens:

```
3590 \StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctutation into hyphens:

```
3591 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
```

```

3592 \StrSubstitute{\LWR@thisnewfilename}'{-}[\LWR@thisnewfilename]
3593 \StrSubstitute{\LWR@thisnewfilename}%
3594 {\LWR@origampersand}{-}[\LWR@thisnewfilename]
3595 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
3596 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
3597 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
3598 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
3599 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
3600 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
3601 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
3602 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
3603 \StrSubstitute{\LWR@thisnewfilename}"{-}[\LWR@thisnewfilename]
3604 \StrSubstitute{\LWR@thisnewfilename}%
3605 {\textless}{-}[\LWR@thisnewfilename]
3606 \StrSubstitute{\LWR@thisnewfilename}%
3607 {\textgreater}{-}[\LWR@thisnewfilename]
3608 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
3609 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
3610 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
3611 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
3612 \StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
3613 \StrSubstitute{\LWR@thisnewfilename}%
3614 {\textbackslash}{-}[\LWR@thisnewfilename]
3615 \StrSubstitute{\LWR@thisnewfilename}{^}{-}[\LWR@thisnewfilename]
3616 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
3617 %      "~{}" for babel
3618 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
3619 \StrSubstitute{\LWR@thisnewfilename}{]}{-}[\LWR@thisnewfilename]
3620 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]

```

Convert short words:

```

3621 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
3622 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
3623 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
3624 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
3625 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
3626 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
3627 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
3628 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
3629 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
3630 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
3631 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
3632 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
3633 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
3634 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
3635 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
3636 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
3637 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
3638 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

3639 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
3640 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
3641 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
3642 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
3643 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
3644 %      emdash
3645 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
3646 %      emdash
3647 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
3648 \endgroup
3649 }

```

`\LWR@newhtmlfile` $\{ \langle section\ name \rangle \}$

Finishes the current HTML page with footnotes, footer, navigation, then starts a new HTML page with an HTML comment telling where to split the page and what the new filename and CSS are, then adds navigation, side TOC, header, and starts the text body.

```
3650 \newcommand*{\LWR@newhtmlfile}[1]{
```

At the bottom of the ending file:

```

3651 \LWR@htmlend{section}{tbody}
3652
3653 \LWR@printpendingfootnotes
3654

```

No footer between files if EPUB:

```

3655 \ifbool{FormatEpub}
3656 {}
3657 {
3658 \LWR@htmlend{footer}
3659
3660 \LWR@pagebottom
3661
3662 \LWR@htmlend{footer}
3663 }

```

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

```

3664 \ifthenelse{\boolean{FormatEpub}\OR\boolean{FormatWordProcessor}}
3665 {}
3666 {\ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}}

```

End of this HTML file:

```
3667 \LWR@stoppars
3668 \LWR<htmltag{/body}>\LWR<orignewline
3669 \LWR<htmltag{/html}>\LWR<orignewline
3670 \LWR<orignewpage
3671
3672 \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.

```
3673 \ifbool{FileSectionNames}%
3674 {\LWR<filenamenooblanks{#1}}
3675 {\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.

```
3676 \LWR<htmlblockcomment{%
3677 |Start file|%
3678 \LWR<htmlsectionfilename{\LWR<thisfilename>}|%
3679 }
```

At the top of the starting file:

```
3680 \LWR@stoppars
3681
3682 \LWR<filestart{ -- #1}% there is an EMDash in front of the #1
3683
```

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

```
3684 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3685 {}
3686 {\LWR<topnavigation>}
3687
```

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

```
3688 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3689 {}
3690 {
3691 \LWR<html element{header}>
3692
```

```

3693 \LWR@pagetop
3694
3695 \LWR@html elementend{header}
3696 }
3697

```

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

```

3698 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3699 {}
3700 {\ifcvoid{thetitle}{}\LWR@printthetitle}
3701

```

No sideTOC if formatting for an EPUB or word processor:

```

3702 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3703 {}
3704 {\LWR@sidetoc}
3705

```

Start of the <textbody>:

```

3706 \LWR@html elementclass{section}{textbody}
3707

```

Keep paragraph tags disabled for now:

```

3708 \LWR@stoppars
3709

```

Track the page numbers:

```

3710 \setcounter{LWR@latestautopage}{\value{page}}
3711 }

```

```

3712 \end{warpHTML}

```

44 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing L^AT_EX 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 `pdflatex` to generate characters and `pdftotext` to read them. If extra symbols appear in the text, it may be that



pdf_latex is actually producing a symbol over or under a character, resulting in pdf_totext picking up the accent symbol separately.

X_lLa_TE_X and Lua_LA_TE_X directly support accented section and file names.

for HTML output: 3713 \begin{warpHTML}

44.1 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.



```
3714 \newbool{LWR@mainmatter}
3715 \DeclareDocumentCommand{\mainmatter}{}{%
3716 \booltrue{LWR@mainmatter}%
3717 }
```

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

```
3718 \DeclareDocumentCommand{\frontmatter}{}{%
3719 \boolfalse{LWR@mainmatter}%
3720 }
```

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

```
3721 \DeclareDocumentCommand{\backmatter}{}{%
3722 \boolfalse{LWR@mainmatter}
3723 }
```

44.2 Sectioning support macros

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

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

```
3724 \newcommand*{\LWR@sectionnumber}[1]{%
3725 \InlineClass{sectionnumber}{#1} %
3726 }
```

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

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

Create an autosection tag.

```
3727 \newcommand*{\LWR@createautosec}[1]{%
3728 \LWR@htmltag{#1 id="autosec-\thepage"{} }%
3729 }
```

`\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.

```
3730 \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.

```
3731 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
3732 \LWR@closeprevious{#1}%
```

Push a new section depth:

```
3733 \LWR@pushoneclose{#1}{#2}%
3734 }
```

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.

```
3735 \newcounter{LWR@prevFileDepth}
3736 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

`\LWR@section` * [$\langle TOC name \rangle$] $\langle name \rangle$ $\langle sectiontype \rangle$

The common actions for the high-level sectioning commands.

```
3737 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
3738 \LWR@traceinfo{LWR@section}%
3739 \LWR@stoppars%
```

Cancel special minipage horizontal space interaction:

```
3740 \global\boolfalse{LWR@minipagethispar}%
```

Start a new HTML file if not starred, and is a shallow sectioning depth:

```
3741 \LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
3742 \IfBooleanTF{#1}{% starred
```

Generate a new LaTeX page so that TOC and index page number points to the section:

```
3743 \LWR@orignewpage%
3744
3745 }{% not starred
3746 \ifthenelse{%
3747 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}%
3748 \AND%
3749 \(%
3750 \NOT\boolean{CombineHigherDepths}\OR%
3751 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}%
3752 \)%
3753 }%
```

If so: start a new HTML file:

```
3754 {% new file
3755 \LWR@traceinfo{LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
3756 \IfNoValueTF{#2}%
```

If no optional entry

```
3757 {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
3758 {\LWR@newhtmlfile{#2}}%
3759 }% new file
```

Else: No new HTML file:

```
3760 {% not new file
```

Generate a new LaTeX page so that TOC and index page number points to the section:

```

3761 \LWR@orignewpage%
3762
3763 }% not new file
3764 }% not starred

```

Remember this section's name for \nameref:

```

3765 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
3766 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%

```

Print an opening comment with the level and the name; ex: "section" "Introduction"

```

3767
3768 \ifbool{HTMLDebugComments}{%
3769 \LWR@htmlcomment{Opening #4 ‘‘#3’’}}
3770 }{}
3771

```

For inline sections paragraph and subparagraph, start a new paragraph now:

```

3772 \ifthenelse{%
3773 \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
3774 }%
3775 {\LWR@startpars}
3776 {}

```

Create the opening tag with an autosec:

```

3777 \LWR@createautosec{\csuse{LWR@tag#4}}%

```

If not starred, step counter and add to TOC:

```

3778 \IfBooleanTF{#1}%
3779 {}% starred
3780 {% not starred

```

Only add a numbered TOC entry if section number is not too deep:

```

3781 \ifthenelse{%
3782 \cnttest{\csuse{LWR@depth#4}}{<=}{{\value{secnumdepth}}}%
3783 }%
3784 {% if secnumdepth

```

If in the main matter, step the counter and add the TOC entry. For `article` class, `lwarp` assumes that all is mainmatter.

```

3785 \LWR@traceinfo{LWR@section: about to test main matter}%
3786 \ifbool{LWR@mainmatter}%
3787 {%

```

```
3788 \LWR@traceinfo{LWR@section: yes mainmatter}%
3789 \refstepcounter{#4}%
```

Add main matter numbered TOC entry with the TOC name or the regular name:

```
3790 \LWR@traceinfo{LWR@section: about to addcontentsline}%
3791 \addcontentsline{toc}{#4}%
3792 {\protect\LWR@sectionnumber{\csuse{the#4}}%
3793 \IfValueTF{#2}{#2}{#3}}%
3794 \LWR@traceinfo{LWR@section: finished addcontentsline}%
3795 }% end of if main matter
```

If not main matter, add unnumbered TOC name or regular name:

```
3796 {% not main matter
3797 \LWR@traceinfo{LWR@section: no main matter}%
3798 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
3799 }% end of not main matter
3800 }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```
3801 {%
3802 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
3803 }%
```

For part, print the section type:

```
3804 \ifbool{LWR@mainmatter}%
3805 {%
3806 \ifthenelse{%
3807 \(\cnttest{\csuse{LWR@depth#4}}{<=}%
3808 {\value{secnumdepth}}\)\ \AND
3809 \(\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\)
3810 }%
3811 {\csuse{#4name}~{}}%
3812 {}%
```

Print the section number:

```
3813 \LWR@traceinfo{LWR@section: about to print section number}%
3814 \ifthenelse{%
3815 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
3816 }%
3817 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
3818 {}%
3819 \LWR@traceinfo{LWR@section: finished print section number}%
3820 }{}%
3821 }% end of not starred
```

Print the section name:

```
3822 #3
```

close the heading tag, such as /H2

```
3823 \LWR@htmltag{\csuse{LWR@tag#4end}}%
```

Generate a L^AT_EX label:

```
3824 \label{autopage-\thepage}%
```

Start paragraph handing unless is an inline paragraph or subparagraph:

```
3825 \ifthenelse{%
3826 \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}}%
3827 {\LWR@startpars}%
3828 {}%
```

If not starred, remember the previous depth to possibly trigger a new HTML page.

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.

```
3829 \IfBooleanTF{#1}{}{% not starred
3830 \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
3831 }% not starred
3832 \LWR@traceinfo{LWR@section: done}%
3833 }
```

44.3 \section and friends

```
\part * [TOC name] {name}
```

```
3834 \@ifundefined{chapter}
3835 {}
3836 {%
3837 \DeclareDocumentCommand{\part}{s o m}{%
3838 \LWR@epubprintpendingfootnotes%
3839 \LWR@stoppars%
3840
3841 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
3842
```

```

3843 \LWR@section{#1}{#2}{#3}{part}%
3844 }
3845 }

```

`\chapter` * [*TOC name*] {*name*}

```

3846 \@ifundefined{chapter}
3847 {}
3848 {%
3849 \DeclareDocumentCommand{\chapter}{s o m}{%
3850 \LWR@traceinfo{chapter #3}%
3851 \LWR@epubprintpendingfootnotes%
3852 \LWR@stoppars%
3853
3854 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
3855
3856 \LWR@traceinfo{chapter: about to LWR@section}%
3857 \LWR@section{#1}{#2}{#3}{chapter}%
3858 \LWR@traceinfo{chapter: done}%
3859 }
3860 }

```

`\section` * [*TOC name*] {*name*}

```

3861 \DeclareDocumentCommand{\section}{s o m}{%
3862 \LWR@epubprintpendingfootnotes%
3863 \LWR@stoppars%
3864
3865 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
3866
3867 \LWR@section{#1}{#2}{#3}{section}%
3868 }

```

`\subsection` * [*TOC name*] {*name*}

```

3869 \DeclareDocumentCommand{\subsection}{s o m}{%
3870 \LWR@epubprintpendingfootnotes%
3871 \LWR@stoppars%
3872
3873 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
3874
3875 \LWR@section{#1}{#2}{#3}{subsection}%
3876 }

```

`\subsubsection` * [*TOC name*] {*name*}

```

3877 \DeclareDocumentCommand{\subsubsection}{s o m}{%

```

```

3878 \LWR@epubprintpendingfootnotes%
3879 \LWR@stoppars%
3880
3881 \LWR@startnewdepth{\LWR@depthsubsubsection}%
3882 {\LWR@printclosesubsubsection}%
3883
3884 \LWR@section{#1}{#2}{#3}{subsubsection}%
3885 }

```

`\paragraph` * [*TOC name*] {*name*}

```

3886 \DeclareDocumentCommand{\paragraph}{s o m}{%
3887 \LWR@epubprintpendingfootnotes%
3888 \LWR@stoppars%
3889
3890 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
3891
3892 \LWR@section{#1}{#2}{#3}{paragraph}%
3893 }

```

`\subparagraph` * [*TOC name*] {*name*}

```

3894 \DeclareDocumentCommand{\subparagraph}{s o m}{%
3895 \LWR@epubprintpendingfootnotes%
3896 \LWR@stoppars%
3897
3898 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
3899
3900 \LWR@section{#1}{#2}{#3}{subparagraph}%
3901 }

3902 \end{warpHTML}

```

45 Starting a new file

for HTML & PRINT: 3903 `\begin{warpall}`

`\HTMLLanguage` Default language for the HTML `lang` tag.

```

3904 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
3905
3906 \newcommand*{\HTMLLanguage}[1]{%
3907 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
3908 }

```

```
3909 \end{warpall}
```

for HTML output: 3910 \begin{warpHTML}

```
\LWR@filestart  {\langle title\_suffix \rangle}
```

Creates the opening HTML tags.

```
3911 \newcommand*{\LWR@filestart}[1]{
3912 \LWR@traceinfo{\LWR@filestart}
```

Locally temporarily disable direct-formatting commands:

```
3913 \begingroup
3914 \LWR@traceinfo{\LWR@filestart: A}
3915 \LWR@nullfonts
3916 \LWR@traceinfo{\LWR@filestart: B}
```

Create the page's HTML header:

```
3917 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
3918 \LWR@traceinfo{\LWR@filestart: C}
```

The language is user-adjustable:

```
3919 \LWR@htmltag{html lang="\LWR@currentHTMLLanguage"}\LWR@orignewline
```

Start of the meta data:

```
3920 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
3921 \LWR@htmltag{meta charset="UTF-8" /}\LWR@orignewline
```

Author:

```
3922 \ifcsemt{theHTMLAuthor}{-}{
3923 \LWR@htmltag{meta name="author" content="\theHTMLAuthor" /}\LWR@orignewline
3924 }
```

lwarp is the generator:

```
3925 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /}%
3926 \LWR@orignewline
```

If there is a description, add it now:

```

3927 \ifdefempty{\LWR@currentHTMLDescription}{-}{-%
3928 \LWR@htmltag{%
3929 meta name="description" content="\LWR@currentHTMLDescription" /}%
3930   \LWR@orignewline
3931 }%

```

Mobile-friendly viewport:

```

3932 \LWR@htmltag{meta name="viewport" %
3933 content="width=device-width, initial-scale=1.0" /}%
3934   \LWR@orignewline

```

IE patch:

```

3935 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
3936 \LWR@htmltag{%
3937 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{-}}%
3938 \LWR@htmltag{/script}\LWR@orignewline
3939 \LWR@htmltag{![endif]{-}{-}}\LWR@orignewline

```

The page's title:

```

3940 \ifcvoid{thetitle}{-}{-%
3941 \LWR@htmltag{title}\thetitle#1\LWR@htmltag{/title}\LWR@orignewline%
3942 }%

```

The page's stylesheet:

```

3943 \LWR@htmltag{%
3944 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
3945 \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.

```

3946 \ifbool{mathjax}%
3947 {%
3948 \boolfalse{LWR@verbtags}
3949 \verbatiminput{lwarp_mathjax.txt}%
3950 \booltrue{LWR@verbtags}
3951 \LWR@stoppars
3952 }% end of mathjax
3953 {}%

```

End of the header:

```

3954 \LWR@htmltag{/head}\LWR@orignewline

```

Start of the body:

```
3955 \LWR@htmltag{body}\LWR@orignewline
3956 \endgroup
3957 \LWR@traceinfo{LWR@filestart: done}
3958 }

3959 \end{warpHTML}
```

46 Starting HTML output

for HTML output: 3960 \begin{warpHTML}

`\LWR@LwarpStart` Executed at the beginning of the entire document.

```
3961 \catcode'\$=\active
3962 \newcommand*{\LWR@LwarpStart}
3963 {%
3964 \LWR@traceinfo{LWR@lwarpStart}
```

If formatting for a word processor, force filedepth to single-file only, force HTML debug comments off.

```
3965 \ifbool{FormatWordProcessor}{%
3966 \setcounter{FileDepth}{-5}%
3967 \boolfalse{HTMLDebugComments}%
3968 }{}
```

Expand and detokenize `\HomeHTMLFilename` and `\HTMLFilename`:

```
3969 \edef\LWR@strresult{\HomeHTMLFilename}
3970 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
3971 \edef\LWR@strresult{\HTMLFilename}
3972 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn and empty page style:

```
3973 \LWR@origonecolumn%
3974 \LWR@origpagestyle{empty}
```

Reduce chance of line overflow in verbatim environments:

```
3975 \LWR@origscriptsize%
```

In PDF output, don't allow line breaks to interfere with HTML tags:

```
3976 \LWR@origraggedright%
3977 \LetLtxMacro{\}\{\LWR@endoffline}%
```

Spread the lines for `pdftotext` to read them well:

```
3978 \linespread{1.3}%
```

For `pdftotext` to reliably identify paragraph splits:

```
3979 \setlength{\parindent}{0pt}
3980 \setlength{\parskip}{2ex}
```

For the `lateximages` record file:

```
3981 \immediate\openout\LWR@file=lateximages.txt
```

Removes space after the caption in the HTML:

```
3982 \setlength{\belowcaptionskip}{-3ex}
```

Redefine the plain page style to be empty when used by index pages:

```
3983 \renewcommand{\ps@plain}{} 
```

```
\centering Not used in the HTML environment:
\raggedleft
\raggedright 3984 \renewcommand*\centering{}
3985 \renewcommand*\raggedleft{}
3986 \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:

```
3987 \LetLtxMacro{\LWR@origtabular}{\tabular}
3988 \LetLtxMacro{\LWR@origendtabular}{\endtabular}
3989 \LetLtxMacro{\tabular}{\LWR@tabular}
3990 \LetLtxMacro{\endtabular}{\endLWR@tabular}
```

Float captions:

```
3991 \let\LWR@origcaption\caption
```

Labels: `\ltx@label` is used in `amsmath` environments and is also patched by [Label in HTML](#) `cleveref`.

```
3992 \let\LWR@origltx@label\ltx@label
3993 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
3994 \let\TeX\LWR@TeX
3995 \let\LaTeX\LWR@LaTeX
3996 \let\LuaTeX\LWR@LuaTeX
3997 \let\LuaLaTeX\LWR@LuaLaTeX
3998 \let\XeTeX\LWR@XeTeX
3999 \let\XeLaTeX\LWR@XeLaTeX
4000 \let\ConTeXt\LWR@ConTeXt
```

Graphics:

```
4001 \LetLtxMacro{\rotatebox}{\LWR@rotatebox}
4002 \LetLtxMacro{\scalebox}{\LWR@scalebox}
4003 \let\reflectbox\LWR@reflectbox
```

Not yet started any paragraph handling:

```
4004 \global\boolfalse{LWR@doingapar}
4005 \global\boolfalse{LWR@doingstartpars}
```

Start a new HTML file and a header:

```
4006 \LWR@traceinfo{LWR@lwarpStart: Starting new file.}
4007 \LWR@filestart{}
4008 \LWR@traceinfo{LWR@lwarpStart: Generating first header.}
4009 \LWR@htmltag{header}\LWR@orignewline
4010 \LWR@startpars
4011 \LWR@firstpagetop
4012 \LWR@stoppars
4013 \LWR@htmltag{/header}\LWR@orignewline
4014 \LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
4015 \LWR@htmltag{section class="textbody"{} }
```

Document and page settings:

```
4016 \mainmatter
4017 \LWR@origpagenumbering{arabic}
```

Set default titlepage thanks footnote marks. See section [48.6](#).

```
4018 \if@titlepage
4019   \thanksmarkseries{arabic}
4020 \else
4021   \thanksmarkseries{fnsymbol}
4022 \fi
```

Patch the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native L^AT_EX environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```
4023 \LWR@patchlists
```

Ensure that math mode is active to call lwarp's patches:

```
4024 \catcode'\$=\active
```

Allow HTML paragraphs to begin:

```
4025 \LWR@startpars
```

```
4026 \LWR@traceinfo{LWR@lwarpStart: done}
```

```
4027 }
```

```
4028 \catcode'\$=3% math shift until lwarp starts
```

```
4029 \end{warpHTML}
```

47 Ending HTML output

for HTML output: 4030 `\begin{warpHTML}`

`\LWR@requesttoc` $\langle\{boolean\}\rangle$ $\langle\{suffix\}\rangle$ Requests that a toc, lof, or lot be generated.

```
4031 \newcommand*\LWR@requesttoc}[2]{%
```

```
4032 \ifbool{#1}
```

```
4033 {
```

```
4034 \expandafter\newwrite\csuse{tf@#2}
```

```
4035 \immediate\openout \csuse{tf@#2} \jobname.#2\relax
```

```
4036 }{}
```

```
4037 }
```

`\LWR@LwarpEnd` Final stop of all HTML output:

```
4038 \newcommand*\LWR@LwarpEnd
```

```
4039 {
```

```
4040 \LWR@stoppars
```

```
4041 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
4042 \LWR@htmlElementclassend{section}{textbody}
```

Print any pending footnotes:

```
4043 \LWR@printpendingfootnotes
```

Create the footer:

```
4044 \LWR@html element{footer}  
4045  
4046 \LWR@pagebottom  
4047  
4048 \LWR@html elementend{footer}
```

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

Presumably has a table-of-contents.

```
4049 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}  
4050 {  
4051 {  
4052 \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{  
4053 }
```

```
4054 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
4055 \LWR@htmltag{/body}\LWR@orignewline  
4056 \LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
4057 \LWR@orignewpage
```

For lateximage commands:

```
4058 \immediate\closeout\LWR@file  
4059 }
```

```
4060 \end{warpHTML}
```

48 Titles and the titling package

Supports and extends the titling package.

Additional functions include `\published` and `\subtitle`, and the `\author` command has an additional `\affiliation` command to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using `\theauthor` in the main text.

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.

`\printtitle`, `\printauthor`, etc., are provided for use inside the `titlepage` or `titlingpage` environments, and these retain the `\thanks` and `\affiliation`.

Several additional hooks are provided in addition to `titling`:

<code>\maketitlehookaa</code>	<code>\maketitlehookaa</code> : Between “published” and the title.
<code>\maketitlehookaaa</code>	<code>\maketitlehookaaa</code> : Between the title and the subtitle.
<code>\prepublished</code>	<code>\prepublished</code> : Before the “published” field.
<code>\postpublished</code>	<code>\postpublished</code> : After the “published” field.
<code>\presubtitle</code>	<code>\presubtitle</code> : Before the subtitle.
<code>\postsubtitle</code>	<code>\postsubtitle</code> : After the subtitle.
<code>\printthanks</code>	<code>\printthanks</code> has been added to force the printing of thanks inside a <code>titlingpage</code> environment when <code>\maketitle</code> is not used.

 **No footnotes!**

Inside a `\titlepage` or `\titlingpage` environment, use `\thanks` for footnotes, do not use `\footnote`.

At the end of the `titlingpage` environment, footnote marks are forced to reset to zero.

Inside a `titlingpage` environment with the `article` document class, thanks marks will be `fnsymbol` instead of `arabic`. `arabic` is usually used when inside `titlepage` environments where the title page is on its own page, but is not automatically used inside a `titlingpage` environment.

To force the thanks marks to be `arabic`:

```
\begin{titlingpage}
\thanksmarkseries{arabic}
...
```

48.1 Setting the title, etc.

The following provide setting commands for both HTML and print outputs.

<code>\@published</code>	<code>\@title</code> , <code>\@subtitle</code> , <code>\@author</code> , etc. store the values as originally assigned, including any <code>\thanks</code> , <code>\and</code> , or <code>\affiliation</code> . These are low-level macros intended to be used by other macros only inside a <code>titlepage</code> or <code>titlingpage</code> , and are used by <code>\maketitle</code> . The author is printed inside a single-column table, which becomes multiple single-column tables if multiples authors are included.
<code>\@title</code>	
<code>\@subtitle</code>	
<code>\@author</code>	
<code>\@date</code>	
<code>\printpublished</code>	<code>\printtitle</code> , <code>\printsubtitle</code> , etc. are user-level macros intended to be used in <code>titlepage</code> and <code>titlingpage</code> environments in cases where <code>\maketitle</code> is not desired. These commands preserve the <code>\thanks</code> , etc., and should not be used in the main text. The author is printed inside a single-column table, which becomes multiple single-column tables if multiples authors are included.
<code>\printtitle</code>	
<code>\printsubtitle</code>	
<code>\printauthor</code>	
<code>\printdate</code>	
<code>\thepublished</code>	<code>\thetitle</code> , <code>\thesubtitle</code> , <code>\theauthor</code> , etc. are user-level sanitized versions which have removed the <code>\thanks</code> and <code>\affiliation</code> , and <code>\and</code> is changed for inline text usage. The author is printed inline without <code>\affiliation</code> or <code>\thanks</code> , with <code>\and</code> 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:
<code>\thetitle</code>	
<code>\thesubtitle</code>	
<code>\theauthor</code>	
<code>\thedate</code>	
<code>\HTMLPageBottom</code>	<code>{\text}</code>

```

\HTMLPageBottom{
\begin{center}\textcopyright~2016 \theauthor\end{center}
}

```

`\author` `{\author}` While using `\maketitle`, the author is treated as a single-column table and the `\and` feature finishes the current table then starts a new one for the next author. Each author thus is placed into its own table, and an affiliation may be placed on its own line such as

```
\author{Name \\ Affiliation \and Second Name \\ Second Affiliation}
```

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.

A solution, provide here, is to define a macro `\affiliation` which during `\maketitle` starts a new table row and adds the affiliation, but after `\maketitle` is finished `\affiliation` is re-defined to throw away its argument, thus printing only the author names when `\author` is later used inline.

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

Adds the affiliation to the author for use in `\maketitle`. Nullified when later used for inline use of `\theauthor`.

for HTML output: 4061 `\begin{warpHTML}`
 4062 `\newrobustcmd{\affiliation}[1]{\ \ \ \InlineClass{affiliation}{#1}}`
 4063 `\end{warpHTML}`

for PRINT output: 4064 `\begin{warpprint}`
 4065 `\newrobustcmd{\affiliation}[1]{\ \ \ \textsc{small#1}}`
 4066 `\end{warpprint}`

The following are based on the original titling code:

for HTML & PRINT: 4067 `\begin{warpall}`

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

Redefined to nullify `\affiliation`, etc. before printing the authors inline.

`\@author` retains the entire author with its `\thanks`, while `\theauthor` will have `\thanks` removed and `\and` simplified.

```
4068 \renewcommand{\author}[1]{%
4069 \gdef\@author{#1}
4070 \begingroup
4071 \renewcommand{\thanks}[1]{ }
4072 \renewcommand{\and}{\unskip, }
4073 \renewcommand{\thanksmark}[1]{ }
4074 \renewcommand{\thanksgap}[1]{ }
4075 \renewcommand{\affiliation}[1]{ }
4076 \protected@xdef\theauthor{#1}
4077 \endgroup}
```

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

```
4078 \newcommand{\published}[1]{%
4079 \gdef\@published{#1}
4080 \begingroup
4081 \renewcommand{\thanks}[1]{ }
4082 \renewcommand{\thanksmark}[1]{ }
4083 \renewcommand{\thanksgap}[1]{ }
4084 \protected@xdef\thepublished{#1}
4085 \endgroup
4086 }
4087 \newcommand{\@published}{ }
4088 \newcommand{\thepublished}{ }
```

```

\subtitle {{text}}

4089 \newcommand{\subtitle}[1]{%
4090 \gdef\@subtitle{#1}
4091 \begingroup
4092 \renewcommand{\thanks}[1]{ }
4093 \renewcommand{\thanksmark}[1]{ }
4094 \renewcommand{\thanksgap}[1]{ }
4095 \protected@xdef\thesubtitle{#1}
4096 \endgroup
4097 }
4098 \newcommand{\@subtitle}{ }
4099 \newcommand{\thesubtitle}{ }

4100 \end{warpall}

```

48.2 Changes to HTML titlepage and titlingpage

for HTML output: 4101 \begin{warpHTML}

Env `titlepage` Sets up a titlepage div with a L^AT_EX PDF minipage inside.

```

4102 \renewenvironment*{titlepage}
4103 {
4104 \LWR@forcenewpage
4105 \BlockClass{titlepage}\LWR@subminipage
4106 }
4107 {\LWR@endsubminipage\endBlockClass}

```

Env `titlingpage`

```

4108 \renewenvironment*{titlingpage}
4109 {%

```

Start an HTML titlepage div:

```

4110 \begin{titlepage}

```

Prepare for a custom version of `\maketitle` inside the `titlingpage`:

```

4111 \LWR@maketitlesetup
4112 \let\maketitle\LWR@titlingmaketitle
4113 }
4114 {

```

At the end of the environment, end the HTML titlepage div:

```
4115 \end{titlepage}
```

Reset the footnote counter:

```
4116 \@bscontmark
4117 }
```

```
4118 \end{warpHTML}
```

for HTML & PRINT: 4119 \begin{warpall}

`\printthanks` Forces the `\thanks` to be printed.

This is necessary in a `titlingpage` environment when `\maketitle` was not used.

```
4120 \newcommand*\printthanks{\@thanks}
```

Env `titlingpage` At the end of the `titlingpage` for both print and HTML, reset footnote markers to zero.

```
4121 \AtEndEnvironment{titlingpage}{\@bscontmark}
```

```
4122 \end{warpall}
```

48.3 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: 4123 \begin{warpHTML}

Patch the pre/post title/author/date to add HTML tags, then initialize:

```
4124 \newcommand{\prepublished}[1]{%
4125 \def\@bsprepublished{\BlockClass{published}#1}%
4126 }
4127
4128 \newcommand{\postpublished}[1]{%
4129 \def\@bspostpublished{#1\endBlockClass}%
4130 }
4131
4132 \renewcommand{\pretitle}[1]{%
4133 \def\@bspretitle{#1\LWR@stoppars\LWR@htmltag{h1}}%
4134 }
```

```
4135
4136 \renewcommand{\posttitle}[1]{%
4137 \def\@bsposttitle{\LWR@htmltag{/h1}\LWR@startpars#1}%
4138 }
4139
4140 \newcommand{\presubtitle}[1]{%
4141 \def\@bspresubtitle{\BlockClass{subtitle}#1}%
4142 }
4143
4144 \newcommand{\postsubtitle}[1]{%
4145 \def\@bspostsubtitle{#1\endBlockClass}%
4146 }
4147
4148 \renewcommand{\preauthor}[1]{%
4149 \def\@bspreauthor{\BlockClass{author}#1}%
4150 }
4151
4152 \renewcommand{\postauthor}[1]{%
4153 \def\@bspostauthor{#1\endBlockClass}%
4154 }
4155
4156 \renewcommand{\predate}[1]{%
4157 \def\@bspredate{#1\BlockClass{titledate}}%
4158 }
4159
4160 \renewcommand{\postdate}[1]{%
4161 \def\@bspostdate{\endBlockClass#1}%
4162 }
4163
4164 \prepublished{\begin{center}}
4165 \postpublished{\par\end{center}}
4166
4167 \pretitle{\begin{center}}
4168 \posttitle{\par\end{center}}
4169
4170 \presubtitle{\begin{center}}
4171 \postsubtitle{\par\end{center}}
4172
4173 \preauthor{\begin{center}%
4174 \begin{tabular}[t]{c}%
4175 }
4176 \postauthor{\end{tabular}\par\end{center}}
4177
4178 \predate{\begin{center}}
4179 \postdate{\par\end{center}}
```

\printpublished

```
4180 \newcommand*\printpublished{
```

```
4181 \ifthenelse{\equal{\thepublished}{}}
4182 {}
4183 {
4184 \begin{BlockClass}{published}
4185 \@published
4186 \end{BlockClass}
4187 }
4188 }
```

`\printtitle`

```
4189 \newcommand*{\printtitle}
4190 {
4191 \LWR@stoppars
4192 \LWR@htmltag{h1}%
4193 \@title%
4194 \LWR@htmltag{/h1}
4195 \LWR@startpars
4196 }
```

`\LWR@printthetitle` A private version which prints the title without footnotes, used to title each HTML page.

```
4197 \newcommand*{\LWR@printthetitle}
4198 {
4199 \LWR@stoppars
4200 \LWR@htmltag{h1}%
4201 \thetitle%
4202 \LWR@htmltag{/h1}
4203 \LWR@startpars
4204 }
```

`\printsubtitle`

```
4205 \newcommand*{\printsubtitle}{
4206 \ifthenelse{\equal{\thesubtitle}{}}
4207 {}
4208 {
4209 \begin{BlockClass}{subtitle}
4210 \@subtitle
4211 \end{BlockClass}
4212 }
4213 }
```

`\printauthor`

```
4214 \newcommand*{\printauthor}{
```

```

4215 \begin{BlockClass}{author}
4216 \begin{tabular}{c}\@author\end{tabular}
4217 \end{BlockClass}
4218 }

```

`\printdate`

```

4219 \newcommand*\printdate{%
4220 \begin{BlockClass}{titledate}
4221 \@date
4222 \end{BlockClass}
4223 }

```

```
4224 \end{warpHTML}
```

48.4 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: 4225 `\begin{warpprint}`

`\printpublished`

```
4226 \newcommand*\printpublished{{\Large\scshape\@published}}
```

`\printtitle`

```
4227 \newcommand*\printtitle{{\Huge\@title}}
```

`\printsubtitle`

```
4228 \newcommand*\printsubtitle{{\Large\itshape\@subtitle\bigskip}}
```

`\printauthor`

```

4229 \newcommand*\printauthor
4230   {{\large\begin{tabular}{t}{c}\@author\end{tabular}}}

```

`\printdate`

```
4231 \newcommand*\printdate{{\small\textit{\@date}}}
```

48.5 \maketitle for print output

`\maketitle` From the titling package, patched to add the publisher and subtitle.

```

4232 \providecommand{\maketitle}{}
4233 \if@titlepage
4234   \renewcommand{\maketitle}{\begin{titlepage}%
4235     \let\footnotesize\small
4236     \let\footnoterule\relax
4237     \LetLtxMacro{\footnote}{\thanks}
4238     \@bsmarkseries
4239     \def\@makefnmark{\rlap{\@textsuperscript{%
4240       \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
4241     \long\def\@makefntext##1{\makethanksmark ##1}
4242     \null\vfil
4243     \vskip 60\p@
4244     \vspace*{\droptitle}
4245     \maketitlehooka
4246     \ifcsempy{@published}
4247   {}
4248   {{\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
4249   {\@bspretitle \@title \@bsposttitle}
4250   \ifcsempy{@subtitle}
4251 {}
4252 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
4253   \maketitlehookb
4254   {\@bspreauthor \@author \@bspostauthor}
4255   \maketitlehookc
4256   {\@bspredate \@date \@bspostdate}
4257   \maketitlehookd
4258   \par
4259   \@thanks
4260   \vfil\null
4261   \end{titlepage}%
4262   \@bscontmark % \setcounter{footnote}{0}%
4263   %% \bsmtitleempty
4264 } % end titlepage defs
4265 \else
4266 \renewcommand{\maketitle}{\par
4267 \begingroup
4268   \@bsmarkseries
4269   \def\@makefnmark{\rlap{\@textsuperscript{%
4270     \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
4271   \long\def\@makefntext##1{\makethanksmark ##1}
4272   \if@twocolumn
4273     \ifnum \col@number=\@ne
4274       \@maketitle
4275     \else
4276       \twocolumn[\@maketitle]%

```

```

4277     \fi
4278     \else
4279     \newpage
4280     \global\@topnum\z@
4281     \@maketitle
4282     \fi
4283     \thispagestyle{plain}\@thanks
4284 \endgroup
4285 \@bscontmark % \setcounter{footnote}{0}%
4286 %%% \@bsmtitleempty
4287 } % end non-titlepage
4288
4289 \def\@maketitle{%
4290     \newpage
4291     \null
4292     \vskip 2em%
4293     \vspace*{\droptitle}
4294     \maketitlehooka
4295     \ifcseempty{@published}
4296 {}
4297 {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
4298     {\@bspretitle \@title \@bsposttitle}
4299     \ifcseempty{@subtitle}
4300 {}
4301 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
4302     \maketitlehookb
4303     {\@bspreadauthor \@author \@bspostauthor}
4304     \maketitlehookc
4305     {\@bspredate \@date \@bspostdate}
4306     \maketitlehookd
4307     \par
4308     \vskip 1.5em}
4309 \fi
4310
4311 \providecommand{\maketitlehookaa}{}
4312
4313 \providecommand{\maketitlehookaaa}{}
4314
4315 \newcommand{\prepublished}[1]{%
4316 \def\@bsprepublished{#1}%
4317 }
4318
4319 \newcommand{\postpublished}[1]{%
4320 \def\@bspostpublished{#1}%
4321 }
4322
4323 \newcommand{\presubtitle}[1]{%
4324 \def\@bspresubtitle{#1}%
4325 }

```

`\presubtitle` Hook after printing the subtitle.

```

4326 \newcommand{\postsubtitle}[1]{%
4327 \def\bspostsubtitle{#1}%
4328 }

Initial settings:

4329 \if@titlepage
4330 \prepublished{
4331 \vspace*{-\baselineskip}\vspace*{-\medskipamount}\vspace*{-2em}
4332 \begin{center}}
4333 \postpublished{\par\end{center}\vskip 2em}
4334
4335 \presubtitle{\unskip\begin{center}\unskip}
4336 \postsubtitle{\par\end{center}\vskip 2em}
4337 \else
4338 \prepublished{\begin{center}}
4339 \postpublished{\par\end{center}\vskip 0.5em}
4340
4341 \presubtitle{\begin{center}\unskip}
4342 \postsubtitle{\par\end{center}\vskip 0.5em}
4343 \fi

4344 \end{warpprint}

```

48.6 `\maketitle` for HTML output

An HTML div of class `titlepage` is created, inside of which a L^AT_EX PDF minipage is generated (without HTML tags), allowing the `\thanks` footnotes to be generated immediately at the end of the title page during HTML output. This is desirable when a large table of contents immediately follows the title.

`\thanks` are a form of footnotes used in the title page. See section 41 for other kinds of footnotes.

See `\thanksmarkseries{series}`, below, to set the style of the footnote marks.

for HTML output: 4345 `\begin{warppHTML}`

`\LWR@maketitlesetup` Patches `\thanks` macros to use L^AT_EX minipage footnotes.

```
4346 \newcommand*\LWR@maketitlesetup{%
```

Select which kind of footnote marks to use:

```
4347 \@bsmarkseries
4348 \@mpbsmarkseries
```

Redefine the footnote mark:

```
4349 \def\@makefnmark{\textsuperscript{\thefootnote}}

\thefootnote ⇒ \nameuse{arabic}{footnote}, or
\thefootnote ⇒ \nameuse{fnsymbol}{footnote}
```

Redefine the footnote text:

```
4350 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
4351 \makethanksmark \LWR@origspace{1in}

\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒
\@thefnmark ⇒ \itshape a (or similar)
```

Print the text:

```
4352 ##1%
4353 }%
4354 }
```

`\@fnsymbol` $\{ \langle counter \rangle \}$

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used `\|` which was not being found by `pdftotext`.

```
4355 \def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
4356 \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLUnicode{2016}}\or
4357 **\or \HTMLentity{dagger}\HTMLentity{dagger} \or
4358 \HTMLentity{Dagger}\HTMLentity{Dagger} \else\@ctrerr\fi}
```

`\maketitle` Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

```
4359 \renewcommand*\maketitle}{%
```

An HTML titlepage div is used for all classes.

```
4360 \begin{titlepage}
```

Set up special patches:

```
4361 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4362 \@maketitle
```

Immediately generate any \thanks footnotes:

```
4363 \@thanks
```

Close the HTML titlepage div:

```
4364 \end{titlepage}
```

Reset the footnote counter:

```
4365 \@bscontmark
4366 }
```

`\@maketitle` Typesets the title, etc. for HTML:

```
4367 \DeclareDocumentCommand{\@maketitle}{-}{%
4368 \maketitlehooka
4369 \ifcempty{@published}
4370 {}
4371 {\@bsprepublished \@published \@bspublished}\maketitlehookaa}
4372 {\@bspretitle \@title \@bsposttitle}
4373 \ifcempty{@subtitle}
4374 {}
4375 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
4376 \maketitlehookb
4377 {\@bspreadauthor \@author \@bspostauthor}
4378 \maketitlehookc
4379 {\@bspredate \@date \@bspostdate}
4380 \maketitlehookd
4381 }

4382 \providecommand{\maketitlehookaa}{}
4383 \providecommand{\maketitlehookaaa}{}

```

`\LWR@titlingmaketitle` `\maketitle` for use inside an HTML titlingpage environment.

```
4384 \newcommand*\LWR@titlingmaketitle}{%
```

Typeset the title, etc:

```
4385 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
4386 \@thanks
4387 }
```

```
\thanksmarkseries {{series}}
```

Sets the type of footnote marks used by `\thanks`, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc. Modified to use the L^AT_EX PDF minipage which is included with the title page.

```
4388 \renewcommand{\thanksmarkseries}[1]{%
4389 \def\@mpbsmarkseries{%
4390 \renewcommand*\thempfootnote{\@nameuse{#1}{mpfootnote}}}%
4391 \def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
4392 }

4393 \end{warpHTML}
```

49 Abstract

The following code replaces the L^AT_EX default, and will itself be replaced later if the `abstract` package is loaded.

for HTML output: 4394 `\begin{warpHTML}`

```
\abstractname User-redefinable title for the abstract.
```

Also over-written by the `babel` package.

```
4395 \providecommand*\abstractname{Abstract}
```

Env `abstract`

```
4396 \DeclareDocumentEnvironment{abstract}{}
4397 {
4398 \LWR@forcenewpage
4399 \BlockClass{abstract}
4400 \BlockClassSingle{abstracttitle}{\abstractname}
4401 }
4402 {
4403 \endBlockClass
4404 }
```

```
4405 \end{warpHTML}
```

50 Quote and verse

50.1 Citations and attributions

`\attribution` for use inside quote, quotation, verse:

ex: `\attribution{author name} --- \citetitle{book name}`

for HTML output: 4406 `\begin{warpHTML}`
 4407 `\newcommand{\attribution}[1]{%`
 4408 `\InlineClass{attribution}{--\,#1}}% emdash`
 4409 `\end{warpHTML}`

for PRINT output: 4410 `\begin{warpprint}`
 4411 `\newcommand{\attribution}[1]{\textsc{---\,#1}}`
 4412 `\end{warpprint}`

`\citetitle` for use inside quote, quotation, verse:

for HTML output: 4413 `\begin{warpHTML}`
 4414 `\newcommand{\citetitle}[1]{%`
 4415 `\InlineClass{citetitle}{--\,#1}}% emdash`
 4416 `\end{warpHTML}`

for PRINT output: 4417 `\begin{warpprint}`
 4418 `\newcommand{\citetitle}[1]{\textsl{---\,#1}}`
 4419 `\end{warpprint}`

50.2 Quotes, quotations

for HTML output: 4420 `\begin{warpHTML}`

Env `quote`

```
4421 \renewenvironment*{quote}
4422 {
4423 \LWR@forcenewpage
4424 \LWR@htmlblocktag{blockquote}
4425 }
4426 {\LWR@htmlblocktag{/blockquote}}
```

```

4427
4428 \renewenvironment*{quotation}
4429 {
4430 \LWR@forcenewpage
4431 \LWR@htmlblocktag{blockquotation}
4432 }
4433 {\LWR@htmlblocktag{/blockquotation}}

4434 \end{warpHTML}

```

50.3 Verse

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

```

\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}

```

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

`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.

for HTML & PRINT: 4435 `\begin{warpall}`

The following lengths may be set in either print or HTML output, but are only used in HTML. This allows the user to set `\vleftskip` and `\leftmargini` for print output, and optionally select different values for HTML.

`Len \TMLvleftskip` Sets `\vleftskip` inside a `verse` environment in HTML.

```
4436 \newlength{\HTMLvleftskip}
4437 \setlength{\HTMLvleftskip}{1em}
```

Len `\TMLleftmargini` Sets `\leftmargini` inside a `verse` environment in HTML.

```
4438 \newlength{\HTMLleftmargini}
4439 \setlength{\HTMLleftmargini}{4.5em}

4440 \end{warpall}
```

51 Verbatim

for HTML & PRINT: 4441 `\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.

```
4442 \newlength{\VerbatimHTMLWidth}
4443 \setlength{\VerbatimHTMLWidth}{4in}
4444 \end{warpall}
```

for HTML output: 4445 `\begin{warpHTML}`

Bool `LWR@verbtags` Used to temporarily turn off verbatim tags while doing `\verbatiminput` in the HTML head.

```
4446 \newbool{LWR@verbtags}
4447 \booltrue{LWR@verbtags}
```

`\LWR@atbeginverbatim` [*style*] [*class*]

Encloses a verbatim environment with the given CSS class.

```
4448 \newcommand*{\LWR@atbeginverbatim}[2] []
4449 {%
```

Avoid excessive space between lines:

```
4450 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
4451 \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.

```
4452 \ifbool{LWR@verbtags}{%
4453 \LWR@htmltag{pre class="#2"
4454 \ifthenelse{\equal{#1}{}}{}{style="#1"}}%
4455 }% pre
4456 }{}%
```

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.

```
4457 \begingroup%
4458 \LWR@origttfamily%
```

Do not produce HTML tags for `\hspace` inside a `verse par`. Restore plain \LaTeX `\hspace` functionality:

```
4459 \letLtxMacro{\hspace}{\LWR@orighspace}%
4460 }
```

`\LWR@afterendverbatim` Finishes enclosing a `verbatim` environment.

```
4461 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the `pre`:

```
4462 \endgroup%
4463 \unskip%
```

At the end of the environment, close the `pre`:

```
4464 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}
4465
4466 }{}%
```

Resume regular paragraph handling:

```
4467 \LWR@startpars%
4468 }
```

`\verbatiminput` $\{ \langle filename \rangle \}$

Patch `\verbatiminput` to add HTML tags:

```
4469 \let\LWRV@origverbatim@input\verbatim@input
4470
```

```

4471 \renewcommand{\verbatim@input}[2]{%
4472 \ifbool{LWR@verbtags}{\LWR@forcenewpage}{}%
4473 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
4474 \LWRV@origverbatim@input{#1}{#2}%
4475 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
4476 }

```

Env `verbatim`

```

4477 \AfterEndPreamble{
4478 \LWR@traceinfo{Patching verbatim.}
4479 \AtBeginEnvironment{verbatim}{%
4480 \LWR@forcenewpage
4481 \LWR@atbeginverbatim{verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
4482 }
4483 \AfterEndEnvironment{verbatim}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
4484 }

4485 \end{warpHTML}

```

52 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 span of class `theoremlabel`.
- The contents are placed inside a div of class `theoremcontents`.

for HTML output: 4486 `\begin{warpHTML}`

`\@begintheorem` $\{\langle name \rangle\}$ $\{\langle number \rangle\}$

```

4487 \renewcommand{\@begintheorem}[2]{%
4488 \LWR@forcenewpage
4489 \BlockClass{theoremcontents}
4490 \InlineClass{theoremlabel}{#1\ #2\ }
4491 }

```

`\@opargbegintheorem` $\{\langle name \rangle\}$ $\{\langle number \rangle\}$ $\{\langle oparg \rangle\}$

```

4492 \renewcommand{\@opargbegintheorem}[3]{%
4493 \LWR@forcenewpage
4494 \BlockClass{theoremcontents}
4495 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
4496 }

```

`\@endtheorem`

```

4497 \renewcommand*\@endtheorem{%
4498 \endBlockClass% theoremcontents
4499 }

```

```

4500 \end{warpHTML}

```

53 Lists

The environments `itemize`, `enumerate`, and `description` are patched when `lwarp` is started. These patches support the standard L^AT_EX 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 L^AT_EX 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}

```

53.1 Itemize

for HTML output: 4501 `\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.

```
4502 \newcommand*\LWR@itemizeitem}{%
4503 \LWR@stoppars%
4504 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloseitemize{}}%
4505 \LWR@htmltag{li}%
4506 \LWR@startpars%
4507 \LWR@origitem%
4508 }
```

Env `itemize` [*⟨options⟩*]

```
4509 \newcommand*\LWR@itemizestart}{%
4510 \LWR@stoppars%
4511 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4512 \LWR@htmltag{ul style="list-style-type:none"}%
4513 \LWR@startpars%
4514 \let\item\LWR@itemizeitem%
4515 }
4516
4517 \newcommand*\LWR@itemizeend}{%
4518 \LWR@stoppars%
4519 \LWR@closeprevious{\LWR@depthlistitem}%
4520 \LWR@closeoneprevious{}}%
4521 \LWR@startpars%
4522 }
```

53.2 Enumerate

An HTML unordered list is used with customized L^AT_EX-generated labels.

Env `enumerate` [*⟨options⟩*]

```
4523 \newcommand*\LWR@enumeratestart}{%
4524 \LWR@stoppars%
4525 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4526 \LWR@htmltag{ul style="list-style-type:none"}%
4527 \LWR@startpars%
4528 \let\item\LWR@itemizeitem%
4529 }
4530
```

```

4531 \newcommand*{\LWR@enumerateend}{%
4532 \LWR@stoppars%
4533 \LWR@closeprevious{\LWR@depthlistitem}%
4534 \LWR@closeoneprevious{}%
4535 \LWR@startpars%
4536 }

```

53.3 Description

`\LWR@descitem` [*label*] Handles an `\item` inside a description.

```

4537 \newcommand*{\LWR@descitem}[1][]{%
4538 {%
4539 \LWR@stoppars%
4540 \LWR@setlatestname{#1}%
4541 \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.

```

4542 \LetLtxMacro{\hspace}{\LWR@nohspace}%

```

Process the original `\item` code:

```

4543 \LWR@origitem[]%

```

Restore `\hspace` for use in the item text:

```

4544 \LetLtxMacro{\hspace}{\LWR@hspace}%

```

Be sure the label doesn't print to the left of the rest of the file:

```

4545 \LWR@orighspace{1in}
4546 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
4547 \LWR@orignewline%
4548 \LWR@htmltag{dd}%
4549 \LWR@startpars%
4550 }

```

Env `description` [*options*]

```

4551 \newcommand*{\LWR@descriptionstart}{%
4552 \LWR@stoppars%
4553 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}%
4554 \LWR@htmltag{dl}%

```

```

4555 \LWR@startpars%
4556 \let\item\LWR@descitem%
4557 }
4558
4559 \newcommand*\LWR@descriptionend}{%
4560 \LWR@stoppars%
4561 \LWR@closeprevious{\LWR@depthlistitem}%
4562 \LWR@closeoneprevious{}}%
4563 \LWR@startpars%
4564 }

```

`\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 L^AT_EX environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```

4565 \newcommand*\LWR@patchlists}{%
4566 \LetLtxMacro{\LWR@origitem}{\item}
4567 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
4568 \AtEndEnvironment{itemize}{\LWR@itemizeend}
4569 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
4570 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
4571 \AtBeginEnvironment{description}{\LWR@descriptionstart}
4572 \AtEndEnvironment{description}{\LWR@descriptionend}
4573 }

4574 \end{warpHTML}

```

54 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

Limitations:

- When defining environments or macros which include `tabular` plus 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:

```

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

```

△ misplaced alignment
alignment tab character &

- column types

 - 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.
 - \newcolumntype is ignored; unknown column types are set to 1.
 - tabularx ignores the width, but X columns do produce paragraph columns or multicolumns.
-  \multirow & \multicolumn

 - Multirow and multicolumn cannot be used at the same time. (No rectangular holes wider than one column or taller than one row.)
-  \multirow

 - For multirow, insert \mrowcell into any empty multi-row cells. This will be a null function for the print output, and is a placeholder for parsing the table for HTML output.
- \multirow 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.
-  paragraphs

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

 - For longtable, place headings and footings which do not apply to HTML inside \warpprintonly{}
-  \warpprintonly

 - 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.
-  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 \\}
```

54.1 Token lookahead

Used by `\LWR@futurenonpacelet` to look at the next token.

for HTML output: 4575 `\begin{warpHTML}`

`\LWR@mynexttoken`

```
4576 \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`

```
4577 \def\LWR@futurenonpacelet#1{\def\LWR@cs{#1}%
4578 \afterassignment\LWR@fnslone\let\nexttoken= }
4579 \def\LWR@fnslone{\expandafter\futurelet\LWR@cs\LWR@fnsltwo}
4580 \def\LWR@fnsltwo{%
4581 \expandafter\ifx\LWR@cs\@sptoken\let\next=\LWR@fnslthree%
4582 \else\let\next=\nexttoken\fi\next}
4583 \def\LWR@fnslthree{\afterassignment\LWR@fnslone\let\next= }
```

`\LWR@getmynexttoken` Looks ahead and copies the next token into `\LWR@mynexttoken`.

```
4584 \newcommand*{\LWR@getmynexttoken}{%
4585 \LWR@traceinfo{\LWR@getmynexttoken}%
4586 % nothing must follow this next line
4587 \LWR@futurenonpacelet\LWR@mynexttoken\LWR@tabledatacolumnstag
4588 }
```

54.2 Booleans

Bool `LWR@startedrow` True if should print a row tag before this column.

```
4589 \newbool{LWR@startedrow}
4590 \boolfalse{LWR@startedrow}
```

Bool LWR@doinghline True if the next row will have an hline above it.

```
4591 \newbool{LWR@doinghline}
4592 \boolfalse{LWR@doinghline}
```

Bool LWR@doingtbrule True if the next row will have a top/bottom rule above it.

```
4593 \newbool{LWR@doingtbrule}
4594 \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.

```
4595 \newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty multi-row cell, and thus there is no data tag to close.

```
4596 \newbool{LWR@skippingmrowcell}
```

Bool LWR@skipatbang True if just finished a \multicolumn so should not print the trailing @ or ! columns.

```
4597 \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.

```
4598 \newbool{LWR@intabularmetadata}
4599 \boolfalse{LWR@intabularmetadata}
```

54.3 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>

\LWR@closetabledatacell If LWR@skippingmrowcell then there is no data tag to close. Otherwise, close any paragraphs, then close the data tag.

```
4600 \newcommand*{\LWR@closetabledatacell}{%
4601 \global\booltrue{LWR@intabularmetadata}%
4602 \ifbool{LWR@exitingtabular}{}%
4603 {% not exiting tabular
4604 \ifbool{LWR@skippingmrowcell}{}%
4605 {% not skippingmrowcell
```

Insert any < then any @ and ! column contents:

```
4606 \unskip%
4607 \LWR@getexparray{LWR@colafterspec}{\theLWR@tablecolspos}%
4608 % % \LWR@getexparray{LWR@colatspec}{\theLWR@tablecolspos}%
4609 % \LWR@printatbang{at}{\theLWR@tablecolspos}%
4610 % % \LWR@getexparray{LWR@colbangspec}{\theLWR@tablecolspos}%
4611 % \LWR@printatbang{bang}{\theLWR@tablecolspos}%
```

Close paragraphs:

```
4612 \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
4613 \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell. Skip the @ and ! cells if are closing a multicolumn cell.

```
4614 \leavevmode\unskip\LWR@htmltag{/td}\LWR@orignewline%
4615 \ifbool{LWR@skipatbang}%
4616 {\boolfalse{LWR@skipatbang}}%
4617 {%
4618 % \LWR@getexparray{LWR@colatspec}{\theLWR@tablecolspos}%
4619 \LWR@printatbang{at}{\theLWR@tablecolspos}%
4620 % \LWR@getexparray{LWR@colbangspec}{\theLWR@tablecolspos}%
4621 \LWR@printatbang{bang}{\theLWR@tablecolspos}%
4622 }% not skipping at or bang
4623 }% not skipping mrowcell
4624 }% not exiting tabular
4625 \global\boolfalse{LWR@skippingmrowcell}%
4626 }
```

LWR@tabulardepth tracks whether & is being used inside a tabular.

```
4627 \newcounter{LWR@tabulardepth}
4628 \setcounter{LWR@tabulardepth}{0}
4629
```

When not used inside a tabular, & performs its original function as recorded here (with catcode 4).

```
4630 \let\LWR@origampmacro&
4631 \end{warpHTML}
```

54.3.1 Localizing & catcodes

for HTML & PRINT: 4632 \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.

```
4633 \newcommand{\StartDefiningTabulars}{%
4634 \warpHTMLonly{\catcode'\&=\active}%
4635 }
```

`\EndDefiningTabulars` Place after defining something with `&` in it.

```
4636 \newcommand{\EndDefiningTabulars}{%
4637 \warpHTMLonly{\catcode'\&=4}%
4638 }
```

```
4639 \end{warpall}
```

54.3.2 Handling `&`

for HTML output: 4640 `\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.

```
4641 \newcommand*\LWR@tabularampersand{%
4642 \LWR@traceinfo{LWR@tabularampersand}%
4643 \ifthenelse{\cnttest{\value{LWR@tabulardepth}}{>}{0}}%
4644 {%
```

If not skipping a multirow cell, close the current data cell.

```
4645 \unskip%
4646 \LWR@closetabledatacell%
```

Move to the next column.

```
4647 \addtocounter{LWR@tablecolspos}{1}%
```

Look at the next token to decide multi or single column data tag.

```
4648 \LWR@getmynexttoken%
4649 }%
```

If not inside a tabular, performs the original action:

```
4650 {\LWR@origampmacro}%
4651 }
```

& 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.

54.4 Handling `\\`

Inside `tabular`, `\\` is redefined to `\LWR@tabularendofline`

Throws away options `\\[dim]` or `*`

```
\LWR@tabularendofline
```

```
4652 \NewDocumentCommand{\LWR@tabularendofline}{s o}
4653 {%
4654 \LWR@closetabledatacell%
```

Finish the previous row:

```
4655 \LWR@htmltag{/tr}\LWR@orignewline
4656 \global\booltrue{LWR@intabularmetadata}
```

Not yet started a table row:

```
4657 \global\boolfalse{LWR@startedrow}
```

Additional setup:

```
4658 \global\boolfalse{LWR@doinghline}%
4659 \global\boolfalse{LWR@doingtbrule}%
4660 \LWR@clearmidrules%
```

Start at first column:

```
4661 \setcounter{LWR@tablecolspos}{1}
```

Look at the next token to decide between single column data tag or a special case:

```
4662 \LWR@getmynexttoken%
4663 }
```

54.5 Variables

```
4664 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
4665 \newcommand*{\LWR@pposition}{}
4666 \newcommand*{\LWR@pleft}{}
4667 \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`.

```
4668 \newcommand*{\LWR@tablecolspec}{}

```

`\LWR@strresult` Holds the result of `Str` functions.

```
4669 \newcommand*{\LWR@strresult}{}
4670 \newcommand*{\LWR@strresulttwo}{}

```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

```
4671 \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.)

```
4672 \newcounter{LWR@tablecolwidth}

```

`Ctr` `LWR@tablecolspos` Where are currently looking into the table column specification. Index starts at 1.

```
4673 \newcounter{LWR@tablecolspos}

```

- Ctr `LWR@tabletotalcols` Holds the final number of table columns, not counting @ and ! columns.
- ```
4674 \newcounter{LWR@tabletotalcols}
```
- Ctr `LWR@tabletotalcolsnext` Holds the next column while parsing. Is one more than `LWR@tabletotalcols`.
- ```
4675 \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 28.
- `LWR@colbangspec` A data array of specifications for ! columns. The leftmost's index is `leftedge`, the others are counter values. See section 28.
- `LWR@colbeforespec` A data array of specifications for > columns.
- `LWR@colafterspec` A data array of specifications for < columns.

54.6 Parsing @, >, <, ! columns

Holds the parsed argument for @, >, <, or ! columns:

```
4676 \newcommand*{\LWR@colparameter}{}
```

`\LWR@parseatcolumn` Handles @{text} columns.

```
4677 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
4678 \LWR@traceinfo{at column}%
```

```
4679 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into `\LWR@strresult`, expanding once:

```
4680 \LWR@traceinfo{about to read the next token:}%
```

```
4681 \expandarg%
```

```
4682 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]
```

```
4683 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
4684 \LWR@traceinfo{have now read the next token}%
```

```
4685 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}
```

```
4686 {% left edge of the table:
```

```
4687 \LWR@traceinfo{at the left edge}%
```

```

4688 \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@colparameter}%
4689 \LWR@traceinfo{at the left edge: %
4690 \LWR@getexparray{LWR@colatspec}{leftedge}}%
4691 }%
4692 {% not at the left edge:
4693 \LWR@traceinfo{not at the left edge}%
4694 \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
4695 \LWR@traceinfo{at \theLWR@tabletotalcols: %
4696 \LWR@getexparray{LWR@colatspec}{\theLWR@tabletotalcols}}%
4697 }%
4698 \let\LWR@colparameter\relax%
4699 \booltrue{LWR@validtablecol}%
4700 }

```

`\LWR@parsebangcolumn` Handles `!{text}` columns.

```

4701 \newcommand*{\LWR@parsebangcolumn}{%

```

Move to the next token after the '!':

```

4702 \LWR@traceinfo{bang column}%
4703 \addtocounter{LWR@tablecolspos}{1}%

```

Read the next token into `\LWR@strresult`, expanding once:

```

4704 \LWR@traceinfo{about to read the next token:}%
4705 \expandarg%
4706 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]
4707 \fullexpandarg%

```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```

4708 \LWR@traceinfo{have now read the next token}%
4709 \ifthenelse{\cinttest{\value{LWR@tabletotalcols}}=0}
4710 {% left edge of the table:
4711 \LWR@traceinfo{at the left edge}%
4712 \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@colparameter}%
4713 }%
4714 {% not at the left edge:
4715 \LWR@traceinfo{not at the left edge}%
4716 \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
4717 \LWR@traceinfo{bang \theLWR@tabletotalcols: \LWR@colparameter!}%
4718 }%
4719 \let\LWR@colparameter\relax%
4720 \booltrue{LWR@validtablecol}%
4721 }

```

`\LWR@parsebeforecolumn` Handles `>{text}` columns.

```
4722 \newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
4723 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token, expanding once into `\LWR@strresult`:

```
4724 \expandarg%
```

```
4725 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[LWR@colparameter]%
```

```
4726 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
4727 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{LWR@colparameter}%
```

```
4728 \let\LWR@colparameter\relax%
```

```
4729 \booltrue{LWR@validtablecol}%
```

```
4730 }
```

`\LWR@parseaftercolumn` Handles `<{text}` columns.

```
4731 \newcommand*{\LWR@parseaftercolumn}{%
```

Move to the next token after the '<':

```
4732 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token, expanding once into `\LWR@strresult`:

```
4733 % \StrChar{#1}{\theLWR@tablecolspos}[LWR@colparameter]
```

```
4734 \expandarg%
```

```
4735 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[LWR@colparameter]%
```

```
4736 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
4737 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcols}{LWR@colparameter}%
```

```
4738 \let\LWR@colparameter\relax%
```

```
4739 \booltrue{LWR@validtablecol}%
```

```
4740 }
```

`\LWR@parseskipcolumn` Handles columns to skip, such as the vertical bar.

```
4741 \newcommand*{\LWR@parseskipcolumn}{%
```

```
4742 \booltrue{LWR@validtablecol}%
```

```
4743 }
```

54.7 Parsing ‘l’, ‘c’, or ‘r’ columns

`\LWR@parsenormalcolumn` $\{ \langle thiscolumn \rangle \}$

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```

4744 \newcommand*\LWR@parsenormalcolumn}[1]{%
4745 \appto\LWR@tablecolspec{#1}%
4746 \addtocounter{LWR@tabletotalcols}{1}%
4747 \addtocounter{LWR@tabletotalcolsnext}{1}%
4748 \LWR@traceinfo{normal column \theLWR@tabletotalcols: #1}%
4749 \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcolsnext}{}%
4750 \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcolsnext}{}%
4751 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{}%
4752 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcolsnext}{}%
4753 \booltrue{LWR@validtablecol}%
4754 }
```

54.8 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn` $\{ \langle thiscolumn \rangle \}$ The width will be ignored.

```

4755 \newcommand*\LWR@parsepcolumn}[1]{%

Converts to the given column type:

4756 \LWR@parsenormalcolumn{#1}%

Skips the following width token:

4757 \addtocounter{LWR@tablecolspos}{1}%
4758 }
```

54.9 Parsing ‘D’ columns

From the dcolumn package.

`\LWR@parseDcolumn` $\{ \langle thiscolumn \rangle \}$ The three parameters will be ignored.

```

4759 \newcommand*\LWR@parseDcolumn}[1]{%
```

Table 7: Tabular baseline

l	p	m	b	r
			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

Converts to the given column type.

```
4760 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
4761 \addtocounter{LWR@tablecolspos}{3}%
4762 }
```

54.10 Parsing the column specifications



HTML CSS cannot exactly match the L^AT_EX concept of a baseline for a table row. Table 7 shows the L^AT_EX 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 8 for details.

Table 8 describes how each kind of column is converted to HTML.

Bool `LWR@validtablecol` True if found a valid table column type.

```
4763 \newbool{LWR@validtablecol}
```

`\LWR@parsetablecols` $\{ \langle colspecs \rangle \}$

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`.

```
4764 \newcommand*{\LWR@parsetablecols}[1]{%
4765 \LWR@traceinfo{LWR@parsetablecols started}%
```

Remember the original supplied column spec:

Table 8: Tabular HTML column conversions

-
- l, r, c:** Converted to table cells without paragraph tags.
Uses CSS `vertical-align:middle` so that top or bottom-aligned cells may go above or below this cell.
- p:** Converted to table cells with paragraph tags. Ref: Table 7, \LaTeX places the top line of a parbox aligned with the rest of the text line, so CSS `vertical-align:bottom` is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the CSS may be desired, changing `td.tdp`, `td.tdP`, `td.tdprule`, and `td.tdPrule` to `vertical-align: middle`. Another possibility is to change L,R,C, and P to `vertical-align: top` and not worry about the alignment of B and M cells or trying to approximate \LaTeX baselines.
- m:** With paragraph tags, CSS `vertical-align:middle`.
- b:** With paragraph tags, CSS `vertical-align:top` so that the bottom of the text is closest to the middle of the text line.
- P, M, B:** Horizontally-centered versions.
- S:** Converted to 'r'. From the `siunitx` package.
- D:** Converted to 'c'. From the `dcolumn` package.
- @, !, >, <:** One each, in that order.
- Unknown:** Converted to 'l'.
- \newcolumn:** Currently treated as unknown.
-

```
4766 \renewcommand*{\LWR@origcolspec}{#1}%
```

Clear the parsed resulting column spec:

```
4767 \renewcommand*{\LWR@tablecolspec}{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
4768 \setcounter{\LWR@tabletotalcols}{0}%
4769 \setcounter{\LWR@tabletotalcolsnext}{1}%
4770 \LWR@setexparray{\LWR@colatspec}{leftedge}{}%
4771 \LWR@setexparray{\LWR@colatspec}{1}{}%
4772 \LWR@setexparray{\LWR@colatspec}{2}{}%
4773 \LWR@setexparray{\LWR@colatspec}{3}{}%
4774 \LWR@setexparray{\LWR@colbangspec}{leftedge}{}%
4775 \LWR@setexparray{\LWR@colbangspec}{1}{}%
4776 \LWR@setexparray{\LWR@colbangspec}{2}{}%
4777 \LWR@setexparray{\LWR@colbangspec}{3}{}%
4778 \LWR@setexparray{\LWR@colbeforespec}{1}{}%
4779 \LWR@setexparray{\LWR@colbeforespec}{2}{}%
4780 \LWR@setexparray{\LWR@colbeforespec}{3}{}%
4781 \LWR@setexparray{\LWR@colafterspec}{1}{}%
4782 \LWR@setexparray{\LWR@colafterspec}{2}{}%
4783 \LWR@setexparray{\LWR@colafterspec}{3}{}%
```

Starting at the first column specification:

```
4784 \setcounter{\LWR@tablecolspos}{1}%
```

Place the colspecs string length into `\LWR@strresult`, and remember the number of characters in the column specification:

```
4785 \LWR@traceinfo{about to StrLen}%
4786 \noexpandarg%
4787 \StrLen{#1}[\LWR@strresult]%
4788 \fullexpandarg%
4789 \LWR@traceinfo{finished StrLen}%
4790 \setcounter{\LWR@tablecolswidth}{\LWR@strresult}%
```

Scan through the column specifications:

```
4791 \whileof{\not\value{\LWR@tablecolspos}>\value{\LWR@tablecolswidth}}{%
```

Place the next single-character column type into `\LWR@strresult`:

```
4792 \noexpandarg%
4793 \StrChar{#1}{\the\LWR@tablecolspos}[\LWR@strresult]%
4794 \LWR@traceinfo{position \arabic{\LWR@tablecolspos}: \LWR@strresult}%
```

4795 \fullexpandarg%

Not yet found a valid column type

4796 \boolfalse{LWR@validtablecol}%



Note that the parameter for a p{spec} column is a token list which will NOT match l,c,r,p.

```
4797 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{}%
4798 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{}%
4799 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{}%
4800 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{}%
4801 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{}%
4802 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{}%
4803 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{}%
4804 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{}%
4805 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
4806 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
4807 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
4808 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
4809 \IfStrEq{\LWR@strresult}{|}{\LWR@parseskipcolumn}{}%
4810 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
4811 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
4812 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%
```

From the dcolumn package:

```
4813 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{}%
```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```
4814 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%
```

Many people define centered versions “P”, “M”, and “B”:

```
\newcolumnstype{P}[1]{>{\centering\arraybackslash}p{#1}}
```

```
4815 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
4816 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
4817 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%
```

If this column was an invalid column type, convert it to an l column:

```
4818 \ifbool{LWR@validtablecol}{}{}%
4819 \LWR@traceinfo{invalid column type: \LWR@strresult}%
4820 \LWR@parsenormalcolumn{l}%
```

```

4821 }%
4822 \addtocounter{LWR@tablecolspos}{1}%
4823 }%
4824 }%

```

54.11 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.

```

4825 \newcommand*{\LWR@maybenewtablerow}
4826 {%
4827 \ifbool{LWR@startedrow}%
4828 {}% started the row
4829 {% not started the row

```

Remember that now have started the row:

```

4830 \global\booltrue{LWR@startedrow}%

```

Create the row tag, with a class if necessary.

```

4831 \global\booltrue{LWR@intabularmetadata}%
4832 \ifbool{LWR@doinghline}%
4833 {\LWR@htmltag{tr class="hline"{}}\LWR@orignewline}%
4834 {% not doing hline
4835 \ifbool{LWR@doingtbrule}%
4836 {\LWR@htmltag{tr class="tbrule"{}}\LWR@orignewline}%
4837 {\LWR@htmltag{tr}\LWR@orignewline}%
4838 }% end of not doing hline
4839 }% end of not started the row
4840 }

```

54.12 Printing at or bang tags

`\LWR@printatbang` $\langle at -or - bang \rangle$ $\langle index \rangle$

```

4841 \newcommand*{\LWR@printatbang}[2]{%
4842 \edef\LWR@atbangspec{\LWR@getexparray{LWR@col#1spec}{#2}}
4843 \LWR@traceinfo{atbang: !\LWR@atbangspec!}
4844 \ifdefempty{\LWR@atbangspec}%
4845 % \ifthenelse{\isempty{\LWR@atbangspec}}%
4846 {}%
4847 {%

```

```

4848 \LWR@htmltag{td class="td#1%
4849 \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
4850 "%}
4851 \LWR@atbangspec%
4852 \LWR@htmltag{/td}\LWR@orignewline%
4853 }%
4854 }%

```

54.13 Data opening tag

`\LWR@tabledatasinglecolumn` tag Print a table data opening tag with style for alignment

```

4855 \newcommand*{\LWR@tabledatasinglecolumn}{%
4856 {%
4857 \LWR@maybe newtable row%

```

Don't start a new paragraph tag if have already started one:

```

4858 \ifbool{\LWR@intabularmetadata}{%

```

If have found the end of tabular command, do not create the next data cell:

```

4859 \ifbool{\LWR@exitingtabular}{}%
4860 {% not exiting tabular

```

Print the @ and ! contents before first column:

```

4861 \ifthenelse{\cnttest{\value{\LWR@tablecolspos}}=1}{%
4862 {%
4863 \LWR@printatbang{at}{leftedge}%
4864 \LWR@printatbang{bang}{leftedge}%
4865 }% left edge
4866 }% not left edge

```

Fetch the current column's alignment character into `\LWR@strresult`:

```

4867 \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}[\LWR@strresult]%

```

print the start of a new table data cell:

```

4868 \LWR@htmltag{td class="td%

```

append this column's spec:

```

4869 \LWR@strresult%

```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag:

```
4870 \ifthenelse{\equal{\LWR@getexpparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule-}{}%
4871 "}{}%
```

If this is a p, m, b, or X column, allow paragraphs:

```
4872 \ifthenelse{%
4873 \equal{\LWR@strresult}{p}\OR%
4874 \equal{\LWR@strresult}{m}\OR%
4875 \equal{\LWR@strresult}{b}\OR%
4876 \equal{\LWR@strresult}{P}\OR%
4877 \equal{\LWR@strresult}{M}\OR%
4878 \equal{\LWR@strresult}{B}\OR%
4879 \equal{\LWR@strresult}{X}}%
4880 }%
4881 {% allow pars
4882 \LWR@startpars%
4883 \global\booltrue{\LWR@tableparcell}}%
4884 }% allow pars
4885 }% no pars
```

Print the > contents:

```
4886 \LWR@getexpparray{\LWR@colbeforespec}{\theLWR@tablecolspos}}%
4887 \global\boolfalse{\LWR@intabularmetadata}}%
4888 }% not exiting tabular
4889 }{}% in tabular metadata
4890 }%
```

54.14 Midrules

`LWR@midrules` `LWR@midrules` is a data array (section 28) of columns containing Y if a midrule should be created for each column.

Ctr `LWR@midrulecounter` Indexes across the `LWR@midrules` data array.

```
4891 \newcounter{\LWR@midrulecounter}
```

`\LWR@clearmidrules` Start new midrules. Called at beginning of tabular and also at `\`.

Clears all `LWR@midrules` markers for this line.

```
4892 \newcommand*{\LWR@clearmidrules}
4893 {%
4894 \setcounter{\LWR@midrulecounter}{1}}%
```

```

4895 \whiledo{%
4896 \cnttest{\value{LWR@midrulecounter}}{<=}{\value{LWR@tablecolwidth}}%
4897 }%
4898 {%
4899 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{}%
4900 \addtocounter{LWR@midrulecounter}{1}%
4901 }%
4902 }

```

`\LWR@subcmidrule` $\langle width \rangle$ $\langle trim \rangle$ $\langle leftcolumn \rangle$ $\langle rightcolumn \rangle$

Marks `LWR@midrules` data array elements to be “Y” from left to right columns.

```

4903 \newcommand*\LWR@subcmidrule[4]{%
4904 \setcounter{LWR@midrulecounter}{#3}%
4905 \whiledo{\cnttest{\value{LWR@midrulecounter}}{<=}{#4}}%
4906 {%
4907 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{Y}%
4908 \addtocounter{LWR@midrulecounter}{1}%
4909 }% end of the whiledo
4910 }

```

`\LWR@docmidrule` $\langle width \rangle$ $\langle trim \rangle$ $\langle leftcolumn-rightcolumn \rangle$

Marks `LWR@midrules` array elements to be “Y” from left to right columns.

```

4911 \NewDocumentCommand{\LWR@docmidrule}{o d() >{\SplitArgument{1}{-}}m}%
4912 {\LWR@subcmidrule{#1}{#2}{#3}

```

54.15 Multicolumns

54.15.1 Parsing multicolumns

```
4913 \newcounter{LWR@tablemulticolwidth}
```

Indexes into the multicolumn specification:

```
4914 \newcounter{LWR@tablemulticolspos}
```

`\LWR@printmccoltype` $\langle colspec \rangle$ Print any valid column type found. Does not print @, !, >, or < columns or their associated tokens.

This is printed as part of the table data tag’s `class`.

```

4915 \newcommand*\LWR@printmccoltype[1]{%
4916 \LWR@traceinfo{lw@printmccoltype -#1-}%

```

Get one token of the column spec:

```
4917 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```
4918 \IfStrEq{\LWR@strresult}{l}{l}{}%
4919 \IfStrEq{\LWR@strresult}{c}{c}{}%
4920 \IfStrEq{\LWR@strresult}{r}{r}{}%
4921 \IfStrEq{\LWR@strresult}{p}{p}{}%
4922 \IfStrEq{\LWR@strresult}{m}{m}{}%
4923 \IfStrEq{\LWR@strresult}{b}{b}{}%
4924 \IfStrEq{\LWR@strresult}{P}{P}{}%
4925 \IfStrEq{\LWR@strresult}{M}{M}{}%
4926 \IfStrEq{\LWR@strresult}{B}{B}{}%
4927 \IfStrEq{\LWR@strresult}{S}{r}{}%
4928 \IfStrEq{\LWR@strresult}{X}{p}{}%
4929 \LWR@traceinfo{lwr@printmccoltype done}%
4930 }
```

`\LWR@multicolpartext` Print the data with paragraph tags:

```
4931 \newcommand*{\LWR@multicolpartext}{%
4932 \LWR@startpars%
4933 \LWR@multicoltext%
4934 \LWR@stoppars%
4935 }
```

`\LWR@multicolother` $\{<colspec>\}$ For @, !, >, <, print the next token without paragraph tags:

```
4936 \newcommand*{\LWR@multicolother}[1]{%
4937 \addtocounter{LWR@tablemulticolspos}{1}%
4938 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
4939 \LWR@strresult%
```

A valid column data type was found:

```
4940 \booltrue{LWR@validtablecol}%
4941 }
```

`\LWR@multicolskip` Nothing to print for this column type.

```
4942 \newcommand*{\LWR@multicolskip}{%
```

A valid column data type was found:

```
4943 \booltrue{LWR@validtablecol}%
4944 }
```

`\LWR@printmccoldata` $\langle\textit{colspec}\rangle$ Print the data for any valid column type found.

```
4945 \newcommand*\LWR@printmccoldata[1]{%
4946 \LWR@traceinfo{lwr@printmccoldata -#1}%
```

Not yet found a valid column type:

```
4947 \boolfalse\LWR@validtablecol}%
```

Get one token of the column spec:

```
4948 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
4949 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
4950 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
4951 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
4952 \IfStrEq{\LWR@strresult}{D}{}%
4953 \addtocounter{LWR@tablemulticolspos}{3}% skip parameters
4954 \LWR@multicoltext%
4955 }{}%
4956 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolparttext}{}%
4957 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolparttext}{}%
4958 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolparttext}{}%
4959 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolparttext}{}%
4960 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolparttext}{}%
4961 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolparttext}{}%
4962 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolparttext}{}%
4963 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolparttext}{}%
4964 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
4965 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@multicolother{#1}}{%
4966 \IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}{%
4967 \IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}{%
4968 \IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}{%
```

If an invalid column type:

```
4969 \ifbool{LWR@validtablecol}{\LWR@multicoltext}%
```

Tracing:

```
4970 \LWR@traceinfo{lwr@printmccoldata done}%
4971 }
```

`\parsemulticolumnalignment` $\langle\textit{1: colspec}\rangle$ $\langle\textit{2: printresults}\rangle$

Scan the multicolumn specification and execute the printfunction for each entry.

Note that the spec for a `p{spec}` column, or `@`, `>`, `<`, is a token list which will NOT match `l`, `c`, `r`, or `p`.

```
4972 \newcommand*\LWR@parsemulticolumnalignment}[2]{%
4973 \setcounter{LWR@tablemulticolspos}{1}%
4974 \StrLen{#1}[\LWR@strresult]%
4975 \setcounter{LWR@tablemulticolwidth}{\LWR@strresult}%
```

Scan across the tokens in the column spec:

```
4976 \whiledo{%
4977 \not\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolwidth}%
4978 }%
4979 {%
```

Execute the assigned print function for each token in the column spec:

```
4980 #2{#1}%
```

Move to the next token in the column spec:

```
4981 \addtocounter{LWR@tablemulticolspos}{1}%
4982 }%
4983 }
```

54.15.2 High-level multicolumn interface

```
4984 \newcommand{\LWR@multicoltext}{}
```

```
\LWR@domulticolumn {<1: numcols>} {<2: colspec>} {<3: text>}
```

```
4985 \NewDocumentCommand{\LWR@domulticolumn}{m m +m}{%
4986 \LWR@traceinfo{lwr@domulticolumn -#1- -#2-}%
```

Remember the text to be inserted, and remember that a valid column type was found:

```
4987 \renewcommand{\LWR@multicoltext}{%
4988 #3%
4989 \booltrue{LWR@validtablecol}%
4990 }%
```

Row processing:

```
4991 \LWR@maybenewtablerow%
```

Begin the opening table data tag:

```
4992 \LWR@htmltag{td colspan="#1"
4993 class="td%
```

Print the column type:

```
4994 \LWR@parsemulticolumnalignment{#2}{\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”.

```
4995 \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule-}{%
```

Close the class tag’s opening quote:

```
4996 "%
4997 }% end of the opening table data tag
4998 \global\boolfalse{\LWR@intabularmetadata}%
4999 \LWR@parsemulticolumnalignment{#2}{\LWR@printmccoldata}%
5000 }
```

54.15.3 Longtable captions

Bool `LWR@starredlongtable` Per the caption package, step the counter if `longtable*`.

```
5001 \newbool{\LWR@starredlongtable}
5002 \boolfalse{\LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
5003 \providecommand*\LTcapttype{table}
```

`\LWR@longtabledatacaptiontag` * [*toc entry*] {*caption*}

```
5004 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
5005 {%
```

Remember the latest name for `\nameref`:

```
5006 \IfValueTF{#2}{% optional given?
5007 \ifthenelse{\equal{#2}{}}{% optional empty?
5008 {\LWR@setlatestname{#3}}% empty
5009 {\LWR@setlatestname{#2}}% given and non-empty
5010 }% optional given
5011 {\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:

```
5012 \LWR@tabularhtmlcolumns{1}{\theLWR@tabletotalcols}
```

Create the multicolumn tag:

```
5013 \LWR@domulticolumn{\theLWR@tabhtmlcoltotal}{P}{% \LWR@domulticolumn
5014 \IfBooleanTF{#1}% star?
```

Star version, show a caption but do not make a LOT entry:

```
5015 {% yes star
5016 \LWR@htmlblocktag{figcaption}%
5017 #3%
5018 \LWR@htmlblocktag{/figcaption}%
5019 }%
5020 {% No star:
```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```
5021 \ifbool{LWR@starredlongtable}%
5022 {%
5023 \ifthenelse{\equal{#2}{}}% TOC entry
5024 {}%
5025 {%
5026 \refstepcounter{\LTcaption}%
5027 \protected@edef\@currentlabel{%
5028 \csuse{p@\LTcaption}\csuse{the\LTcaption}}%
5029 }%
5030 }{}%
```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```
5031 \LWR@htmlblocktag{figcaption}%
5032 \csuse{fnum@\LTcaption}\CaptionSeparator#3%
5033 \LWR@htmlblocktag{/figcaption}%
```

See if an optional caption was given:

```
5034 \ifthenelse{\equal{#2}{}}% TOC entry empty
```

if the optional caption was given, but empty, do not form a TOC entry

```
5035 {}%
```

If the optional caption was given, but might only be []:

```
5036 {% TOC entry not empty
5037 \IfNoValueTF{#2}% No TOC entry?
```

The optional caption is []:

```
5038 {% No TOC entry
5039 \addcontentsline%
5040 {\csuse{ext@\LTcaption}}%
5041 {\LTcaption}%
5042 {%
5043 \protect\numberline%
5044 {\csuse{p@\LTcaption}\csuse{the\LTcaption}}%
5045 {\ignorespaces #3\protect\relax}%
5046 }%
5047 }% end of No TOC entry
```

The optional caption has text enclosed:

```
5048 {% yes TOC entry
5049 \addcontentsline%
5050 {\csuse{ext@\LTcaption}}%
5051 {\LTcaption}%
5052 {%
5053 \protect\numberline%
5054 {\csuse{p@\LTcaption}\csuse{the\LTcaption}}%
5055 {\ignorespaces #2\protect\relax}%
5056 }%
5057 }% end of yes TOC entry
5058 }% end of TOC entry not empty
5059 }% end of no star
```

Skip any trailing @ or ! columns for this cell:

```
5060 \booltrue{LWR@skipatbang}%
5061 }% end of \LWR@domulticolumn
5062
5063 \addtocounter{LWR@tablecolspos}{\theLWR@tabletotalcols}
5064 \addtocounter{LWR@tablecolspos}{-1}
5065
5066 }
```

54.15.4 Counting HTML tabular columns

The L^AT_EX 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 L^AT_EX `\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.

```
5067 \newcounter{LWR@tabhtmlcolindex}
5068 \newcounter{LWR@tabhtmlcolend}
5069 \newcounter{LWR@tabhtmlcoltotal}
```

```
\LWR@tabularhtmlcolumns {<starting LATEX column>} {<number LATEX columns>}
```

Compute the total number of HTML columns being spanned, considering the starting L^AT_EX table column and the number of L^AT_EX tabular columns being spanned. Any @ and ! columns within this span are included in the total count. The resulting number of HTML columns is returned in the counter `LWR@tabhtmlcoltotal`.

```
5070 \newcommand*{\LWR@tabularhtmlcolumns}[2]{%
```

Count the starting index, compute ending index, and begin with the count being the L^AT_EX span, to which additional @ and ! columns may be added:

```
5071 \setcounter{LWR@tabhtmlcolindex}{#1}%
5072 \setcounter{LWR@tabhtmlcoltotal}{#2}%
5073 \setcounter{LWR@tabhtmlcolend}{#1}%
5074 \addtocounter{LWR@tabhtmlcolend}{#2}%
```

Walk across the L^AT_EX columns looking for @ and ! columns:

```
5075 \whiledo{\value{LWR@tabhtmlcolindex}<\value{LWR@tabhtmlcolend}}{%
```

Temporarily define a macro equal to the @ specification for this column:

```
5076 \edef\LWR@atbangspec{\LWR@getexpparray{LWR@colatspec}{\theLWR@tabhtmlcolindex}}%
```

If the @ specification is not empty, add to the count:

```
5077 \ifdefempty{\LWR@atbangspec}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the ! columns:

```
5078 \edef\LWR@atbangspec{\LWR@getexpparray{LWR@colbangspec}{\theLWR@tabhtmlcolindex}}%
5079 \ifdefempty{\LWR@atbangspec}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Move to the next L^AT_EX column:

```
5080 \addtocounter{LWR@tabhtmlcolindex}{1}%
5081 }%
```

If at the left-most column, also skip the leftmost @ and ! cells:

```
5082 \ifthenelse{\value{LWR@tablecolspos}=1}{%
5083 \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{leftedge}}%
5084 \ifdefempty{\LWR@atbangspec}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
5085 \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{leftedge}}%
5086 \ifdefempty{\LWR@atbangspec}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
5087 }{}%
5088 }
```

54.15.5 \tabledatamulticolumntag

`\LWR@tabledatamulticolumntag` $\langle numcols \rangle$ $\langle alignment \rangle$ $\langle text \rangle$

```
5089 \NewDocumentCommand{\LWR@tabledatamulticolumntag}{m m +m}%
5090 {%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
5091 \LWR@tabularhtmlcolumns{\theLWR@tablecolspos}{#1}
```

Create the multicolumn tag:

```
5092 \LWR@domulticolumn{\theLWR@tabhtmlcoltotal}{#2}{#3}%
```

Move to the next L^AT_EX column:

```
5093 \addtocounter{LWR@tablecolspos}{#1}%
5094 \addtocounter{LWR@tablecolspos}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
5095 \booltrue{LWR@skipatbang}%
5096 }
```

54.16 Multirow

Pkg `multirow`

`\LWR@tabledatamultirowtag` $\langle numrows \rangle$ [$\langle bigstruts \rangle$] $\langle width \rangle$ [$\langle fixup \rangle$] $\langle text \rangle$

```
5097 \NewDocumentCommand{\LWR@tabledatamultirowtag}{m o m o m}%
5098 {%
5099 \LWR@maybe newtable row%
```

Print the start of a new table data cell:

```
5100 \LWR@htmltag{td rowspan="#1" class="td%
```

Append this column's spec:

```
5101 \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}%
```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag:

```
5102 \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
5103 "{}}
```

While printing the text, redefine `\` to generate a new line

```
5104 \begingroup \LetLtxMacro{\}{\LWR@endoffline} #5 \endgroup
5105 \LWR@stoppars%
5106 \global\boolfalse{\LWR@intabularmetadata}%
5107 }%
```

54.17 Utility macros inside a table

```
5108 \newcommand*{\LWR@donothing}{}
5109 \newcommand*{\LWR@domidrule}{\booltrue{\LWR@doinghline}}
5110 \newcommand*{\LWR@dotbrule}{\booltrue{\LWR@doingtbrule}}
```

54.18 Checking for a new table cell

`\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:

```
5111 \newbool{\LWR@exitingtabular}
5112
5113 \newcommand*{\LWR@tabledatacolumnntag}%
5114 {%
5115 \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:

```
5116 \let\LWR@mynext\LWR@tabledatasinglecolumnntag%
```

If exiting the tabular:

```
5117 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\end}}%
5118 {\booltrue{LWR@exitingtabular}}{ }%
```

longtable can have a caption in a cell

```
5119 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\caption}}%
5120 {\let\mynext\LWR@donothing}{ }%
```

Look for other things which would not start a table cell:

```
5121 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multicolumn}}%
5122 {\let\mynext\LWR@donothing}{ }%
5123 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multirow}}%
5124 {\let\mynext\LWR@donothing}{ }%
```

if come to an \mrowcell, this is a cell to be skipped over

```
5125 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\mrowcell}}%
5126 {\let\mynext\LWR@donothing}{ }%
5127 %
5128 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\hrline}}%
5129 {\let\mynext\LWR@donothing}{ }%
5130 %
5131 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\toprule}}%
5132 {\let\mynext\LWR@donothing}{ }%
5133 %
5134 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\midrule}}%
5135 {\let\mynext\LWR@donothing}{ }%
5136 %
5137 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cmidrule}}%
5138 {\let\mynext\LWR@donothing}{ }%
5139 %
5140 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\specialrule}}%
5141 {\let\mynext\LWR@donothing}{ }%
5142 %
5143 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cline}}%
5144 {\let\mynext\LWR@donothing}{ }%
5145 %
5146 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\bottomrule}}%
5147 {\let\mynext\LWR@donothing}{ }%
5148 %
5149 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpprintonly}}%
5150 {\let\mynext\LWR@donothing}{ }%
5151 %
5152 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpHTMLonly}}%
5153 {\let\mynext\LWR@donothing}{ }%
```

no action for an \end token

Add similar to the above for any other non-data tokens which might appear in the table.

Start the new table cell if was not any of the above:

```
5154 \mynext%
5155 }

5156 \end{warpHTML}
```

54.19 `\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: 5157 `\begin{warpall}`
 5158 `\newcommand*{\mrowcell}{}`
 5159 `\end{warpall}`

54.20 New `\tabular` definition

for HTML output: 5160 `\begin{warpHTML}`

These are default definitions in case `booktabs` is not loaded, and are not expected to used, but must exist as placeholders.

```
5161 \newcommand*\LWR@origtoprule}[1]{\hline}
5162 \newcommand*\LWR@origmidrule}[1]{\hline}
5163 \LetLtxMacro\LWR@origcmidrule\cline
5164 \newcommand*\LWR@origbottomrule}[1]{\hline}
5165 \newcommand*\LWR@origaddlinespace}[1][1]{}
5166 \newcommand*\LWR@origmorecmidrules{}
5167 \newcommand*\LWR@origspecialrule}[3]{\hline}
```

Env `LWR@tabular` [*vertical position*] [*colspecs*]

The new tabular environment will be `\let` in `\LWR@LwarpStart`, since `siunitx` might redefine `tabular` in the user's document.

```
5168 \StartDefiningTabulars
5169
5170 \newenvironment*LWR@tabular}[2][1]
5171 {%
5172 \LWR@traceinfo{LWR@tabular started}%
5173 \addtocounter{LWR@tabulardepth}{1}%
```

Not yet started a table row:

```
5174 \global\boolfalse{LWR@startedrow}%
```

Not yet doing an hline:

```
5175 \global\boolfalse{LWR@doinghline}%
```

Not yet doing a top/bottom rule:

```
5176 \global\boolfalse{LWR@doingtbrule}%
```

Have not yet found the end of tabular command:

```
5177 \boolfalse{LWR@exitingtabular}%
```

Create the table tag:

```
5178 \global\booltrue{LWR@intabularmetadata}%
```

```
5179 \LWR@forcenewpage
```

```
5180 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
5181 \LWR@parsetablecols{#2}%
```

Table col spec is: `\LWR@tablecolspec` which is a string of `llccrr`, etc.

Do not place the table inside a paragraph:

```
5182 \LWR@stoppars%
```

Track column # for setting text-align:

```
5183 \setcounter{LWR@tablecolspos}{1}%
```

Start looking for midrules:

```
5184 \LWR@clearmidrules%
```

`\` becomes a macro to end the table row:

```
5185 \LetLtxMacro{\}{\LWR@tabularendoffline}%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with `\LWR@getmynextoken` to see if the next token might create a new data cell:

```

5186 \LWR@traceinfo{LWR@tabular: redefining macros}%
5187 \renewcommand*{\hline}{\LWR@domidrule\LWR@getmynexttoken}%
5188
5189 \RenewDocumentCommand{\cline}{m}%
5190 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%
5191
5192 \DeclareDocumentCommand{\toprule}{o}{\LWR@dotbrule\LWR@getmynexttoken}%
5193
5194 \DeclareDocumentCommand{\midrule}{o}{\LWR@domidrule\LWR@getmynexttoken}%
5195
5196 \DeclareDocumentCommand{\cmidrule}{o d() m}%
5197 {\LWR@docmidrule[##1](##2){##3}\LWR@getmynexttoken}%
5198
5199 \DeclareDocumentCommand{\bottomrule}{o}{\LWR@dotbrule\LWR@getmynexttoken}%
5200 \DeclareDocumentCommand{\addlinespace}{o}{ }
5201 \DeclareDocumentCommand{\morecmidrules}{-}{ }
5202 \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`.

```

5203 \LetLtxMacro{\multicolumn}{\LWR@tabledata multicolumntag}%
5204 \LetLtxMacro{\multirow}{\LWR@tabledata multirowtag}%
5205 \renewcommand*{\mrowcell}{\global\booltrue{LWR@skippingmrowcell}}%
5206 \LetLtxMacro{\caption}{\LWR@longtabledata captiontag}%

```

Reset for new processing:

```

5207 \global\boolfalse{LWR@tableparcell}%
5208 \global\boolfalse{LWR@skippingmrowcell}%
5209 \global\boolfalse{LWR@skipatbang}%

```

Set `&` for its special meaning inside the tabular:

```

5210 \StartDefiningTabulars%
5211 \protected\gdef&{\LWR@tabularampersand}%

```

Look ahead for a possible table data cell:

```

5212 \LWR@traceinfo{LWR@tabular: about to LWR@getmynexttoken}%
5213 \LWR@getmynexttoken%
5214 }%

```

Ending the environment:

```

5215 {%
5216 \LWR@traceinfo{LWR@tabular ending}%
5217 \LWR@closetabledatacell%

```

```

5218 \LWR@htmlblocktag{/tr}%
5219 \LWR@htmlblocktag{/table}%
5220 \global\boolfalse{LWR@intabularmetadata}%

```

Unnest one level of tabular:

```
5221 \addtocounter{LWR@tabulardepth}{-1}%
```

Restore & to its usual meaning:

```

5222 \protected\gdef&{\LWR@origampmacro}%
5223 \EndDefiningTabulars%
5224 \LWR@traceinfo{LWR@tabular finished ending}%
5225 }
5226
5227 \EndDefiningTabulars

5228 \end{warpHTML}

```

54.21 Array

Pkg `array`

`array` is also automatically loaded by `siunitx`.

55 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 9 shows the data structures related to cross-referencing.

for HTML output: `5229 \begin{warpHTML}`

55.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

Table 9: Cross-referencing data structures

Original L^AT_EX:	(print and HTML)
\refstepcounter: Steps the counter and sets \@currentlabel.	
\@currentlabel: \p@<ctr>\the<ctr> Updated by \refstepcounter.	
\label: Writes to the .aux file: <code>\newlabel{<label>}{\@currentlabel}{\thepage}}</code>	
\newlabel: When the .aux file is read, sets \r@<label>.	
\r@<label>: Set to: {\@currentlabel}{\thepage}	
\ref: Returns the first part of \r@<label>.	
\pageref: Returns the second part of \r@<label>.	
Added by lwarp:	(HTML only)
\label: Adds HTML tags (section 55.3), plus \splabel data (section 55.2):	
zLWR@name: The section name for this label.	
zLWR@htmlfilenumber: The file number or name for this label.	
zLWR@lateximagedepth: The lateximagedepth for this label.	
zLWR@lateximagenumber: The lateximagenumber for this label.	
\nameref: Emulated from hyperref for lwarp. See section 55.4.	
\ref and \nameref: Adds HTML tags. See section 55.4.	
Added by amsmath:	(print and HTML)
\label: Execution is delayed until the math environment is completed.	
\ltx@label: L ^A T _E X \label, (HTML: patched by lwarp,) later patched by cleveref.	
Added by cleveref:	(print and HTML)
\refstepcounter: Added: sets \cref@currentlabel.	
\cref@currentlabel: (<type>=<ctr> unless an alias is used): [<type>][\arabic{<ctr>}][<parent ctrs>]{\p@<ctr>\the<ctr>} Also see section 41.4 for use with footnotes.	
\label: Writes to the .aux file: <code>\newlabel{<label>@cref}{\cref@currentlabel}{\thepage}}</code>	
\newlabel: (Unchanged.) When the .aux file is read, sets \r@<label>@cref.	
\r@<label>@cref: Set to: {\cref@currentlabel}{\thepage}	
Utility functions: See \cref@getlabel, \cref@gettype, \cref@getcounter, \cref@getprefix.	
Cross-referencing names: \crefname and \Crefname assign human-readable names for references to this counter type.	
Additionally patched by lwarp:	(HTML only)
\cref, etc.: Modified for lwarp. See section 64.	
\label inside math: See section 59.4.1.	
Footnotes: See \noteentry in section 41.4.	

```
5230 \newcommand*{\@currentlabelname}{}
```

```
\LWR@stripperperiod {<text>} [⟨.⟩]
```

Removes a trailing period.

```
5231 \def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%
```

```
\LWR@setlatestname {<object name>}
```

Removes `\label`, strips any final period, and remembers the result.

```
5232 \newcommand*{\LWR@setlatestname}[1]{%
```

Remove `\label` and other commands from the name, the strip any final period.
See `zref-titleref` and `getttitlestring`.

```
5233 \GetTitleStringExpand{#1}%
```

```
5234 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
```

```
5235 \edef\@currentlabelname{%
```

```
5236 \expandafter\LWR@stripperperiod\@currentlabelname%
```

```
5237 \ltx@empty.\ltx@empty\@nil%
```

```
5238 }%
```

```
5239 }
```

55.2 Zref setup

See:

[http://tex.stackexchange.com/questions/57194/
extract-section-number-from-equation-reference](http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference)

Create a new property list called `special`:

```
5240 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
5241 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
5242 \zref@newprop{zLWR@htmlfilenumber}{%
```

```
5243 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilenumber}}%
```

```
5244 }%
```

Additional properties for lateximages:

```
5245 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
5246 \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:

```
5247 \zref@addprop{special}{zLWR@name}
5248 \zref@addprop{special}{zLWR@htmlfilenumber}
5249 \zref@addprop{special}{zLWR@lateximagedepth}
5250 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
5251 \newcommand*{\LWR@spreff}[2]{%
5252 \zref@extractdefault{#1}{#2}{??}}
```

`\LWR@nameref` $\langle label \rangle$ Returns the section name for this label:

```
5253 \newcommand*{\LWR@nameref}[1]{%
5254 \LWR@spreff{#1}{zLWR@name}%
5255 }
```

`\LWR@htmlfileref` $\langle label \rangle$ Returns the file number for this label:

```
5256 \newcommand*{\LWR@htmlfileref}[1]{%
5257 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
5258 \LWR@spreff{#1}{zLWR@htmlfilenumber}%
5259 }
```

`\LWR@lateximagedepthref` $\langle label \rangle$ Returns the lateximagedepth for this label:

```
5260 \newcommand*{\LWR@lateximagedepthref}[1]{%
5261 \LWR@spreff{#1}{zLWR@lateximagedepth}%
5262 }
```

`\LWR@lateximagenumberref` $\langle label \rangle$ Returns the lateximagenumber for this label:

```
5263 \newcommand*{\LWR@lateximagenumberref}[1]{%
5264 \LWR@spreff{#1}{zLWR@lateximagenumber}%
5265 }
```

`\LWR@splabel` $\langle label \rangle$ Sanitize the name and then creates the label:

```

5266 \newcommand*\LWR@splabel}[1]{%
5267 \LWR@setlatestname{\@currentlabelname}%
5268 \zref@labelbylist{#1}{special}}

```

55.3 Labels

`\LWR@subsublabel` $\langle label \rangle$ Creates an HTML id tag.

```

5269 \newcommand*\LWR@subsublabel}[1]{%

```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```

5270 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{%
5271 }{%
5272 {% not lateximage

```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```

5273 \ifbool{LWR@doingstartpars}%
5274 {% pars allowed
5275 \ifbool{LWR@doingapar}
5276 {% par started
5277 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5278 }% par started
5279 {% par not started
5280 \LWR@stoppars%
5281 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5282 \LWR@startpars%
5283 }% par not started
5284 }% pars allowed
5285 {% pars not allowed
5286 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5287 }% pars not allowed
5288 }% not lateximage
5289 }

```

`\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.

```

5290 \NewDocumentCommand{\LWR@newlabel}{m o}{%
5291 \LWR@traceinfo{LWR@newlabel: starting}%
5292 \LWR@traceinfo{LWR@newlabel: !#1!}%
5293 % \@bspack%

```

Create a traditional LaTeX label, as modified by `cleveref`:

```

5294 \LWR@origlabel{#1}%

```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```

5295 \LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
5296 \LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
5297 \LWR@traceinfo{LWR@newlabel: LWR@htmlfilenumber is \theLWR@htmlfilenumber}%
5298 \LWR@splabel{#1}%
5299 \LWR@subsublabel{#1}%
5300 % \@espack%
5301 \LWR@traceinfo{LWR@newlabel: done}%
5302 }

```

55.4 References

`\LWR@startref` $\{label\}$ (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, `#` character, and a label.

```

5303 \newcommand*{\LWR@startref}[1]
5304 {%
5305 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
5306 \LWR@traceinfo{LWR@startref A: !#1!}%

```

Create the filename part of the link:

```

5307 \LWR@htmltag{a href="%
5308 \LWR@traceinfo{LWR@startref B}%
5309 \LWR@htmlrefsectionfilename{#1}}%
5310 \LWR@traceinfo{LWR@startref C}%
5311 \#%

```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```

5312 \LWR@traceinfo{LWR@startref D: !#1!}%
5313 \ifthenelse{\equal{\LWR@lidref}{??}}%

```

“??” if LWR@lateximagedepth is unknown, so create a link with an unknown destination:

```
5314 {%
5315 \LWR@traceinfo{LWR@startref D0: ??}%
5316 ??}%
```

If LWR@lateximagedepth is known. Use a lateximage if the depth is greater than zero, or a regular link otherwise:

```
5317 {%
5318 \LWR@traceinfo{LWR@startref D1: \LWR@lidref}%
5319 \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}%
5320 {%
5321 \LWR@traceinfo{LWR@startref D2: \LWR@lidref}%
5322 lateximage\LWR@lateximagenumberref{#1}%
5323 }%
5324 {%
5325 \LWR@traceinfo{LWR@startref D3}%
5326 #1%
5327 }%
5328 }%
5329 \LWR@traceinfo{LWR@startref E}%
```

Closing quote:

```
5330 "{}}%
5331 \LWR@traceinfo{LWR@startref F}%
5332 }
```

`\LWR@subnewref` $\langle label \rangle$ $\langle label \text{ or } sub@label \rangle$

Factored for the subfig package. Uses the original label for the hyper-reference, but prints its own text, such as “1(b)”.

```
5333 \NewDocumentCommand{\LWR@subnewref}{m m}{%
5334 \LWR@traceinfo{LWR@subnewref #1 #2}%
5335 \LWR@startref{#1}%
5336 \LWR@origref{#2}%
5337 \LWR@htmltag{/a}%
5338 }
```

`\ref` * $\langle label \rangle$ `\ref` is `\let` to `\LWR@newref`

`\LWR@newref` * $\langle label \rangle$ Create an internal document reference link, or without a link if starred per hyperref.

```

5339 \NewDocumentCommand{\LWR@newref}{s m}{%
5340 \LWR@traceinfo{LWR@newref #2}%
5341 \IfBooleanTF{#1}%
5342 {\LWR@origref{#2}}%
5343 {\LWR@subnewref{#2}{#2}}%
5344 }

```

`\pagerefPageFor` Text for starred page references.

```
5345 \newcommand*{\pagerefPageFor}{see }
```

`\pageref` * $\langle label \rangle$ Create an internal document reference, or just the unlinked number if starred, per `hyperref`.

```

5346 \NewDocumentCommand{\LWR@newpageref}{s m}{%
5347 \IfBooleanTF{#1}%
5348 {\(\pagerefPageFor\LWR@origref{#2})}%
5349 {\(\cpageref{#2})}%
5350 }

```

`\nameref` $\langle label \rangle$

```

5351 \newcommand*{\nameref}[1]{%
5352 \LWR@traceinfo{nameref A}%
5353 \LWR@startref{#1}%
5354 \LWR@traceinfo{nameref B}%
5355 \LWR@nameref{#1}%
5356 \LWR@traceinfo{nameref C}%
5357 \LWR@htmltag{/a}%
5358 \LWR@traceinfo{nameref D}%
5359 }

```

`\Nameref` $\langle label \rangle$ In print, adds the page number. In HTML, does not.

```
5360 \let\Nameref\nameref
```

55.5 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.

```
5361 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
5362 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Create a link with a text name:

```
\LWR@subhyperref {<URL>} {<text>}
```

```
5363 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
5364 \LWR@htmltag{a href="#1" target="\_{}blank"\LWR@orignewline}#2\LWR@htmltag{/a}%
5365 \LWR@ensuredoingapar%
5366 }
```

```
\LWR@subhyperrefclass {<URL>} {<text>} {<htmlclass>}
```

```
5367 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
5368 \LWR@htmltag{a href="#1"
5369 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
5370 \LWR@ensuredoingapar%
5371 }
```

```
\href [ <options> ] {<URL>} {<text>}
```

Create a link with accompanying text:

```
5372 \NewDocumentCommand{\LWR@hrefb}{0{} m +m}{%
5373 \LWR@subhyperref{#2}{#3}%
5374 \endgroup%
5375 \LWR@ensuredoingapar%
5376 }
5377
5378 \newcommand{\href}{%
5379 \LWR@ensuredoingapar%
5380 \begingroup%
5381 \catcode'\_ =12
5382 \LWR@hrefb%
5383 }
```

```
\nolinkurl {<URL>}
```

Print the name of the link without creating the link:

```

5384 \newcommand*\LWR@nolinkurlb}[1]{#1\endgroup\LWR@ensuredoingapar}
5385
5386 \newcommand{\nolinkurl}{%
5387 \LWR@ensuredoingapar%
5388 \begingroup\catcode'\_ =12
5389 \LWR@nolinkurlb%
5390 }

```

`\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`.

```

5391 \newcommand*\LWR@urlb}[1]{%
5392 \href{#1}{#1}%
5393 \endgroup%
5394 \LWR@ensuredoingapar%
5395 }
5396
5397 \newcommand{\LWR@urla}{%
5398 \LWR@ensuredoingapar%
5399 \begingroup\catcode'\_ =12
5400 \LWR@urlb%
5401 }
5402
5403 \let\url\LWR@urla

```

`\LWR@subinlineimage` $[\langle alttag \rangle] \{\langle class \rangle\} \{\langle filename \rangle\} \{\langle extension \rangle\} \{\langle style \rangle\}$

```

5404 \newcommand*\LWR@subinlineimage}[5] [] {%
5405 \ifthenelse{\equal{#1}{}}%
5406 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"{}}}%
5407 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{}}}%
5408 }
5409 \end{warpHTML}

```

Table 10: 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 L^AT_EX 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.

56 Floats

Floats are supported, although partially through emulation.

Table 10 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”.

56.1 Float captions

for HTML output: 5410 `\begin{warpHTML}`

`\LWR@floatbegin` $\langle type \rangle$ [$\langle placement \rangle$]

Begins a `\newfloat` environment.

```
5411 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
5412 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
5413
5414 === #1 begin
5415
5416 }{}}%
5417 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
5418 \addtocounter{LWR@thisfloat}{1}%
5419 \booltrue{LWR@freezethisfloat}%
```

```
5420 \beginingroup
```

Settings while inside the environment:

```
5421 \LWR@origraggedright
```

Open an HTML figure tag:

```
5422 \LWR@htmltag{figure id="autofloat-\arabic{LWR@thisfloat}" class="#1"}
```

```
5423 \renewcommand*{\@capttype}{#1}
5424 \caption@settype{#1}
5425 \LWR@startpars
5426 }
```

`\@float` Support packages which create floats directly.
`\@dblfloat`

```
5427 \let\@float\LWR@floatbegin
5428 \let\@dblfloat\LWR@floatbegin
```

`\LWR@floatend` Ends a `\newfloat` environment.

```
5429 \newcommand*{\LWR@floatend}{%
5430 \LWR@stoppars%
5431 \LWR@htmlend{figure}}%
```

```

5432 \endgroup%
5433 \boolfalse{LWR@freezethisfloat}%
5434 \LWR@startpars%
5435 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
5436
5437 === end
5438
5439 }{}%
5440 }

```

`\end@float` Support packages which create floats directly.
`\end@dblfloat`

```

5441 \let\end@float\LWR@floatend
5442 \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.

```

5443 \newcounter{LWR@thisfloat}

```

Bool `LWR@freezethisfloat` Prevents multiple increments of `\LWR@thisfloat` inside a float.

```

5444 \newbool{LWR@freezethisfloat}
5445 \boolfalse{LWR@freezethisfloat}

```

`\LWR@maybeinthisfloat`

```

5446 \newcommand*{\LWR@maybeinthisfloat}{%
5447 \ifbool{LWR@freezethisfloat}{\addtocounter{LWR@thisfloat}{1}}%
5448 }

```

`\@capttype` Remembers which float type is in use.

```

5449 \newcommand*{\@capttype}{}

```

56.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```

5450 \AtBeginDocument{\providecommand*\CaptionSeparator}{:-}}

```

`\@makecaption` $\langle name \text{ and } num \rangle$ $\langle text \rangle$

Prints the float type and number, the caption separator, and the caption text.

```

5451 \AtBeginDocument{\renewcommand{\@makecaption}[2]{#1\CaptionSeparator#2}}

```

56.1.2 Caption and LOF linking and tracking

When a new HTML file is marked in the L^AT_EX PDF file, the L^AT_EX page number at that point is stored in `LWR@latestautopage`, (and the associated filename is remembered by the special L^AT_EX 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.

```
5452 \newcounter{LWR@nextautofloat}
5453 \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.

```
5454 \newcommand*{\LWRsetnextfloat}[2]{%
5455 \setcounter{LWR@nextautopage}{#1}%
5456 \setcounter{LWR@nextautofloat}{#2}%
5457 }
```

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.

```
5458 \newcounter{LWR@latestautopage}
5459 \setcounter{LWR@latestautopage}{1}
```

```
5460 \let\LWR@origcaption@begin\caption@begin
5461 \let\LWR@origcaption@end\caption@end
5462 \let\LWR@orig@@par\@@par
```

`\LWR@caption@begin` Low-level patches to create HTML tags for captions.

```
5463 \newcommand{\LWR@caption@begin}
5464 {
5465 \LWR@traceinfo{LWR@caption@begin}%
```

Keep par and minipage changes local:

```
5466 \begingroup%
```

The caption code was not allowing the closing par tag:

```
5467 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a minipage or \parbox inside the caption:

```
5468 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{-}{-}%
```

```
5469 \RenewDocumentCommand{\parbox}{0{t} o 0{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
5470 \LWR@htmlblocktag{figcaption}%
```

```
5471 \LWR@origcaption@begin%
```

```
5472 }
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
5473 \newcommand{\LWR@caption@end}
```

```
5474 {%
```

```
5475 \LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
5476 \LWR@origvspace*{\baselineskip}%
```

Closing tag:

```
5477 \LWR@htmlblocktag{/figcaption}%
```

```
5478 \endgroup%
```

```
5479 % \leavevmode% avoid bad space factor (0) error
```

```
5480 \LWR@traceinfo{\LWR@caption@end: done}%
```

```
5481 }
```

`\caption@begin` Low-level patches to create HTML tags for captions.

`\caption@end`

```
5482 \AtBeginDocument{
```

```
5483 \let\caption@begin\LWR@caption@begin
```

```
5484 \let\caption@end\LWR@caption@end
```

```
5485 }
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```

5486 \let\LWR@origcaptionlistentry\captionlistentry
5487
5488 \renewcommand*\captionlistentry{%
5489 \LWR@maybeinthisfloat%
5490 \LWR@ensuredoingapar%
5491 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5492 \LWR@origcaptionlistentry%
5493 }
5494
5495 \def\LWR@LTcaptionlistentry{%
5496 \LWR@ensuredoingapar%
5497 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5498 \bgroup
5499 \@ifstar{\egroup\LWR@LTcaptionlistentry}% gobble *
5500 {\egroup\LWR@LTcaptionlistentry}}%
5501 \def\LWR@LTcaptionlistentry#1{%
5502 \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.

```

5503 \let\LWR@origaddcontentsline\addcontentsline
5504
5505 \renewcommand*\addcontentsline[3]{%
5506 \ifthenelse{\equal{#1}{toc}}{ }{%
5507 \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{ }
5508 \addtocontents{\@nameuse{ext@#2}}{%
5509 \protect\LWRsetnextfloat%
5510 {\arabic{LWR@latestautopage}}%
5511 {\arabic{LWR@thisfloat}}%
5512 }% addtocontents
5513 }% not toc
5514 \LWR@origaddcontentsline{#1}{#2}{#3}%
5515 }

```

`\captionof` Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

5516 \AtBeginDocument{
5517 \let\LWR@origcaptionof\captionof
5518
5519 \renewcommand*\captionof{%
5520 \LWR@maybeinthisfloat%
5521 \LWR@stoppars
5522 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5523 \LWR@origcaptionof%
5524 }

```

```
5525 }
```

```
5526 \end{warpHTML}
```

57 Table of Contents, LOF, LOT

This section controls the generation of the TOC, LOF, LOT.

The `.toc`, `.lof`, and `.lot` files are named by the source code `\jobname`.

In HTML, the printed tables are placed inside a div of class `.toc`, `.lof`, or `.lot`.

A “sidetoc” is provided which prints a subset of the TOC on the side of each page other than the homepage.

The regular \LaTeX infrastructure is used for TOC, along with some patches to generate HTML output.

for HTML output: 5527 `\begin{warpHTML}`

57.1 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 \LaTeX , the file is not reset after being read, since the TOC may be referred to again in each HTML page, and is used for the sideTOC.

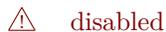
```
5528 \newcommand*{\LWR@myshorttoc}[1]{
5529 \LWR@ensuredoingapar
```

Only if the file exists:

```
5530 \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 `pdf \LaTeX` , also change to `latin1` encoding. When reading back a file with accented characters, the encoding change seems to be required, rather than leaving it `utf8`.

```

5531 \begingroup
5532 % \ifxetexorluatex%
5533 % \else
5534 % \inputencoding{latin1}% currently disabled
5535 % \fi
5536 \makeatletter

```

Read in the TOC file:

```

5537 \@input{\jobname.#1}
5538 % \makeatother
5539 \endgroup
5540 }%
5541 {}%
5542 }

```

`\LWR@subtableofcontents` $\{ \langle toc/lof/lot \rangle \} \{ \langle sectionstarname \rangle \}$

Places a TOC/LOF/LOT at the current position.

```

5543 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%

```

Closes previous levels:

```

5544 \@ifundefined{chapter}
5545 {\LWR@closeprevious{\LWR@depthsection}}
5546 {\LWR@closeprevious{\LWR@depthchapter}}

```

Prints any pending footnotes so that they appear above the potentially large TOC:

```

5547 \LWR@printpendingfootnotes

```

Place the list into its own chapter (if defined) or section:

```

5548 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}

```

Create a new HTML nav containing the TOC/LOF/LOT:

```

5549 \LWR@htmlclass{nav}{#1}

```

Create the actual list:

```

5550 \LWR@myshorttoc{#1}

```

Close the nav:

```

5551 \LWR@htmlclassend{nav}{#1}
5552 }

```

Patch `\@starttoc` to encapsulate the TOC inside HTML tags:

```
5553 \let\LWR@orig@starttoc\@starttoc
5554
5555 \renewcommand{\@starttoc}[1]{
5556 \LWR@htmlclass{nav}{#1}
5557 \LWR@orig@starttoc{#1}
5558 \LWR@htmlclassend{nav}{#1}
5559 }
```

Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.

```
5560 \let\LWR@origtableofcontents\tableofcontents
5561 \let\LWR@origlistoffigures\listoffigures
5562 \let\LWR@origlistoftables\listoftables
5563
5564 \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:

```
5565 \ifbool{FormatWordProcessor}{}{
```

Copy the `.toc` file to `.sidetoc` for printing the sideTOC. The original `.toc` file is renewed when `\tableofcontents` is finished.

```
5566 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
5567 \LWR@printpendingfootnotes
5568 \LWR@origtableofcontents
5569 }
5570 }
5571 \renewcommand*{\listoffigures}{
5572 \ifbool{FormatWordProcessor}{}{
5573 \LWR@printpendingfootnotes
5574 \LWR@origlistoffigures
5575 }
5576 }
5577
5578 \renewcommand*{\listoftables}{
5579 \ifbool{FormatWordProcessor}{}{
5580 \LWR@printpendingfootnotes
5581 \LWR@origlistoftables
5582 }
5583 }
```

57.2 High-level TOC commands

`\listof` $\langle type \rangle$ $\langle title \rangle$

Emulate the `\listof` command from the float package (section 103). Used to create lists of custom float types. Also used to redefine the standard L^AT_EX `\listoffigures` and `\listoftables` commands.

```
5584 \NewDocumentCommand{\listof}{m +m}{%
5585 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
5586 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
5587 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
5588     \jobname.\csuse{ext@#1}\relax
5589 }
```

57.3 Side TOC

The “side TOC” is a table-of-contents positioned to the side.

It may be renamed by redefining `\sidetocname`, and may contain paragraphs.

CSS may be used to format the sideTOC:

CSS related to sideTOC:

nav.sidetoc: The entire sideTOC.

div.sidetoctitle: The title.

div.sidetoccontents: The table of contents.

```
5590 \end{warpHTML}
```

for HTML & PRINT: 5591 `\begin{warpall}`

Ctrl SideTOCDepth Controls how deep the side-TOC gets. Use a standard L^AT_EX section level similar to `tocdepth`.

```
5592 \newcounter{SideTOCDepth}
5593 \setcounter{SideTOCDepth}{1}
```

`\sidetocname` Holds the default name for the sideTOC.

```
5594 \newcommand{\sidetocname}{Contents}
```

```
5595 \end{warpall}
```

for HTML output: 5596 \begin{warpHTML}

`\LWR@sidetoc` Creates the actual side-TOC.

```
5597 \newcommand*{\LWR@sidetoc}{
5598 \LWR@forcenewpage
5599 \LWR@stoppars
5600
```

The entire sideTOC is placed into a nav of class `sidetoc`.

```
5601 \LWR@htmlclass{nav}{sidetoc}
5602
5603 \setcounter{tocdepth}{\value{SideTOCDepth}}
5604
```

The title is placed into a div of class `sidetoc title`, and may contain paragraphs.

```
5605 \begin{BlockClass}{sidetoc title}
5606 \sidetocname
5607 \end{BlockClass}
```

The table of contents is placed into a div of class `sidetoc contents`.

```
5608 \begin{BlockClass}{sidetoc contents}
5609 \LinkHome
5610
5611 \LWR@myshorttoc{sidetoc}
5612 \end{BlockClass}
5613 \LWR@htmlclassend{nav}{sidetoc}
5614 }
```

57.4 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:

```
5615 \renewcommand*{\numberline}[1]{%
5616 \LWR@sectionnumber{#1}%
5617 }
```

`\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

```
5618 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
5619 \ifthenelse{\cinttest{#1}{<=} {\value{tocdepth}}}{%
```

```
5620 \LWR@startpars%
```

Create an HTML link to `filename#autosec-(page)`, with text of the caption, of the given HTML class.

```
5621 \LWR@subhyperrefclass{%
```

```
5622 \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
```

```
5623 \LWR@stoppars%
```

```
5624 }
```

```
5625 {}
```

```
5626 }
```

Ctr `lofdepth` TOC depth for figures.

```
5627 \newcounter{lofdepth}
```

```
5628 \setcounter{lofdepth}{1}
```

Ctr `lotdepth` TOC depth for tables.

```
5629 \newcounter{lotdepth}
```

```
5630 \setcounter{lotdepth}{1}
```

`\hypertocfloat` $\langle 1: depth \rangle$ $\langle 2: type \rangle$ $\langle 3: ext of parent \rangle$ $\langle 4: caption \rangle$ $\langle 5: page \rangle$

#1 is depth

#2 is figure, table, etc.

#3 is lof, lot, of the parent.

#4 the text of the caption

#5 page number

```
5631 \newcommand{\hypertocfloat}[5]{%
5632 \LWR@startpars
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
5633 \@ifundefined{c@#3depth}{%
5634 \newcounter{#3depth}%
5635 \setcounter{#3depth}{1}%
5636 }{}%
```

Respond to `lofdepth`, etc.:

```
5637 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
5638 \ifthenelse{\cnttest{#1}{<=}{\arabic{#3depth}}}{%
5639 \LWR@startpars%
```

Create an HTML link to `filename#autofloat-(float number)`, with text of the caption, of the given HTML class.

```
5640 \LWR@subhyperrefclass{%
5641 \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
5642 \#autofloat-\arabic{LWR@nextautofloat}}%
5643 {#4}{toc#2}%
5644 \LWR@stoppars%
5645 }{}%
5646 }
```

Automatically called by `\contentsline`:

```
5647 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
5648 \DeclareDocumentCommand{\l@chapter}{m m}
5649 {\hypertoc{0}{chapter}{#1}{#2}}
5650 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
5651 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
5652 \renewcommand{\l@subsubsection}[2]
5653 {\hypertoc{3}{subsubsection}{#1}{#2}}
5654 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
5655 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
5656 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
5657 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

5658 \end{warpHTML}
```

58 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: 5659 `\begin{warpHTML}`

```
5660 \newcounter{LWR@autoindex}
5661 \setcounter{LWR@autoindex}{0}
5662
5663 \newcounter{LWR@autoglossary}
5664 \setcounter{LWR@autoglossary}{0}
```

`\LWR@indexsection` Controls whether the index will be in a section or a chapter, depending on the documentclass.

```
5665 \@ifundefined{chapter}
5666 {\newcommand*\LWR@indexsection{\section{\indexname}}}
5667 {\newcommand*\LWR@indexsection{\chapter{\indexname}}}
```

`\printindex`

```
5668 \let\LWR@origprintindex\printindex
5669
5670 \renewcommand*\printindex
5671 {
5672 \LWR@indexsection
5673 \LWR@startpars
5674 \LWR@origprintindex
5675 }
```

Env `theindex`

```
5676 \renewenvironment*{theindex}{%
5677 \let\item\LWR@indexitem%
5678 \let\subitem\LWR@indexsubitem%
5679 \let\subsubitem\LWR@indexsubsubitem%
5680 }{}
```

`\LWR@indexitem`

```

5681 \newcommand{\LWR@indexitem}{
5682
5683 \InlineClass{indexitem}{}
5684 }

```

`\LWR@indexitem`

```

5685 \newcommand{\LWR@indexsubitem}{
5686
5687 \InlineClass{indexsubitem}{}
5688 }

```

`\LWR@indexitem`

```

5689 \newcommand{\LWR@indexsubsubitem}{
5690
5691 \InlineClass{indexsubsubitem}{}
5692 }

```

`\@wrindex` $\langle term \rangle$ Redefined to write the `LWR@latestautopage` counter instead of page

```

5693 \def\@wrindex#1{%
5694 \addtocounter{LWR@autoindex}{1}%
5695 \LWR@newlabel{LWRindex-\theLWR@autoindex}%
5696 \protected@write\@indexfile{}%
5697 {\string\indexentry{#1}{\theLWR@autoindex}}%
5698 \endgroup
5699 \@esphack}

```

`\@wrglossary` $\langle term \rangle$ Redefined to write the `LWR@latestautopage` counter instead of page

```

5700 \def\@wrglossary#1{%
5701 \addtocounter{LWR@autoglossary}{1}%
5702 \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
5703 \protected@write\@glossaryfile{}%
5704 {\string\glossaryentry{#1}{\theLWR@autoglossary}}%
5705 \endgroup
5706 \@esphack}

```

`\hyperindexref` $\langle autosecnumber \rangle$

`\hyperindexref{web address}` is inserted into *.ind by the xindy style file `lwarp.xdy`

```

5707 \newcommand*\hyperindexref[1]{\nameref{LWRindex-#1}}

```

```
5708 \end{warpHTML}
```

for PRINT output: A null command for print mode, in case `hyperref` was not used:

```
5709 \begin{warpprint}
5710 \newcommand{\hyperindexref}[1]{#1}
5711 \end{warpprint}
```

for HTML & PRINT: For the `glossaries` package, try to prevent an error where `\glo@name` was not found:

```
5712 \begin{warpall}
5713 \providecommand{\glo@name}{}
5714 \end{warpall}
```

59 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 font⁹, 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,

⁹See section 189 regarding fonts and fractions.

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 7 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

 **siunitx inside an equation**

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>

L^AT_EX macros

- MathJax does not automatically support custom L^AT_EX macros, but they may be set up by the user.

custom MathJax macros

As an example of using custom L^AT_EX macros with MathJax, place the following at the start of the document, after `\begin{document}`:

```
\begin{warpHTML} % Only for HTML output,
\ifbool{mathjax} % and only for MathJax output:
{
\(\ % New macros for MathJax are placed inside a math expression:
\newcommand{\expval}[1]{\langle#1\rangle}
\newcommand{\abs}[1]{\lvert#1\rvert}
\)
}{}
\end{warpHTML}
```

for HTML output: 5715 `\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.

```
5716 \let\LWR@origtextdollar\$
5717
5718 \renewcommand*{\$}{%
5719 \ifthenelse{\cinttest{\value{LWR@lateximagedepth}}{>}{0}}%
5720 {\LWR@origtextdollar}%
5721 {\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
5722 }
```

Ctrl LWR@externalfilecnt Counter for the external files which are generated and then referenced from the HTML:

```
5723 \newcounter{LWR@externalfilecnt}
```

59.1 Inline and display math

```

5724 \LetLtxMacro\LWR@origdollar$
5725 \LetLtxMacro\LWR@secondorigdollar$% balance for editor syntax highlighting

5726 \LetLtxMacro\LWR@origopenparen\(  

5727 \LetLtxMacro\LWR@origcloseparen\  

5728 \LetLtxMacro\LWR@origopenbracket\  

5729 \LetLtxMacro\LWR@origclosebracket\]

```

\$ Redefine the dollar sign to place math inside a `lateximage`, or use MathJax:
\$\$

```

5730 \begingroup
5731 \catcode'\$=\active%
5732 \protected\gdef$#{@ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

\LWR@doubledollar Redefine the double dollar sign to place math inside a `lateximage`, or use MathJax:

```

5733 \gdef\LWR@doubledollar$#1$$#{
5734 \ifbool{mathjax}%

```

For MathJax, print the math between `\[` and `\]`:

```

5735 {\textbackslash[\LWR@HTMLsanitize{#1}\textbackslash]}

```

For SVG, print the math inside a `lateximage`, with an ALT tag of the \LaTeX code:

```

5736 {% not mathjax
5737
5738 \begin{lateximage}%
5739 [\textbackslash{[] \LWR@HTMLsanitize{#1} \textbackslash{[]}]%
5740 \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
5741 \end{lateximage}
5742
5743 }
5744 }%

```

\LWR@singledollar Redefine the single dollar sign to place math inside a `lateximage`, or use MathJax:

```

5745 \gdef\LWR@subsingledollar#1{%
5746 \ifbool{mathjax}%

```

For MathJax, print the math between `\(` and `\)`:

```

5747 {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}

```

For SVG, print the math inside a `lateximage`, with an ALT tag of the \LaTeX code:

```

5748 {% not mathjax
5749 \begin{lateximage}%

```

```

5750 [\textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)]%
5751 \LWR@origdollar#1\LWR@origdollar%
5752 \end{lateximage}%
5753 }%
5754 }
5755
5756 \gdef\LWR@singledollar#1${%
5757 \LWR@subsingledollar{#1}%
5758 }

```

\(Redefine to the above dollar macros.
\)

```

5759 \gdef\(#1\){$#1$}
5760 \gdef\[#1\]{$$#1$$}
5761
5762 \endgroup
5763
5764
5765

```

`\@ensuredmath` $\{expression\}$ Not yet adapted to lwarp.

```

5766 \let\LWR@orig@ensuredmath\@ensuredmath
5767
5768 \newcommand*\LWR@ensuredmath}[1]{%
5769 \LWR@traceinfo{\@ensuredmath: page \thepage: \detokenize{#1}}%
5770 \LWR@subsingledollar{#1}%
5771 }

```

Remove the old `math` and `displaymath` environments:

```

5772 \let\math\relax
5773 \let\endmath\relax
5774 \let\displaymath\relax
5775 \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`.

```

5776 \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`.

```

5777 \NewEnviron{displaymath}{\expandafter[\BODY]\@ignoretrue}

```

59.2 MathJax support

Ctrl `LWR@nextequation` Used to add one to compute the next equation number.

```
5778 \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 .$$

```
5779 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
5780 \ifcsdef{thechapter}{
5781 \InlineClass{hidden}{
5782 \textbackslash(
5783 \textbackslash{}seteqsection \{\thechapter\}
5784 \textbackslash)
5785 }
5786 }
5787 }{}% not using chapters
```

MathJax doesn’t allow setting the equation number to 1:

```
5788 \ifthenelse{\cnttest{\value{equation}}>0}
5789 {
```

Tell MathJax that the next set of equations begins with the current \LaTeX equation number, plus one.

```
5790 \setcounter{LWR@nextequation}{\value{equation}}
5791 \addtocounter{LWR@nextequation}{1}
```

Place the MathJax command inside “\($\) and “\($\)” characters, to be printed to HTML, not interpreted by \LaTeX .$$

```
5792 \InlineClass{hidden}{
5793 \textbackslash(
5794 \textbackslash{}seteqnumber \{\arabic{LWR@nextequation}\}
5795 \textbackslash)
5796 }
5797 }{}% not eq > 1
5798 }
```

`\LWR@restoreorigformatting` While producing SVG math or a `lateximage`, use regular \LaTeX formatting instead of HTML tags.

```

5799 \newcommand*{\LWR@restoreorigformatting}{%
5800 \LetLtxMacro{\hspace}{\LWR@orighspace}%
5801 \LetLtxMacro\rule{\LWR@origrule}%
5802 \let\,\LWR@origcomma% disable HTML short unbreakable space
5803 \LetLtxMacro{\textrm}{\LWR@origtextrm}%
5804 \LetLtxMacro{\textsf}{\LWR@origtextsf}%
5805 \LetLtxMacro{\texttt}{\LWR@origtexttt}%
5806 \LetLtxMacro{\textbf}{\LWR@origtextbf}%
5807 \LetLtxMacro{\textmd}{\LWR@origtextmd}%
5808 \LetLtxMacro{\textit}{\LWR@origtextit}%
5809 \LetLtxMacro{\textsl}{\LWR@origtextsl}%
5810 \LetLtxMacro{\textsc}{\LWR@origtextsc}%
5811 \LetLtxMacro{\textup}{\LWR@origtextup}%
5812 \LetLtxMacro{\textnormal}{\LWR@origtextnormal}%
5813 \LetLtxMacro{\emph}{\LWR@origemph}%
5814 \LetLtxMacro{\rmfamily}{\LWR@origrmfamily}%
5815 \LetLtxMacro{\sffamily}{\LWR@origsffamily}%
5816 \LetLtxMacro{\ttfamily}{\LWR@origttfamily}%
5817 \LetLtxMacro{\bfseries}{\LWR@origbfseries}%
5818 \LetLtxMacro{\mdseries}{\LWR@origmdseries}%
5819 \LetLtxMacro{\upshape}{\LWR@origupshape}%
5820 \LetLtxMacro{\slshape}{\LWR@origslshape}%
5821 \LetLtxMacro{\scshape}{\LWR@origscshape}%
5822 \LetLtxMacro{\itshape}{\LWR@origitshape}%
5823 \LetLtxMacro{\em}{\LWR@origem}%
5824 \LetLtxMacro{\normalfont}{\LWR@orignormalfont}%
5825 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
5826 \let\textsuperscript\LWR@origtextsuperscript%
5827 \let\textsubscript\LWR@origtextsubscript%
5828 \let-\LWR@origtilde%
5829 \let\enskip\LWR@origenskip%
5830 \let\quad\LWR@origquad%
5831 \let\qquad\LWR@origqquad%
5832 \LetLtxMacro{\tabular}{\LWR@origtabular}%
5833 \LetLtxMacro{\endtabular}{\LWR@origendtabular}%
5834 \LetLtxMacro\toprule{\LWR@origtoprule}%
5835 \LetLtxMacro\midrule{\LWR@origmidrule}%
5836 \LetLtxMacro\cmidrule{\LWR@origcmidrule}%
5837 \LetLtxMacro\bottomrule{\LWR@origbottomrule}%
5838 \LetLtxMacro\addlinespace{\LWR@origaddlinespace}%
5839 \LetLtxMacro\morecmidrules{\LWR@origmorecmidrules}%
5840 \LetLtxMacro\specialrule{\LWR@origspecialrule}%
5841 \let\newline\LWR@orignewline%
5842 \LetLtxMacro{\raisebox}{\LWR@origraisebox}%
5843 \LetLtxMacro{\scalebox}{\LWR@origscalebox}%
5844 \LetLtxMacro{\rotatebox}{\LWR@origrotatebox}%
5845 \let\reflectbox\LWR@origreflectbox%

```

```

5846 \let\framebox\LWR@origframebox%
5847 \let\makebox\LWR@origmakebox%
5848 \let\fbbox\LWR@origfbbox%
5849 \let\fbboxBlock\LWR@origfbbox%
5850 \LetLtxMacro{\fminipage}{\LWR@fminipageprint}%
5851 \LetLtxMacro{\endfminipage}{\endLWR@fminipageprint}%
5852 \LetLtxMacro{\minipage}{\LWR@origminipage}%
5853 \let\endminipage\LWR@endminipage%
5854 \LetLtxMacro{\parbox}{\LWR@origparbox}%
5855 }

```

`\LWR@hidelatexequation` $\{ \langle environment \rangle \} \{ \langle contents \rangle \}$

Creates the L^AT_EX version of the equation inside an HTML comment.

```
5856 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```

5857 \LWR@stoppars
5858 \LWR@htmlopencomment
5859

```

Start the L^AT_EX math environment inside the HTML comment:

```

5860 \begingroup
5861 \csuse{LWR@orig#1}

```

While in the math environment, restore various commands to their L^AT_EX meanings.

```
5862 \LWR@restoreorigformatting
```

See `\LWR@htmlmathlabel` in section [59.4.1](#).

Print the contents of the equation:

```
5863 #2
```

End the L^AT_EX math environment inside the HTML comment:

```

5864 \csuse{LWR@origend#1}
5865 \endgroup
5866

```

Close the HTML comment and resume HTML paragraph handling:

```

5867 \LWR@htmlclosecomment
5868 \LWR@startpars
5869 }

```

`\LWR@addmathjax` $\{ \langle environment \rangle \} \{ \langle contents \rangle \}$

Given the name of a math environment and its contents, create a MathJax instance. The contents are printed to HTML output, not interpreted by L^AT_EX.

```
5870 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
```

Enclose the MathJax environment inside printed “\ (“ and “\)” characters.

```
5871 \LWR@origtilde\LWR@orignewline
5872 \textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
5873 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
5874 \textbackslash{}end\{#1\}
5875 \LWR@orignewline
5876 }
```

59.3 Equation environment

Remember existing `equation` environment:

```
5877 \let\LWR@origequation\equation
5878 \let\LWR@origendequation\endequation
```

Remove existing `equation` environment:

```
5879 \let\equation\relax
5880 \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 L^AT_EX 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.

```
5881 \NewEnviron{equation}{%
```

```
5882
5883 \ifbool{mathjax}
```

MathJax output:

```
5884 {
```

Print commands to synchronize MathJax's equation number and format to the current L^AT_EX chapter/section and equation number:

```
5885 \LWR@syncmathjax
```

Print the L^AT_EX math inside an HTML comment:

```
5886 \LWR@hidelatexequation{equation}{\BODY}
5887 }
```

SVG output: Create the `lateximage` along with an HTML ALT tag having an equation number, the L^AT_EX equation environment commands, and the contents of the environment's `\BODY`.

```
5888 {% not mathjax
```

Begin the `lateximage` with an ALT tag containing the math source:

```
5889 \begin{lateximage}[(\theequation) \textbackslash{begin\{equation\}}]%
5890 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}}%
5891 \textbackslash{end\{equation\}}]% alt tag
```

Create the actual L^AT_EX-formatted equation inside the `lateximage` using the contents of the environment.

```
5892 \LWR@origequation
5893 \BODY% contents collected by NewEnviron
5894 \LWR@origendequation
5895 \end{lateximage}%
5896 }
5897
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
5898 }[\ifbool{mathjax}{\LWR@addmathjax{equation}{\BODY}}{]}
```

59.4 AMS Math environments

59.4.1 Support macros

Bool `LWR@amsmultiline` True if processing a multiline environment.

To compensate for `multiline`-specific code, `LWR@amsmultiline` is used to add extra horizontal space in `\LWR@htmlmathlabel` if is used in an `amsmath` environment which is not a `multiline` environment and not an equation.

```
5899 \newbool{LWR@amsmultiline}
5900 \boolfalse{LWR@amsmultiline}
```

`\LWR@htmlmathlabel` $\langle label \rangle$

`lwarp` points `\ltx@label` here. This is used by `\label` when inside a \LaTeX AMS math environment's math display environment.

`\LWR@origltx@label` points to the \LaTeX original, modified by `lwarp`, then by `amsmath`, then by `cleveref`.

```
5901 \newcommand*{\LWR@htmlmathlabel}[1]{%
5902 \LWR@traceinfo{\LWR@htmlmathlabel #1}%
5903 \ifbool{mathjax}{%
```

The combined \LaTeX & HTML label is printed in a `\text` field:

```
5904 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
5905 \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.)

```
5906 \LWR@htmlclosecomment%
5907 \LWR@origltx@label{#1}%
5908 \LWR@htmlopencomment%
5909 }% text
5910 }% mathjax
5911 {%
5912 \LWR@origltx@label{#1}%
5913 }%
5914 }
```

`\LWR@beginhideamsmath` Starts hiding L^AT_EX math inside an HTML comment.

```
5915 \newcommand*{\LWR@beginhideamsmath}{
5916 \LWR@stoppars
5917 \LWR@origtilde\LWR@orignewline
5918 \LWR@htmlopencomment
5919
5920 \begingroup
5921 \LWR@restoreorigformatting
5922 }
```

`\LWR@endhideamsmath` Ends hiding L^AT_EX math inside an HTML comment.

```
5923 \newcommand*{\LWR@endhideamsmath}{
5924 \endgroup
5925
5926 \LWR@htmlclosecomment
5927 \LWR@orignewline
5928 \LWR@startpars
5929 }
```

59.4.2 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 L^AT_EX's equation numbers, typeset with L^AT_EX 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`.

```
5930 \let\LWR@origeqnarray\eqnarray
5931 \let\LWR@origendeqnarray\endeqnarray
```

To remember whether the starred environment was used, and thus whether to number the equations:

```
5932 \newbool{LWR@numbereqnarray}
5933 \booltrue{LWR@numbereqnarray}
```

Common code used by `eqnarray` and `Beqnarray` (from `fancybox`):

```
5934 \newcommand{\LWR@eqnarrayfactor}{%
```

Different behavior depending on MathJax vs SVG math:

```
5935 \ifbool{mathjax}
5936 {
```

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:

```
5937 \LWR@syncmathjax
5938 \boolfalse{\LWR@amsmultline}
5939 \ifbool{\LWR@numbreqnarray}
5940 {
```

If numbering the equations, execute a copy inside an HTML comment block:

```
5941 \LWR@beginhideamsmath
5942 \LWR@origeqnarray
5943 \BODY
5944 \LWR@origendeqnarray
5945 \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MathJax to interpret:

```
5946 \LWR@addmathjax{eqnarray}{\BODY}
5947 }{% not LWR@numbreqnarray
```

If not numbering equations, just create the contents for MathJax:

```
5948 \LWR@addmathjax{eqnarray*}{\BODY}
5949 }% LWR@numbreqnarray
5950 }% mathjax
5951 {% not mathjax
5952 \ifbool{\LWR@numbreqnarray}
5953 {
```

For numbered SVG equations, first create a `lateximage` with an `alt` attribute containing sanitized copy of the source code:

```
5954 \begin{lateximage}[\LWR@addmathjax{eqnarray}{\BODY}]
```

Then create the image contents using an actual `eqnarray`:

```
5955 \LWR@origeqnarray
```

```

5956 \BODY
5957 \LWR@origendeqnarray
5958 \end{lateximage}
5959 }{% not LWR@numbreqnarray

```

If not numbered, do the same, but an extra `\nonumber` seems to be required:

```

5960 \begin{lateximage}[\LWR@addmathjax{eqnarray*}]{\BODY}
5961 \LWR@origeqnarray
5962 \BODY
5963 \nonumber
5964 \LWR@origendeqnarray
5965 \end{lateximage}
5966 }% LWR@numbreqnarray
5967 }% not mathjax

```

Default to number equations in the future:

```

5968 \booltrue{LWR@numbreqnarray}
5969 }

```

`eqnarray` itself is made with a blank line before and after to force it to be on its own line:

```

5970 \RenewEnviron{eqnarray}
5971 {%
5972
5973 \LWR@eqnarrayfactor
5974
5975 }

```

The starred version is patched to turn off the numbering:

```

5976 \csgpreto{eqnarray*}{\boolfalse{LWR@numbreqnarray}}

```

The following $\mathcal{A}\mathcal{M}\mathcal{S}$ environments are more easily patched in-place:

Env `multline`

```

5977 \BeforeBeginEnvironment{multline}{
5978
5979 \ifbool{mathjax}
5980 {
5981 \LWR@syncmathjax
5982 \booltrue{LWR@amsmultline}
5983 \LWR@beginhideamsmath
5984 }
5985 {

```

```
5986 \lateximage[LWRAMSMATHBODY]
5987 }
5988 }
5989
5990 \AfterEndEnvironment{multline}{
5991
5992 \ifbool{mathjax}
5993 {
5994 \LWR@endhideamsmath
5995 \boolfalse{LWR@amsmultline}
5996 \LWR@addmathjax{multline}{\the\@envbody}
5997 }
5998 {\endlateximage}
5999
6000 }
```

Env **multline***

```
6001 \BeforeBeginEnvironment{multline*}{
6002
6003 \ifbool{mathjax}
6004 {
6005 \LWR@syncmathjax
6006 \booltrue{LWR@amsmultline}
6007 \LWR@beginhideamsmath
6008 }
6009 {
6010 \lateximage[LWRAMSMATHBODY]
6011 }
6012 }
6013
6014 \AfterEndEnvironment{multline*}{
6015
6016 \ifbool{mathjax}
6017 {
6018 \LWR@endhideamsmath
6019 \boolfalse{LWR@amsmultline}
6020 \LWR@addmathjax{multline*}{\the\@envbody}
6021 }
6022 {\endlateximage}
6023
6024 }
6025 }
```

Env **gather**

```
6026 \BeforeBeginEnvironment{gather}{
6027
6028 \ifbool{mathjax}
```

```

6029 {
6030 \LWR@syncmathjax
6031 \boolfalse{LWR@amsmultiline}
6032 \LWR@beginhideamsmath
6033 }
6034 {
6035 \lateximage[LWRAMSMATHBODY]
6036 }
6037 }
6038
6039 \AfterEndEnvironment{gather}{
6040
6041 \ifbool{mathjax}
6042 {
6043 \LWR@endhideamsmath
6044 \LWR@addmathjax{gather}{\the\@envbody}
6045 }
6046 {\endlateximage}
6047
6048 }

```

Env **gather***

```

6049 \BeforeBeginEnvironment{gather*}{
6050
6051 \ifbool{mathjax}
6052 {
6053 \LWR@syncmathjax
6054 \boolfalse{LWR@amsmultiline}
6055 \LWR@beginhideamsmath
6056 }
6057 {
6058 \lateximage[LWRAMSMATHBODY]
6059 }
6060 }
6061
6062 \AfterEndEnvironment{gather*}{
6063
6064 \ifbool{mathjax}
6065 {
6066 \LWR@endhideamsmath
6067 \LWR@addmathjax{gather*}{\the\@envbody}
6068 }
6069 {\endlateximage}
6070
6071 }

```

Env `align`

```
6072 \BeforeBeginEnvironment{align}{
6073
6074 \ifbool{mathjax}
6075 {
6076 \LWR@syncmathjax
6077 \boolfalse{LWR@amsmultiline}
6078 \LWR@beginhideamsmath
6079 }
6080 {
6081 \lateximage[LWRAMSMATHBODY]
6082 }
6083 }
6084
6085 \AfterEndEnvironment{align}{
6086
6087 \ifbool{mathjax}
6088 {
6089 \LWR@endhideamsmath
6090 \LWR@addmathjax{align}{\the\@envbody}
6091 }
6092 {\endlateximage}
6093
6094 }
```

Env `align*`

```
6095 \BeforeBeginEnvironment{align*}{
6096
6097 \ifbool{mathjax}
6098 {
6099 \LWR@syncmathjax
6100 \boolfalse{LWR@amsmultiline}
6101 \LWR@beginhideamsmath
6102 }
6103 {
6104 \lateximage[LWRAMSMATHBODY]
6105 }
6106 }
6107
6108 \AfterEndEnvironment{align*}{
6109
6110 \ifbool{mathjax}
6111 {
6112 \LWR@endhideamsmath
6113 \LWR@addmathjax{align*}{\the\@envbody}
6114 }
6115 {\endlateximage}
```

```
6116
6117 }
```

Env **flalign**

```
6118 \BeforeBeginEnvironment{flalign}{
6119
6120 \ifbool{mathjax}
6121 {
6122 \LWR@syncmathjax
6123 \boolfalse{LWR@amsmultline}
6124 \LWR@beginhideamsmath
6125 }
6126 {
6127 \lateximage[LWRAMSMATHBODY]
6128 }
6129 }
6130
6131 \AfterEndEnvironment{flalign}{
6132
6133 \ifbool{mathjax}
6134 {
6135 \LWR@endhideamsmath
6136 \LWR@addmathjax{flalign}{\the\@envbody}
6137 }
6138 {\endlateximage}
6139
6140 }
```

Env **flalign***

```
6141 \BeforeBeginEnvironment{flalign*}{
6142
6143 \ifbool{mathjax}
6144 {
6145 \LWR@syncmathjax
6146 \boolfalse{LWR@amsmultline}
6147 \LWR@beginhideamsmath
6148 }
6149 {
6150 \lateximage[LWRAMSMATHBODY]
6151 }
6152 }
6153
6154 \AfterEndEnvironment{flalign*}{
6155
6156 \ifbool{mathjax}
6157 {
6158 \LWR@endhideamsmath
```

```

6159 \LWR@addmathjax{flalign*}{\the\@envbody}
6160 }
6161 {\endlateximage}
6162
6163 }

6164 \end{warpHTML}

```

60 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: `6165 \begin{warpHTML}`

Ctrl `LWR@lateximagenumber` Sequence the images.

```

6166 \newcounter{LWR@lateximagenumber}
6167 \setcounter{LWR@lateximagenumber}{0}

```

Ctrl `LWR@lateximagedepth` Do not create `\lateximage` inside of `\lateximage`.

```

6168 \newcounter{LWR@lateximagedepth}
6169 \setcounter{LWR@lateximagedepth}{0}

```

Declare the `\LWR@file` for writing to generate file `lateximages.txt`:

```
6170 \ifcsdef{LWR@file}{\newwrite{\LWR@file}}
```

A few utility macros to write special characters:

```
6171 \edef\LWR@hashmark{\string#} % for use in \write
6172 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctrl **LWR@Lpage** Used to reference the PDF page number of a lateximage to be written into lateximages.txt.

```
6173 \newcounter{LWR@Lpage}
```

```
6174 \end{warpHTML}
```

for HTML & PRINT: 6175 \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.

```
6176 \newcommand*{\LateximageFontSizeName}{large}
```

```
6177 \end{warpall}
```

for HTML output: 6178 \begin{warpHTML}

\LWR@HTMLsanitize $\{ \langle text \rangle \}$

Math expressions are converted to lateximages, and some math environments may contain “&”, “<”, or “>”, which should not be allowed inside an HTML ALT tag, so must convert them to HTML entities.

Two versions follow, depending on expansion needs. There may be a better way...

```
6179 \newcommand{\LWR@HTMLsanitize}[1]{%
6180 \protect\StrSubstitute{\detokenize{#1}}%
6181 {\detokenize{&}}%
6182 {\detokenize{&}}[\LWR@strresult]%
6183 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6184 {\detokenize{<}}%
6185 {\detokenize{<}}[\LWR@strresult]%
6186 [\LWR@strresult]%
6187 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6188 {\detokenize{>}}%
6189 {\detokenize{>}}[\LWR@strresult]%
6190 [\LWR@strresult]%
6191 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6192 {\detokenize{##}}%
6193 {\#}%
6194 [\LWR@strresult]%
```

```
6195 \LWR@strresult%
6196 }
```

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

This version expands the argument before sanitizing it.

```
6197 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
6198 \protect\StrSubstitute{\detokenize\expandafter{#1}}%
6199 {\detokenize{&}}%
6200 {\detokenize{&}}%
6201 [\LWR@strresult]%
6202 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6203 {\detokenize{<}}%
6204 {\detokenize{&lt;}}%
6205 [\LWR@strresult]%
6206 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6207 {\detokenize{>}}%
6208 {\detokenize{&gt;}}%
6209 [\LWR@strresult]%
6210 \LWR@strresult%
6211 }
```

```
Env lateximage [⟨alttag⟩]
```

```
6212 \catcode'\$=\active%
6213
6214 \NewDocumentEnvironment{lateximage}{0{image}}{%
6215 \LWR@traceinfo{lateximage: starting on page \arabic{page}}%
6216 \ifthenelse{\cinttest{\value{LWR@lateximagedepth}}{>}{0}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
6217 {%
6218 \addtocounter{LWR@lateximagedepth}{1}%
6219 }%
```

Otherwise, this is the outer-most lateximage:

```
6220 {% start of outer-most lateximage
```

Starting a new lateximage:

```
6221 \addtocounter{LWR@lateximagenumber}{1}%
6222 \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%
```

While inside a lateximage, locally do not use mathjax:

6223 `\boolfalse{mathjax}`

Be sure that are doing a paragraph:

6224 `\LWR@ensuredoingapar%`

Next file:

6225 `\addtocounter{LWR@externalfilecnt}{1}%`

6226 `\LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%`

Figure out what the next page number will be:

6227 `\setcounterpageref{LWR@LPage}{LWRlateximage\theLWR@lateximagenumber}%`

6228 `\LWR@traceinfo{lateximage: LWR@LPage is \arabic{LWR@LPage}}%`

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:

6229 `\LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %`

6230 `class="lateximagesource"{} } \LWR@orignewline`

Write instructions to the `lateximages.txt` file:

6231 `\LWR@traceinfo{lateximage: about to write to lateximages.txt}%`

6232 `\immediate\write\LWR@file{|\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`.

6233 `\LWR@traceinfo{lateximage: about to create open comment}%`

6234 `\LWR@htmlopencomment%`

One level deeper:

6235 `\addtocounter{LWR@lateximagedepth}{1}%`

Start the new PDF page:

6236 `\LWR@traceinfo{lateximage: about to create new page}%`

6237 `\LWR@orignewpage%`

Typeset the image in a “standard” width page and font size:

6238 `\LWR@traceinfo{lateximage: about to create minipage}%`

6239 `\LWR@origminipage{6in}%`

6240 `\csuse{LWR@orig\LateximageFontSizeName}%`

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for `\hspace`, etc. inside a `lateximage`.

```

6241 \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
6242 \LWR@restoreorigformatting%
6243 \LWR@traceinfo{lateximage: about to create label}%
6244 \LWR@origlabel{\LWR@lateximage\arabic{\LWR@lateximagenumber}}%
6245 \LWR@traceinfo{lateximage: finished creating the label}%
6246 \LetLtxMacro$\LWR@origdollar%
6247 \catcode'\$=3% math shift
6248 \LetLtxMacro\(\LWR@origopenparen%
6249 \LetLtxMacro\)\LWR@origcloseparen%
6250 \LetLtxMacro\[\LWR@origopenbracket%
6251 \LetLtxMacro\]\LWR@origclosebracket%
6252 \let\@ensuredmath\LWR@orig@ensuredmath%
6253 }% end of outer-most lateximage
6254 \LWR@traceinfo{lateximage: finished start of environment}%
6255 }% end of \begin{lateximage}
6256 {% start of \end{lateximage}
6257 \LWR@traceinfo{lateximage: starting end of environment}%
6258 \ifthenelse{\cntttest{\value{\LWR@lateximagedepth}}{>}{1}}%

```

If nesting inside an already-existing `lateximage`, simply record one more level:

```

6259 {%
6260 \addtocounter{\LWR@lateximagedepth}{-1}%
6261 }%

```

if this is the outer-most `lateximage`:

```

6262 {% end of outer-most lateximage

```

Finish the `lateximage` minipage and start a new PDF page:

```

6263 \LWR@origendminipage%
6264 \LWR@orignewpage%
6265 \LWR@origscriptsize%

```

Close the HTML comment which encapsulated any traces of the `lateximage` picked up by `pdftotext`:

```

6266 \LWR@htmlclosecomment{\LWR@orignewline%
6267 \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` `multline`, `gather`, or `align` environments.

```

6268 \ifthenelse{\equal{#1}{LWRAMSMATHBODY}}%
6269 {%
6270 \LWR@subinlineimage[%
6271 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}}%
6272 ]%
6273 {lateximage}%
6274 {lateximages\OSPathSymbol{}}lateximage-\theLWR@externalfilecnt}%
6275 {svg}%
6276 {}%
6277 }%
6278 {%
6279 \LWR@subinlineimage[#1]{lateximage}%
6280 {lateximages\OSPathSymbol{}}lateximage-\theLWR@externalfilecnt}{svg}{}%
6281 }%
6282 % \LWR@orignewline% Removed to prevent extra space.

```

Be sure that are doing a paragraph:

```
6283 \LWR@ensuredoingapar%
```

Close the HTML span which has the pdftotext comment and also the link to the .svg image:

```

6284 \LWR@htmltag{/span}%
6285 \ifbool{HTMLDebugComments}{%
6286 \LWR@htmlcomment{End of lateximage}%
6287 }{}%
6288 % \LWR@orignewline% Removed to prevent extra space.

```

Undo one lateximage level:

```

6289 \addtocounter{LWR@lateximagedepth}{-1}%
6290 }% end of outer-most lateximage
6291 \LWR@traceinfo{lateximage: done}%
6292 }%
6293 \catcode'\$=3% math shift
6294 \end{warpHTML}

```

```

for PRINT output: 6295 \begin{warpprint}
6296 % \newenvironment{lateximage}[1][\minipage{\linewidth}]{\endminipage}
6297 \newenvironment{lateximage}[1][\minipage{\linewidth}]{\endminipage}
6298 \end{warpprint}

```

61 center, flushleft, flushright

```
for HTML output: 6299 \begin{warpHTML}
```

Env **center** Replace **center** functionality with CSS tags:

```
6300 \renewenvironment*{center}
6301 {
6302 \LWR@forcenewpage
6303 \BlockClass{center}
6304 }
6305 {\endBlockClass}
```

Env **flushright**

```
6306 \renewenvironment*{flushright}
6307 {
6308 \LWR@forcenewpage
6309 \BlockClass{flushright}
6310 }
6311 {\endBlockClass}
```

Env **flushleft**

```
6312 \renewenvironment*{flushleft}
6313 {
6314 \LWR@forcenewpage
6315 \BlockClass{flushleft}
6316 }
6317 {\endBlockClass}

6318 \end{warpHTML}
```

62 Siunitx

Pkg **siunitx**



per-mode

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

for HTML output: 6319 \begin{warpHTML}

Options for `siunitx`:

```
6320 \PassOptionsToPackage{
6321 detect-mode=true,
6322 per-mode=symbol,% fraction is not seen by pdftotext
6323 text-celsius = {\HTMLentity{deg}C},
6324 text-degree = {\HTMLentity{deg}},
6325 }{siunitx}
```

6326 `\end{warpHTML}`

63 Graphics

Pkg `graphics`

Pkg `graphicx`

- ⚠ **graphics vs. graphicx** If using the older `graphics` syntax, use both optional arguments for `\includegraphics`. A single optional parameter is interpreted as the newer `graphicx` syntax. Note that viewports are not supported by `warp`; the entire image will be shown.
- ⚠ **viewports**
- ⚠ **`\graphicspath`** `\graphicspath` only works for a single directory; all graphics must be in this directory.
 - 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`. Using 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.
- ⚠ **image file types** For `\includegraphics` the user should provide both `.pdf` and `.svg` images, but always refer to `.pdf` images in the document source. All `\includegraphics` references to `.pdf` will automatically be changed to `.svg` for HTML output, and will be left as `.pdf` for print output. Images may also be `.jpg` and `.png`, and will be used as-is for either output.
- `\rotatebox`** `\rotatebox` accepts the optional `origin` key.
- ⚠ **browser support** `\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The CSS3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike \LaTeX , so expect some ugly results for scaling and rotating.

for HTML output: 6327 `\begin{warpHTML}`

63.1 `\graphicspath`

`\graphicspath` $\{\langle path \rangle\}$

6328 `\newcommand*\thisgraphicspath{\}`

6329 `\renewcommand*\graphicspath[1]{\renewcommand*\thisgraphicspath{\#1}}`

`\DeclareGraphicsExtensions` $\{\langle list \rangle\}$

`\DeclareGraphicsRule` $\{\langle \rangle\} \{\langle \rangle\} \{\langle \rangle\} \{\langle \rangle\}$

6330 `\renewcommand*\DeclareGraphicsExtensions[1]{}`

6331 `\renewcommand*\DeclareGraphicsRule[4]{}`

63.2 Length conversions and graphics options



whitespace

A scaled image in L^AT_EX 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.

6332 `\renewcommand*\unitspace{}`

Used to store the user’s selected dimensions and HTML class.

The class defaults to “inlineimage” unless changed by a `class=xyx` option.

6333 `\newlength{\LWR@igwidth}`

6334 `\newlength{\LWR@igheight}`

6335 `\newcommand*\LWR@igwidthstyle{}`

6336 `\newcommand*\LWR@igheightstyle{}`

6337 `\newcommand*\LWR@igorigin{}`

6338 `\newcommand*\LWR@igangle{}`

6339 `\newcommand*\LWR@igxscale{1}`

6340 `\newcommand*\LWR@igyyscale{1}`

6341 `\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:

6342 `\define@key{igraph}{width}{%`

6343 `\setlength{\LWR@igwidth}{\#1}%`

6344 `\ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}%`

6345 `{%`

Default to use the converted fixed length given:

```
6346 \uselengthunit{PT}%
6347 \renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%
```

If ex or em dimensions were given, use those instead:

```
6348 \IfEndWith{#1}{ex}%
6349 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes ex
6350 {}% not ex
6351 \IfEndWith{#1}{em}%
6352 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes em
6353 {}% not em
6354 \IfEndWith{#1}{\}%
6355 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes percent
6356 {}% not percent
6357 \IfEndWith{#1}{px}%
6358 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
6359 {}% not px
6360 }{}% end of length > Opt
6361 }
```

If an optional height was given, set an HTML style:

```
6362 \define@key{igraph}{height}{%
6363 \setlength{\LWR@igheight}{#1}%
6364 \ifthenelse{\lengthtest{\LWR@igheight > Opt}}%
6365 {%
```

Default to use the converted fixed length given:

```
6366 \uselengthunit{PT}%
6367 \renewcommand*{\LWR@igheightstyle}{%
6368 height:\rndprintlength{\LWR@igheight} %
6369 }%
```

If ex or em dimensions were given, use those instead:

```
6370 \IfEndWith{#1}{ex}%
6371 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
6372 {}% not ex
6373 \IfEndWith{#1}{em}%
6374 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
6375 {}% not em
6376 \IfEndWith{#1}{\}%
6377 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
6378 {}% not percent
6379 \IfEndWith{#1}{px}%
6380 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
```

```
6381 {}% not px
6382 }{}% end of length > Opt
6383 }
```

Handle origin key:

```
6384 \define@key{igraph}{origin}{%
6385 \renewcommand*\LWR@igorigin}{#1}%
6386 }
```

Handle angle key:

```
6387 \define@key{igraph}{angle}{\renewcommand*\LWR@igangle}{#1}}
```

Handle class key:

```
6388 \define@key{igraph}{class}{\renewcommand*\LWR@igclass}{#1}}
6389
```

It appears that graphicx does not have separate keys for xscale and yscale. scale adjusts both at the same time.

```
6390 \define@key{igraph}{scale}{%
6391 \renewcommand*\LWR@igxscale}{#1}%
6392 \renewcommand*\LWR@igyyscale}{#1}}
```

Numerous ignored keys:

```
6393 \define@key{igraph}{bb}{}
6394 \define@key{igraph}{bbllx}{}
6395 \define@key{igraph}{bbllx}{}
6396 \define@key{igraph}{bburx}{}
6397 \define@key{igraph}{bbury}{}
6398 \define@key{igraph}{natwidth}{}
6399 \define@key{igraph}{natheight}{}
6400 \define@key{igraph}{hiresbb}{}
6401 \define@key{igraph}{viewport}{}
6402 \define@key{igraph}{trim}{}
6403 \define@key{igraph}{totalheight}{}
6404 \define@key{igraph}{keepaspectratio}{}
6405 \define@key{igraph}{clip}{}
6406 \define@key{igraph}{draft}{}
6407 \define@key{igraph}{type}{}
6408 \define@key{igraph}{ext}{}
6409 \define@key{igraph}{read}{}
6410 \define@key{igraph}{command}{}

```

```
\LWR@rotstyle {<prefix>} {<degrees>}
```

Prints the rotate style with the given prefix.

prefix is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:rotate` style.

```
6411 \newcommand*\LWR@rotstyle}[2]{%
6412 #1transform:rotate(-#2deg);
6413 }
```

`\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.

```
6414 \newcommand*\LWR@scalestyle}[3]{%
6415 #1transform:scale(#2,#3);
6416 }
```

63.3 `\includegraphics`

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```
6417 \newbool{LWR@infloatrow}
6418 \boolfalse{LWR@infloatrow}
```

```
6419 \newcommand*\LWR@imageextension{}
6420 \newcommand*\LWR@expgraphicsfilename{}
```

`\LWR@includegraphicsb` * [$\langle 2: options \rangle$] [$\langle 3: options \rangle$] [$\langle 4: filename \rangle$]

graphics syntax is `\includegraphics` * [$\langle llx, lly \rangle$] [$\langle urx, ury \rangle$] [$\langle file \rangle$]

graphicx syntax is `\includegraphics` [$\langle key values \rangle$] [$\langle file \rangle$]

If #3 is empty, only one optional argument was given, thus `graphicx` syntax.

```
6421 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
6422 {%
```

Start the image tag on a new line, allow PDF output word wrap:

```
6423 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
6424 \ifthenelse{\cinttest{\value{LWR@minipagedepth}}{=}={0}}{%
6425 \ifbool{LWR@infloatrow}%
6426 {}
6427 {% not in a minipage or a floatrow:
6428 \setlength{\linewidth}{6in}%
6429 \setlength{\textwidth}{6in}%
6430 \setlength{\textheight}{9in}%
6431 }%
6432 }{}%
```

See if can find the image by adding an extension:

Preference is `svgz`, then `svg`, `gif`, `png`, and `jpg`.

`\detokenize\expandafter` allows underscore characters in filenames.

```
6433 \edef\LWR@expgraphicsfilename{#4}
6434 \renewcommand*{\LWR@imageextension}{}%
6435 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.jpg}%
6436 {\renewcommand*{\LWR@imageextension}{.jpg}}{}%
6437 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.JPG}%
6438 {\renewcommand*{\LWR@imageextension}{.JPG}}{}%
6439 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.png}%
6440 {\renewcommand*{\LWR@imageextension}{.png}}{}%
6441 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.PNG}%
6442 {\renewcommand*{\LWR@imageextension}{.PNG}}{}%
6443 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.gif}%
6444 {\renewcommand*{\LWR@imageextension}{.gif}}{}%
6445 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.GIF}%
6446 {\renewcommand*{\LWR@imageextension}{.GIF}}{}%
6447 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svg}%
6448 {\renewcommand*{\LWR@imageextension}{.svg}}{}%
6449 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVG}%
6450 {\renewcommand*{\LWR@imageextension}{.SVG}}{}%
6451 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svgz}%
6452 {\renewcommand*{\LWR@imageextension}{.svgz}}{}%
6453 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVGZ}%
6454 {\renewcommand*{\LWR@imageextension}{.SVGZ}}{}%
```

Convert a PDF extension to SVG, leave the result in `\LWR@strresult`:

Must also `\detokenize` `.pdf` and `.svg` comparison strings.

```
6455 \StrSubstitute{\detokenize\expandafter{\LWR@expgraphicsfilename}}%
6456 {\detokenize{.pdf}}{\detokenize{.svg}}[\LWR@strresult]%
6457 %
6458 \StrSubstitute{\LWR@strresult}%
```

```
6459 {\detokenize{.PDF}}{\detokenize{.SVG}}[\LWR@strresult]%
```

For correct em sizing during the width and height conversions:

```
6460 \large%
```

Reset some defaults, possibly will be changed below if options were given:

```
6461 \setlength{\LWR@igwidth}{0pt}%
6462 \setlength{\LWR@igheight}{0pt}%
6463 \renewcommand*\LWR@igwidthstyle{}%
6464 \renewcommand*\LWR@igheightstyle{}%
6465 \renewcommand*\LWR@igorigin{}%
6466 \renewcommand*\LWR@igangle{}%
6467 \renewcommand*\LWR@igxscale{1}%
6468 \renewcommand*\LWR@igyscale{1}%
6469 \renewcommand*\LWR@igclass{inlineimage}%
```

If #3 is empty, only one optional argument was given, thus `graphicx` syntax:

```
6470 \IfValueTF{#3}{-}{%
6471 \IfValueTF{#2}%
6472 {\setkeys{igraph}{#2}}%
6473 {\setkeys{igraph}{}}%
6474 }%
```

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.

```
6475 \href{\thisgraphicspath\LWR@strresult\LWR@imageextension}%
6476 {% start of href
6477 \LWR@htmltag{% start of image tags
6478 img src="\thisgraphicspath\LWR@strresult\LWR@imageextension" \LWR@orignewline
6479 \LWR@origtilde{} alt="\LWR@strresult" \LWR@orignewline
```

Only include a style tag if a width, height, angle, or scale was given:

```
6480 \ifthenelse{
6481 \NOT\equal{\LWR@igwidthstyle}{} \OR
6482 \NOT\equal{\LWR@igheightstyle}{} \OR
6483 \NOT\equal{\LWR@igorigin}{} \OR
6484 \NOT\equal{\LWR@igangle}{} \OR
6485 \NOT\equal{\LWR@igxscale}{1} \OR
6486 \NOT\equal{\LWR@igyscale}{1}
6487 }%
6488 {\LWR@origtilde{} style="%
```

```

6489 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{}}%
6490 {\LWR@igwidthstyle;}{}%
6491 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{}}%
6492 {\LWR@igheightstyle;}{}%
6493 \ifthenelse{\NOT\equal{\LWR@igorigin}{}}%
6494 {\LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{}%
6495 \ifthenelse{\NOT\equal{\LWR@igangle}{}}%
6496 {%
6497 \LWR@rotstyle{-ms-}{\LWR@igangle}%
6498 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
6499 \LWR@rotstyle}{\LWR@igangle}%
6500 }{}%
6501 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%
6502 \NOT\equal{\LWR@igyscale}{1}}%
6503 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}%
6504 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
6505 \LWR@scalestyle}{\LWR@igxscale}{\LWR@igyscale}}{}%
6506 " \LWR@orignewline}{}%

```

Set the class:

```

6507 \LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%
6508 }% end of image tags
6509 }% end of href
6510 \endgroup

```

Return to small-sized output:

```

6511 \LWR@origscriptsize
6512 }

```

`\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.

```

6513 \renewcommand*{\includegraphics}
6514 {%

```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```

6515 \LWR@ensuredoingapar%
6516 \begingroup%
6517 \LWR@includegraphicsb%
6518 }

```

```
6519 \end{warpHTML}
```

for **PRINT output**: For print output, accept and then discard the new `class` key:

```
6520 \begin{warpprint}
6521 \define@key{Gin}{class}{}
6522 \end{warpprint}
```

63.4 `\rotatebox`, `\scalebox`, `\reflectbox`

for **HTML output**: 6523 `\begin{warpHTML}`

`\LWR@rotboxorigin` Holds the origin key letters.

```
6524 \newcommand*\LWR@rotboxorigin{}
```

`\LWR@originname` $\{ \langle letter \rangle \}$

Given one L^AT_EX origin key value, translate into an HTML origin word:

```
6525 \newcommand*\LWR@originname[1]{%
6526 \ifthenelse{\equal{#1}{t}}{top}{}%
6527 \ifthenelse{\equal{#1}{b}}{bottom}{}%
6528 \ifthenelse{\equal{#1}{c}}{center}{}%
6529 \ifthenelse{\equal{#1}{l}}{left}{}%
6530 \ifthenelse{\equal{#1}{r}}{right}{}%
6531 }
```

`\LWR@originnames` $\{ \langle letters \rangle \}$

Given one- or two-letter L^AT_EX origin key values, translate into HTML origin words:

```
6532 \newcommand*\LWR@originnames[1]{%
6533 \StrChar{#1}{1}[\LWR@strresult]%
6534 \LWR@originname{\LWR@strresult}
6535 \StrChar{#1}{2}[\LWR@strresult]%
6536 \LWR@originname{\LWR@strresult}
6537 }
```

Handle the origin key for `\rotatebox`:

```
6538 \define@key{krotbox}{origin}{%
6539 \renewcommand*\LWR@rotboxorigin{#1}%
6540 }
```

These keys are ignored:

```
6541 \define@key{krotbox}{x}{}
6542 \define@key{krotbox}{y}{}
6543 \define@key{krotbox}{units}{}

```

`\rotatebox` [*keyval list*] *{angle}* *{text}*

Will `\let\rotatebox\LWR@rotatebox` at `\LWR@LwarpStart`, in case `\rotatebox` was over-written by a later package load.

```
6544 \LetLtxMacro\LWR@origrotatebox\rotatebox
6545
6546 \NewDocumentCommand{\LWR@rotatebox}{O{} m +m}{%

```

Reset the origin to “none-given”:

```
6547 \renewcommand*\LWR@rotboxorigin{}

```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
6548 \setkeys{krotbox}{#1}%

```

Select `inline-block` so that HTML will transform this span:

```
6549 \LWR@htmltagc{span style="display: inline-block; %

```

If an origin was given, translate and print the origin information:

```
6550 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}%
6551 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%

```

Print the rotation information:

```
6552 \LWR@rotstyle{-ms-}{#2} %
6553 \LWR@rotstyle{-webkit-}{#2} %
6554 \LWR@rotstyle{}{#2} %
6555 "{}\LWR@orignewline%

```

Print the text to be rotated:

```
6556 \begin{LWR@nestspan}%
6557 #3%

```

Close the span:

```
6558 \LWR@htmltagc{/span}%
6559 \end{LWR@nestspan}%
6560 }

```

`\scalebox` $\langle h-scale \rangle$ $[\langle v-scale \rangle]$ $\langle text \rangle$

Will `\let\scalebox\LWR@scalebox` at `\LWR@LwarpStart`, in case `\scalebox` was over-written by a later package load.

```
6561 \LetLtxMacro\LWR@origscalebox\scalebox
6562
6563 \NewDocumentCommand{\LWR@scalebox}{m o m}{%
```

Select `inline-block` so that HTML will transform this span:

```
6564 \LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
6565 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6566 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6567 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6568 "}{}}%
```

Print the text to be scaled:

```
6569 \begin{LWR@nestspan}%
6570 #3%
```

Close the span:

```
6571 \LWR@htmltagc{/span}%
6572 \end{LWR@nestspan}%
6573 }
```

`\reflectbox` $\langle text \rangle$

Will `\let\reflectbox\LWR@reflectbox` at `\LWR@LwarpStart`, in case `\reflectbox` was over-written by a later package load.

```
6574 \let\LWR@origreflectbox\reflectbox
6575
6576 \newcommand{\LWR@reflectbox}[1]{\LWR@scalebox{-1}[1]{#1}}
6577 \end{warpHTML}
```

63.5 Null functions

These functions are not supported by lwarp's HTML conversion.

for HTML output: `6578 \begin{warpHTML}`

```
\resizebox {⟨h-length⟩} {⟨v-length⟩} {⟨text⟩}
```

Simply prints its text argument.

```
6579 \renewcommand{\resizebox}[3]{#3}
```

```
6580 \end{warpHTML}
```

64 Cleveref

Pkg `cleveref` `cleveref` package is used as-is with minor patches.

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 9 on 272 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: 6581 `\begin{warpHTML}`

`\AtEndPreamble` forces `cleveref` to be loaded last:

```
6582 \AtEndPreamble{
6583 \RequirePackage{cleveref}
6584 }
```

The following patches are applied after `cleveref` has loaded, and after `\AtBeginDocument`:

```
6585 \AfterEndPreamble{
6586 \LWR@traceinfo{Patching cleveref.}
```

```
\@@setcref {⟨kindofref⟩} {⟨label⟩}
```

```
6587 \renewcommand*\@@setcref[2]{#1{\ref{#2}}{-}}
```

```

\@@setcrefrange  {\langle text \rangle} {\langle label \rangle} {\langle label \rangle}

6588 \renewcommand{\@@setcrefrange}[3]{%
6589 #1{\ref{#2}}{\ref{#3}}{}{}{}}

\cpagerefFor  Redefinable word between “page(s)” and the page numbers.

6590 \newcommand*\cpagerefFor{for}

\@@setcpageref  {\langle typeofref \rangle} {\langle label \rangle}, where typeofref is “page” or “pages”

6591 \renewcommand*\@@setcpageref[2]{%
6592 #1{\cpagerefFor\ \cref{#2}}{}{}}%
6593 }

6594 \renewcommand{\@@setcpagerefrange}[3]{%
6595 #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}{}{}}
6596 }% AfterEndPreamble

```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

```

6597 \let\LWR@origlabel\label
6598 \let\label\LWR@newlabel
6599 \let\LWR@origref\ref
6600 \let\ref\LWR@newref% \end{ syntax highlighting
6601 \let\LWR@origpageref\pageref
6602 \let\pageref\LWR@newpageref
6603
6604
6605
6606 \end{warpHTML}

```

65 Picture

Env `picture` The `picture` environment is enclosed inside a `\lateximage`.

for HTML output: `6607 \begin{warpHTML}`

Env `picture`

```

6608 \BeforeBeginEnvironment{picture}{\lateximage}
6609
6610 \AfterEndEnvironment{picture}{\endlateximage}

```

6611 `\end{warpHTML}`

66 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 L^AT_EX PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

for HTML output: 6612 `\begin{warpHTML}`

66.1 Counters and lengths

Ctrl LWR@minipagedepth Used to only reset the line width at the outermost minipage.

6613 `\newcounter{LWR@minipagedepth}`
 6614 `\setcounter{LWR@minipagedepth}{0}`

Len `\WR@minipagewidth` Used to convert the width into printable units.

```
6615 \newlength{\LWR@minipagewidth}
```

Len `\WR@minipageheight` Used to convert the height into printable units.

```
6616 \newlength{\LWR@minipageheight}
```

66.2 Footnote handling

Also see section 41 for other forms of footnotes.

66.3 Minipage handling

`\LWR@endminipage` Used to close a minipage.

Copied the L^AT_EX definition and modified to create a `mpfootnotes` div class:

```
6617 \def\LWR@endminipage{%
6618   \par
6619   \unskip
6620   \ifvoid\@mpfootins\else
6621     \vskip\skip\@mpfootins
6622     \normalcolor
6623 \LWR@htmldivclass{mpfootnotes}
6624 \LWR@origmedskip
6625   \unvbox\@mpfootins
6626 \LWR@htmldivclassend{mpfootnotes}
6627   \fi
6628   \@minipagefalse
6629 \color@endgroup
6630 \egroup
6631 \expandafter\@iiiparbox\@mpargs{\unvbox\@tempboxa}}
```

`\LWR@subminipage` Used to create a PDF minipage without creating an HTML minipage. This allows footnotes to appear at the bottom of the minipage instead of the bottom of the HTML page.

```
6632 \newcommand*\LWR@subminipage}{%
6633 \LWR@stoppars
6634 \LWR@origminipage{6in}
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
6635 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
6636 \LWR@startpars%
6637 }
```

`\LWR@endsubminipage` Closes the subminipage.

```
6638 \newcommand*{\LWR@endsubminipage}{%
6639 \LWR@stoppars%
6640 \LWR@endminipage% The following empty line is required:
6641
6642 }
```

Bool `LWR@minipagefullwidth` Should the next minipage have no HTML width?

```
6643 \newbool{LWR@minipagefullwidth}
6644 \boolfalse{LWR@minipagefullwidth}
```

`\minipagefullwidth` Requests that the next minipage have no width tag in HTML:

for HTML output: `6645 \newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}`
`6646 \end{warpHTML}`

for PRINT output: `6647 \begin{warpprint}`
`6648 \newcommand*{\minipagefullwidth}{}`
`6649 \end{warpprint}`

for HTML output: `6650 \begin{warpHTML}`

Bool `LWR@minipagethispar` Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
6651 \newbool{LWR@minipagethispar}
6652 \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.

```
6653 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}
6654 {%
6655 \uselengthunit{in}%
```

```

6656 \setlength{\LWR@minipagewidth}{#4}%
6657 \ifthenelse{\cinttest{\value{LWR@minipagedepth}}{=} {0}}{%
6658 \addtolength{\LWR@minipagewidth}{3em}% room for frames
6659 \setlength{\linewidth}{6in}%
6660 \setlength{\textwidth}{6in}%
6661 \setlength{\textheight}{9in}%
6662 }{}%
6663 \LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}
6664 \addtocounter{LWR@minipagedepth}{1}%
6665 \setlength{\LWR@minipageheight}{\textheight}% default unless specified
6666 \IfValueTF{#2}{\setlength{\LWR@minipageheight}{#2}}{}%

```

\LaTeX 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.

```
6667 \LWR@stoppars%
```

Create the `<div>` tag with optional alignment style:

```

6668 \LWR@traceinfo{minipage: creating div class}%
6669 \LWR@orignewpage%
6670 \LWR@htmltag{div class="minipage" style="%
6671 \ifthenelse{\equal{#1}{t}}{vertical-align: bottom ; }{}%
6672 \ifthenelse{\equal{#1}{c}}{vertical-align: middle ; }{}%
6673 \ifthenelse{\equal{#1}{b}}{vertical-align: top ; }{}%
6674 \ifthenelse{\equal{#3}{t}}{justify-content: flex-start ; }{}%
6675 \ifthenelse{\equal{#3}{c}}{justify-content: center ; }{}%
6676 \ifthenelse{\equal{#3}{b}}{justify-content: flex-end ; }{}%
6677 \ifthenelse{\equal{#3}{s}}{justify-content: space-between ; }{}%

```

Print the width and optional height styles:

```

6678 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
6679 \uselengthunit{PT}%
6680 \ifbool{LWR@minipagefullwidth}%
6681 {\boolfalse{LWR@minipagefullwidth}}%
6682 {%
6683 \ifthenelse{\lengthtest{#4}=\linewidth}%
6684 {%
6685 {width:\rndprintlength{\LWR@minipagewidth} ; }%
6686 }%
6687 \LWR@traceinfo{minipage: about to print the height}%
6688 \IfValueTF{#2}{height:\rndprintlength{\LWR@minipageheight} ; }{}%
6689 "%}

```

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.

```
6690
6691 \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.

```
6692 \setlength{\linewidth}{#4}% the original width
6693 \setlength{\textwidth}{6in}%
6694 \setlength{\textheight}{9in}%
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
6695 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
6696 \LWR@startpars%
6697 \LWR@traceinfo{minipage: finished starting the minipage}%
6698 }
```

End the environment with \LaTeX processing and closing tag:

```
6699 {%
6700 \LWR@stoppars%
6701 \LWR@endminipage% The following empty line is required:
6702
6703 \LWR@htmldivclassend{minipage}%
6704 \LWR@origvspace{1\baselineskip}% required for subcaption
6705 \addtocounter{LWR@minipagedepth}{-1}%
6706 \LWR@startpars%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
6707 \global\booltrue{LWR@minipagethispar}%
6708 }
```

66.4 Parbox, makebox, framebox, fbox, raisebox

```
\parbox [pos] [height] [inner-pos] {width} {text}
```

A parbox uses the minipage code:

```
6709 \RenewDocumentCommand{\parbox}{0{t} o 0{t} m +m}
```

```

6710 {
6711 \LWR@traceinfo{parbox of width #4}%
6712 \begin{minipage}[#1][#2][#3]{#4}
6713 #5
6714 \end{minipage}
6715 }

```

`\makebox` ($\langle() \rangle$ posn) [$\langle width \rangle$] [$\langle pos \rangle$] $\{ \langle text \rangle \}$

Width and positions are ignored.

```

6716 \LetLtxMacro\LWR@origmakebox\makebox
6717
6718 \RenewDocumentCommand{\makebox}{d() o o m}{%
6719 \mbox{#4}
6720 }

```

`\framebox` [$\langle width \rangle$] [$\langle pos \rangle$] $\{ \langle text \rangle \}$

Width and position are ignored.

```

6721 \LetLtxMacro\LWR@origframebox\framebox
6722
6723 \RenewDocumentCommand{\framebox}{o o m}{%
6724 \fbox{#3}
6725 }

```

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

Creates a framed inline span enclosing the text.

```

6726 \let\LWR@origfbox\fbox
6727
6728 \renewcommand{\fbox}[1]{%
6729 \LWR@htmltagc{span class="framebox"}%
6730 \begin{LWR@nestspan}%
6731 #1
6732 \LWR@htmltagc{/span}%
6733 \end{LWR@nestspan}%
6734 }

```

`\fboxBlock` $\{ \langle text \rangle \}$ Creates a framed HTML `<div>` of the text.

A print-output version is also supplied below.

```

6735 \newcommand{\fboxBlock}[1]{%
6736 \begin{BlockClass}{framed}

```

```
6737 #1
6738 \end{BlockClass}
6739 }
```

Env `fminipage` [*align*] [*height*] [*align*] {*width*}

Creates a framed HTML `<div>` around its contents.

A print-output version is also supplied below.

```
6740 \NewDocumentEnvironment{fminipage}{0{t} o 0{t} m}
6741 {%
6742 \setlength{\LWR@tempwidth}{#4}
6743 \begin{BlockClass}[width:\rndprintlength{\LWR@tempwidth};]{framed}%
6744 }
6745 {\end{BlockClass}}
```

`\raisebox` {*raiselen*} [*height*] [*depth*] {*text*}

```
6746 \LetLtxMacro{\LWR@origraisebox}{\raisebox}
6747
6748 \RenewDocumentCommand{\raisebox}{m o o m}{%
6749 #4%
6750 }
```

```
6751 \end{warpHTML}
```

for HTML & PRINT: 6752 `\begin{warppall}`

Env `LWR@fminipageprint` [*align*] [*height*] [*align*] {*width*}

Creates a frame around its contents.

```
6753 \newsavebox{\LWR@fminipagebox}
6754
6755 \NewDocumentEnvironment{LWR@fminipageprint}{0{t} o 0{t} m}
6756 {%
6757 \IfValueTF{#2}%
6758 {\minipage[#1][#2][#3]{#4}}%
6759 {\minipage[#3]{#4}}%
6760 \begin{lrbox}{\LWR@fminipagebox}%
6761 \IfValueTF{#2}%
6762 {\minipage[#1][#2][#3]{\linewidth-2\fbboxsep-2\fbboxrule}}%
6763 {\minipage[#3]{\linewidth-2\fbboxsep-2\fbboxrule}}%
6764 }
6765 {%
6766 \endminipage%
```

```

6767 \end{lrbox}%
6768 \fbox{\usebox{\LWR@fminipagebox}}%
6769 \endminipage%
6770 }

6771 \end{warpall}

```

for PRINT output: 6772 \begin{warpprint}

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

Creates a framed HTML `<div>` around the text.

```
6773 \let\fboxBlock\fbox
```

Env `fminipage` $[\langle align \rangle] [\langle height \rangle] [\langle align \rangle] \{ \langle width \rangle \}$

Creates a frame around its contents.

```

6774 \LetLtxMacro{\fminipage}{\LWR@fminipageprint}
6775 \LetLtxMacro{\endfminipage}{\endLWR@fminipageprint}

6776 \end{warpprint}

```

67 Direct formatting

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

For high-level block and inline custom CSS classes, see section [35.7](#).

for HTML output: 6777 \begin{warpHTML}

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

```

6778 \DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{#1}}
6779 \DeclareRobustCommand{\LWR@nullemph}[1]{#1}
6780 \LetLtxMacro{\emph}{\LWR@HTMLemph}

```

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

```

6781 \DeclareRobustCommand{\LWR@HTMLtextmd}[1]{\LWR@htmlspan{textmd}{#1}}
6782 \DeclareRobustCommand{\LWR@nulltextmd}[1]{#1}
6783 \LetLtxMacro{\textmd}{\LWR@HTMLtextmd}

```

`\textbf` $\langle text \rangle$

```
6784 \DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{#1}}
6785 \DeclareRobustCommand{\LWR@nulltextbf}[1]{#1}
6786 \LetLtxMacro{\textbf}{\LWR@HTMLtextbf}
```

`\textrm` $\langle text \rangle$

```
6787 \DeclareRobustCommand{\LWR@HTMLtextrm}[1]{\LWR@htmlspan{textrm}{#1}}
6788 \DeclareRobustCommand{\LWR@nulltextrm}[1]{#1}
6789 \LetLtxMacro{\textrm}{\LWR@HTMLtextrm}
```

`\textsf` $\langle text \rangle$

```
6790 \DeclareRobustCommand{\LWR@HTMLtextsf}[1]{\LWR@htmlspan{textsf}{#1}}
6791 \DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
6792 \LetLtxMacro{\textsf}{\LWR@HTMLtextsf}
```

`\texttt` $\langle text \rangle$

```
6793 \DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
6794 \DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
6795 \LetLtxMacro{\texttt}{\LWR@HTMLtexttt}
```

`\textup` $\langle text \rangle$

```
6796 \DeclareRobustCommand{\LWR@HTMLtextup}[1]{\LWR@htmlspan{textup}{#1}}
6797 \DeclareRobustCommand{\LWR@nulltextup}[1]{#1}
6798 \LetLtxMacro{\textup}{\LWR@HTMLtextup}
```

`\textit` $\langle text \rangle$

```
6799 \DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}}
6800 \DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
6801 \LetLtxMacro{\textit}{\LWR@HTMLtextit}
```

`\textsc` $\langle text \rangle$

```
6802 \DeclareRobustCommand{\LWR@HTMLtextsc}[1]{\LWR@htmlspan{textsc}{#1}}
6803 \DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}
6804 \LetLtxMacro{\textsc}{\LWR@HTMLtextsc}
```

`\textsl` $\langle text \rangle$

```
6805 \DeclareRobustCommand{\LWR@HTMLtextsl}[1]{\LWR@htmlspan{testsl}{#1}}
```

```
6806 \DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}
6807 \LetLtxMacro{\textsl}{\LWR@HTMLtextsl}
```

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

```
6808 \DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
6809 \DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
6810 \LetLtxMacro{\textnormal}{\LWR@HTMLtextnormal}
```

```
6811 \DeclareRobustCommand{\LWR@HTMLrmfamily}[1]{}
6812 \DeclareRobustCommand{\LWR@HTMLsffamily}[1]{}
6813 \DeclareRobustCommand{\LWR@HTMLttfamily}[1]{}
6814 \DeclareRobustCommand{\LWR@HTMLbfseries}[1]{}
6815 \DeclareRobustCommand{\LWR@HTMLmdseries}[1]{}
6816 \DeclareRobustCommand{\LWR@HTMLupshape}[1]{}
6817 \DeclareRobustCommand{\LWR@HTMLslshape}[1]{}
6818 \DeclareRobustCommand{\LWR@HTMLscshape}[1]{}
6819 \DeclareRobustCommand{\LWR@HTMLitshape}[1]{}
6820 \DeclareRobustCommand{\LWR@HTMLem}[1]{}
6821 \DeclareRobustCommand{\LWR@nullrmfamily}[1]{}
6822 \DeclareRobustCommand{\LWR@nullsffamily}[1]{}
6823 \DeclareRobustCommand{\LWR@nullttfamily}[1]{}
6824 \DeclareRobustCommand{\LWR@nullbfseries}[1]{}
6825 \DeclareRobustCommand{\LWR@nullmdseries}[1]{}
6826 \DeclareRobustCommand{\LWR@nullupshape}[1]{}
6827 \DeclareRobustCommand{\LWR@nullslshape}[1]{}
6828 \DeclareRobustCommand{\LWR@nullscshape}[1]{}
6829 \DeclareRobustCommand{\LWR@nullitshape}[1]{}
6830 \DeclareRobustCommand{\LWR@nullem}[1]{}

```

`\LWR@nullfonts` Removes formatting during filename operations.

```
6831 \newcommand*{\LWR@nullfonts}{%
6832 \LetLtxMacro{\emph}{\LWR@nullemph}%
6833 \LetLtxMacro{\textmd}{\LWR@nulltextmd}%
6834 \LetLtxMacro{\textbf}{\LWR@nulltextbf}%
6835 \LetLtxMacro{\textrm}{\LWR@nulltextrm}%
6836 \LetLtxMacro{\textsf}{\LWR@nulltextsf}%
6837 \LetLtxMacro{\texttt}{\LWR@nulltexttt}%
6838 \LetLtxMacro{\textup}{\LWR@nulltextup}%
6839 \LetLtxMacro{\textit}{\LWR@nulltextit}%
6840 \LetLtxMacro{\textsc}{\LWR@nulltextsc}%
6841 \LetLtxMacro{\textsl}{\LWR@nulltextsl}%
6842 \LetLtxMacro{\textnormal}{\LWR@nulltextnormal}%
6843 \LetLtxMacro{\rmfamily}{\LWR@nullrmfamily}%
6844 \LetLtxMacro{\sffamily}{\LWR@nullsffamily}%
6845 \LetLtxMacro{\ttfamily}{\LWR@nullttfamily}%
6846 \LetLtxMacro{\bfseries}{\LWR@nullbfseries}%

```

```

6847 \LetLtxMacro{\mdseries}{\LWR@nullmdseries}%
6848 \LetLtxMacro{\upshape}{\LWR@nullupshape}%
6849 \LetLtxMacro{\slshape}{\LWR@nullslshape}%
6850 \LetLtxMacro{\scshape}{\LWR@nullscshape}%
6851 \LetLtxMacro{\itshape}{\LWR@nullitshape}%
6852 \LetLtxMacro{\em}{\LWR@nullem}%
6853 \LetLtxMacro{\normalfont}{\LWR@nullnormalfont}%
6854 \renewcommand*{\HTMLunicode}[1]{}%
6855 % \renewcommand*{\HTMLunicode}[1]{%
6856 % \renewcommand*{\HTMLentity}[1]{##1}
6857 \renewcommand*{\HTMLentity}[1]{}%

```

Ampersand becomes “and”, which is a short word and is then removed from the filename.

```

6858 \renewcommand*{\&}{and}%
6859 \renewcommand{\textsuperscript}[1]{##1}%
6860 % \renewcommand{\textsuperscript}[1]{##1}
6861 \renewcommand{\textsubscript}[1]{##1}%
6862 % \renewcommand{\textsubscript}[1]{##1}
6863 \RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
6864 \DeclareExpandableDocumentCommand{\InlineClass}{o m +m}{##3}%
6865 % \DeclareExpandableDocumentCommand{\InlineClass}{o m +m}{##3}%
6866 }

```

`\mdseries`

```
6867 \renewcommand*{\mdseries}{}
```

`\bfseries`

```
6868 \renewcommand*{\bfseries}{}
```

`\rmfamily`

```
6869 \renewcommand*{\rmfamily}{}
```

`\sffamily`

```
6870 \renewcommand*{\sffamily}{}
```

`\ttfamily`

```
6871 \renewcommand*{\ttfamily}{}
```

`\upshape`

```
6872 \renewcommand*\upshape{}
```

`\itshape`

```
6873 \renewcommand*\itshape{}
```

`\scshape`

```
6874 \renewcommand*\scshape{}
```

`\scshape`

```
6875 \renewcommand*\normalfont{}
```

`\sp` $\langle text \rangle$

For siunitx. Must work in math mode.

```
6876 \renewcommand{\sp}[1]{\text{<sup>#1</sup>}}
```

`\sb` $\langle text \rangle$

For siunitx. Must work in math mode.

```
6877 \renewcommand{\sb}[1]{\text{<sub>#1</sub>}}
```

`\textsuperscript` $\langle text \rangle$

```
6878 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

`\textsubscript` $\langle text \rangle$

```
6879 \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.

```
6880 \AtBeginDocument{\let\up\textsuperscript}
```

`\fup` $\langle text \rangle$ Prints superscript.

Supports `fntcount` package.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
6881 \AtBeginDocument{\let\fup\textsuperscript}
```

`\hfill`

```
6882 \renewcommand*{\hfill}{\quad}
```

`\hrulefill`

```
6883 \renewcommand*{\hrulefill}{\rule{1in}{1pt}}
```

`\dotfill`

```
6884 \renewcommand*{\dotfill}{\dots}
```

```
6885 \end{warpHTML}
```

68 Skips, spaces, font sizes

for HTML output: `6886 \begin{warpHTML}`

`\`, must be redefined after `\RequirePackage{printlen}`

```
6887 \let\LWR@origcomma\,
6888 \let\LWR@origtilde~
6889 \let\LWR@origenskip\enskip
6890 \let\LWR@origquad\quad
6891 \let\LWR@origqquad\qquad
6892 \let\LWR@orighspace\hspace
6893 \let\LWR@origvspace\vspace
6894 \let\LWR@origrule\rule
6895 \let\LWR@origmedskip\medskip
```

Direct-formatting space commands become HTML entities:

```
6896 \renewcommand*{\,}{\HTMLUnicode{202f}} % HTML thin non-breakable space
6897 \renewcommand*{~}{\HTMLentity{nbsp}}
6898 \renewcommand*{\textellipsis}{\HTMLUnicode{2026}}
```

Direct-formatting font sizes are ignored:

```

6899 \let\LWR@orignormalsize\normalsize
6900 \let\LWR@origsmall\small
6901 \let\LWR@origfootnotesize\footnotesize
6902 \let\LWR@origscriptsize\scriptsize
6903 \let\LWR@origtiny\tiny
6904 \let\LWR@origlarge\large
6905 \let\LWR@origLarge\Large
6906 \let\LWR@origLARGE\LARGE
6907 \let\LWR@orighuge\huge
6908 \let\LWR@origHuge\Huge
6909 \renewcommand*\normalsize{}
6910 \renewcommand*\small{}
6911 \renewcommand*\footnotesize{}
6912 \renewcommand*\scriptsize{}
6913 \renewcommand*\tiny{}
6914 \renewcommand*\large{}
6915 \renewcommand*\Large{}
6916 \renewcommand*\LARGE{}
6917 \renewcommand*\huge{}
6918 \renewcommand*\Huge{}
6919
6920 \renewcommand*\onecolumn{}
6921
6922 \renewcommand{\twocolumn}[1] [] {}
6923
6924 #1
6925
6926 }

```

`\newline` Uses the HTML `
` element.

```

6927 \newcommand*\LWR@newlinebr{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
6928 \let\newline\LWR@newlinebr

```

`\\` Redefined to `\LWR@endoffline` or `\LWR@tabularendoffline`.

`\LWR@endoffline` * [*len*]

`\\` is assigned to `\LWR@endoffline` at `\LWR@LwarpStart`.

Inside `tabular`, `\\` is temporarily changed to `\LWR@tabularendoffline`.

```

6929 \LetLtxMacro\LWR@origendoffline\\
6930 \NewDocumentCommand{\LWR@endoffline}{s o}
6931 {%
6932 \newline%

```

6933 }

`\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 `\qqquad` until the end of the paragraph, when the closing `p` tag is created.

`\hspace`
`\enskip`
`\quad`
`\qqquad`

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 `\qqquad`'s HTML output.

```
6934 \newcommand*{\LWR@minipagestartpars}{%
6935 \ifbool{LWR@minipagethispar}%
6936 {%
6937 \LWR@startpars%
6938 }{}}%
6939 }
```

`\LWR@minipagestoppars` Placed just after `\hspace`, `\quad`, or `\qqquad`'s HTML output.

```
6940 \newcommand*{\LWR@minipagestoppars}{%
6941 \ifbool{LWR@minipagethispar}%
6942 {%
6943 \LWR@stoppars%
6944 }{}}%
6945 }
```

`\quad` Handles special minipage & horizontal space interactions.

```
6946 \renewcommand*{\quad}{%
6947 \LWR@minipagestoppars%
6948 \HTMLUnicode{2001}%
6949 \LWR@minipagestartpars%
6950 }
```

`\qqquad` Handles special minipage & horizontal space interactions.

```
6951 \renewcommand*{\qqquad}{\quad\quad}
```

`\enskip` Handles special minipage & horizontal space interactions.

```
6952 \renewcommand*{\enskip}{%
6953 \LWR@minipagestoppars%
6954 \HTMLUnicode{2000}%
6955 \LWR@minipagestartpars%
6956 }
```

Len `\WR@tempwidth` Used to compute span width, height, raise for `\hspace` and `\rule`:

```
Len \WR@tempheight 6957 \newlength{\LWR@tempwidth}
Len \WR@tempraise 6958 \newlength{\LWR@tempheight}
6959 \newlength{\LWR@tempraise}
```

`\LWR@hspace` * $\langle length \rangle$

Handles special minipage & horizontal space interactions.

Prints a span of a given width. Ignores the optional star.

`\hspace{\fill}` is converted to `\hspace{2em}`, equal to `\qqquad`.

```
6960 \NewDocumentCommand{\LWR@hspace}{s m}{%
6961 \setlength{\LWR@tempwidth}{#2}%
```

If `\fill`, change to `\qqquad`:

```
6962 \ifnum\gluestretchorder\LWR@tempwidth>0%
6963 \setlength{\LWR@tempwidth}{2em}%
6964 \fi%
```

Only if the width is not zero:

```
6965 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{0pt}}{ }{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
6966 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
6967 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{.16667em}}
6968 {%
6969 \HTMLUnicode{2009}% thin breakable space
6970 }%
```

Print the span with the converted width. Not rounded.

```
6971 {%
6972 \uselengthunit{PT}%
6973 \LWR@htmltagc{%
6974 span style="width:\printlength{\LWR@tempwidth}; display:inline-block;"%
6975 }%
6976 \LWR@htmltagc{/span}%
6977 }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
6978 \LWR@minipagestartpars%
6979 }%
6980 }
```

`\LWR@nohspace` * $\langle length \rangle$

Used to disable `\hspace` while creating description `\items`.

```
6981 \NewDocumentCommand{\LWR@nohspace}{s m}{}
```

`\hspace` * $\langle length \rangle$

Handles special minipage & horizontal space interactions.

```
6982 \LetLtxMacro{\hspace}{\LWR@hspace}
```

`\LWR@vspace` * $\langle length \rangle$ Nullified `vspace`.

```
6983 \NewDocumentCommand{\LWR@vspace}{s m}{}
```

`\vspace` * $\langle length \rangle$ Nullified.

```
6984 \let\vspace\LWR@vspace
```

`\linebreak` [$\langle num \rangle$] Inserts an HTML `br` tag.

```
6985 \renewcommand*{\linebreak}[1] [] {\newline}
```

`\nolinebreak` [$\langle num \rangle$]

```
6986 \renewcommand*{\nolinebreak}[1] [] {}
```

`\pagebreak` [*num*] Starts a new paragraph.

```
6987 \renewcommand*\pagebreak[1] [] {
6988
6989 }
```

`\nopagebreak` [*num*]

```
6990 \renewcommand*\nopagebreak[1] [] {}
```

`\enlargethispage` * {*len*}

```
6991 \RenewDocumentCommand{\enlargethispage}{s m}{}
```

`\LWR@rule` [*raise*] {*width*} {*height*}

Handles special minipage & horizontal space interactions.

Creates a span of a given width and height. Ignores the optional star.

`\fill` is zero-width, so `\hspace{\fill}` is ignored.

```
6992 \NewDocumentCommand{\LWR@rule}{o m m}{%
```

The width is copied into a temporary L^AT_EX length, from which comparisons and conversions may be made:

```
6993 \setlength{\LWR@tempwidth}{#2}%
```

If it's zero-width then skip the entire rule:

```
6994 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}
6995 {}% zero- width
6996 {% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
6997 \ifthenelse{\lengthtest{\LWR@tempwidth>0pt}\AND%
6998 \lengthtest{\LWR@tempwidth<1pt}}%
6999 {\setlength{\LWR@tempwidth}{1pt}}{}%
```

Likewise with height:

```
7000 \setlength{\LWR@tempheight}{#3}%
7001 \ifthenelse{\lengthtest{\LWR@tempheight>0pt}\AND%
7002 \lengthtest{\LWR@tempheight<1pt}}%
7003 {\setlength{\LWR@tempheight}{1pt}}{}%
```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
7004 \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 L^AT_EX code.

```
7005 \uselengthunit{PT}%
7006 \LWR@htmltagc{%
7007 span
7008 style=" %
```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```
7009 background:\LWR@currenttextcolor; %
```

The width and height are printed, converted to PT:

```
7010 width:\printlength{\LWR@tempwidth}; %
7011 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 L^AT_EX length allows a typical L^AT_EX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a L^AT_EX length limits the allowable syntax. To do: A superior method would compute a ratio of L^AT_EX ex height, then print that to HTML with an ex unit.

```
7012 \IfValueTF{#1}%
7013 {%
7014 \setlength{\LWR@tempraise}{Opt-#1}%
7015 \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
7016 \LWR@orignewline%
7017 -ms-transform: translate(0pt,\printlength{\LWR@tempraise}); %
7018 \LWR@orignewline%
7019 -webkit-transform: translate(0pt,\printlength{\LWR@tempraise}); %
7020 \LWR@orignewline%
7021 transform: translate(0pt,\printlength{\LWR@tempraise}); %
7022 \LWR@orignewline%
7023 }{}%
```

Display inline-block to place the span inline with the text:

```
7024 display:inline-block;"%
7025 }%
7026 \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7027 \LWR@minipagestartpars%
7028 }% non-zero width
7029 }
```

`\rule` [*raise*] [*width*] [*height*]

Handles special minipage & horizontal space interactions.

```
7030 \renewcommand{\rule}{\LWR@rule}
7031 \end{warpHTML}
```

69 `\phantomsection`

for HTML output: 7032 `\begin{warpHTML}`

`\phantomsection` Emulate the hyperref `\phantomsection` command, often used to insert the bibliography into table of contents:

```
7033 \newcommand*\phantomsection{\section*{}}
7034 \end{warpHTML}
```

70 `\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>

70.1 HTML logos

for HTML output: 7035 `\begin{warpHTML}`

`\TeX` `\TeX`

`latexlogo` is a CSS class used to properly typeset the E and A in `\LaTeX` and friends.

`latexlogofont` is a CSS class used to select the font for the rest of the logo in `\LaTeX`, `\LuaTeX`, `\ConTeXt`, etc.

```
7036 \newcommand*\LWR@TeX}
7037 {\InlineClass{latexlogofont}%
7038 {\InlineClass{latexlogo}{T\textsubscript{e}X}}}
```

`\LaTeX` `\LaTeX`, `\LaTeX 2 ϵ`

`\LaTeXe`

```
7039 \newcommand*\LWR@LaTeX}
7040 {\InlineClass{latexlogofont}%
7041 {\InlineClass{latexlogo}%
7042 {L\textsuperscript{a}T\textsubscript{e}X}}
7043
7044 \renewcommand*\LaTeXe}
7045 {\LaTeX\InlineClass{latexlogofont}%
7046 {\,2\textsubscript{\textit{\HTMLUnicode{3B5}}}}}
```

`\LuaTeX` `\LuaTeX`, `\LuaLaTeX`

`\LuaLaTeX`

```
7047 \newcommand*\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
7048 \newcommand*\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

`\XeTeX` `\XeTeX`, `\XeLaTeX`

`\XeLaTeX`

`xetexlogo` is a CSS class which aligns the backwards E in `\XeTeX` and spaces `\TeX` appropriately.

`xelatexlogo` is a CSS class which aligns the backwards E in `\XeLaTeX` and spaces `\LaTeX` appropriately.

```
7049 \newcommand*\Xe}
7050 {X\textsubscript{\HTMLUnicode{18e}}}
7051 \newcommand*\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
7052 \newcommand*\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}
```

`\ConTeXt` `\ConTeXt`

```
7053 \newcommand*\LWR@ConTeXt}
7054 {\InlineClass{latexlogofont}{Con}\TeX{}}
7055 \InlineClass{latexlogofont}{t}}
```

`\BibTeX` `\BibTeX`, `MakeIndex`

`\MakeIndex`

```

7056 \providecommand*\BibTeX}
7057 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}
7058
7059 \newcommand*\MakeIndex}
7060 {\InlineClass{latexlogofont}{\textit{MakeIndex}}}}

```

`\AmS` \mathcal{AMS}

`amslogo` is a CSS class used for the \mathcal{AMS} logo.

```

7061 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}}
7062 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}

```

`\MiKTeX` $\text{MiK}\TeX$

```

7063 \newcommand*\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}

```

`\LyX` LyX

`lyxlogo` is a CSS class used for the LyX logo.

```

7064 \newcommand*\LyX}{\InlineClass{lyxlogo}{LyX}}

```

```

7065 \end{warpHTML}

```

70.2 Print logos

```

for PRINT output: 7066 \begin{warpprint}
7067 \newcommand*\XeTeXrevE}
7068   {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
7069 \providecommand*\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
7070 \providecommand*\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
7071 \providecommand*\AmS}{%
7072 \leavevmode\hbox{\mathcal A\kern-.2em\lower.376ex%
7073 \hbox{\mathcal M$}\kern-.2em\mathcal S$}}
7074 \newcommand*\LyX}{\textsf{LyX}}
7075 \providecommand*\LuaTeX}{\mbox{Lua\TeX}}
7076 \providecommand*\LuaLaTeX}{\mbox{Lua\LaTeX}}
7077 \providecommand*\BibTeX}{\mbox{B\textsc{ib}}\TeX}}
7078 \providecommand*\MakeIndex}{\mbox{\textit{MakeIndex}}}
7079 \providecommand*\ConTeXt}{\mbox{Con\TeX{t}}}
7080 \providecommand*\MiKTeX}{\mbox{MiK\TeX}}
7081 \end{warpprint}

```

71 `\AtBeginDocument`, `\AtEndDocument`

for HTML output: 7082 `\begin{warpHTML}`

`\LWR@LwarpStart` Automatically sets up the HTML-related actions for the start and end of the
`\LWR@LwarpEnd` document.

```
7083 \AfterEndPreamble{\LWR@LwarpStart}
```

```
7084 \AtEndDocument{\LWR@LwarpEnd}
```

```
7085 \end{warpHTML}
```

72 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.

Package 2

lwarp-abstract.sty

73 Abstract

(Based on original code by PETER WILSON.)

`Pkg abstract` abstract is supported and patched by lwarp.

abstract is supported. 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{\abstitlekip}%
35     \fi
36     \fi
37     \vspace{\abstitlekip}%
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{\abstitlekip}%
57     \fi
58     \fi
59     \vspace{\abstitlekip}%
60     \fi
61     \put@bsintoc%
62 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
63 {\par\end{@bstr@ctlist}}
64 \fi
65

```

Package 3

lwarp-afterpage.sty

74 Afterpage

Pkg afterpage Emulated.

for **HTML output**: Discard all options for lwarp-afterpage:

```
1 \LWR@ProvidesPackageDrop{afterpage}
```

```
2 \newcommand{\afterpage}[1]{#1}
```

Package 4

lwarp-algorithmicx.sty

75 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 [183.1](#).

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}
```

Package 5

lwarp-alltt.sty

76 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 }
```

Package 6

lwarp-amsthm.sty

77 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   \@xp\@ifdefinable\csname #2\endcsname{%
17     \global\@xp\let\csname end#2\endcsname\@endtheorem
18     \ifx *#1% unnumbered, need to get one more mandatory arg
19       \edef@tempa##1{%
20         \gdef\@xp\@nx\csname#2\endcsname{%
21           \@nx\@thm{\@xp\@nx\csname th@\the\thm@style\endcsname}%
22             {##1}}}%
23     \else % numbered theorem, need to check for optional arg
24       \def@tempa{\@oparg{\@ynthm{#2}} []}%
25     \fi
26 \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#2}}}% 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 }
```

Package 7

lwarp-balance.sty

78 Balance

Pkg `balance` Emulated.

for **HTML output**: Discard all options for lwarp-balance:

```
1 \LWR@ProvidesPackageDrop{balance}
```

```
2 \newcommand*{\balance}{}
```

```
3 \newcommand*{\nobalance}{}
```

Package 8

lwarp-bookmark.sty

79 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]{}

```

Package 9

lwarp-booktabs.sty

80 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

```

Package 10

lwarp-boxedminipage.sty

81 Boxedminipage

Pkg boxedminipage boxedminipage is superceded by boxedminipage2e.

for HTML output: `1 \LWR@loadnever{boxedminipage}{boxedminipage2e}`

Package 11

lwarp-boxedminipage2e.sty

82 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 }
```

Package 12

lwarp-ccaption.sty

83 Ccaption

Pkg `ccaption` `ccaption` is not used. The user is recommended to use `caption` instead.

for **HTML output:** `1 \LWR@loadnever{ccaption}{caption}`

Package 13

lwarp-changepage.sty

84 Changepage

Pkg `changepage` `changepage` is ignored.

for **HTML output:** Discard all options for `lwarp-changepage`:

```
1 \LWR@ProvidesPackageDrop{changepage}

2 \newif\ifoddpag
3 \DeclareRobustCommand{\checkoddpag}{\oddpagetrue}
4 \DeclareRobustCommand{\changetext}[5]{}
5 \DeclareRobustCommand{\changepag}[9]{}
6 \newenvironment{adjustwidth}[2]{}{}
7 \newenvironment{adjustwidth*}[2]{}{}
```

Package 14

lwarp-crop.sty

85 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] [] {}
```

Package 15

lwarp-cutwin.sty

86 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}}
```

Package 16

lwarp-dcolumn.sty

87 Dcolumn

Pkg `dcolumn` `dcolumn` is emulated during HTML output, and the `dcolumn` package is ignored.

```
1 \LWR@ProvidesPackageDrop{dcolumn}
```

Package 17

lwarp-draftwatermark.sty

88 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]{}

```

Package 18

lwarp-ellipsis.sty

89 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}

```

Package 19

lwarp-emptypage.sty

90 Emptypage

Pkg emptypage emptypage is ignored.

for **HTML output:** Discard all options for lwarp-emptypage:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

Package 20

lwarp-endnotes.sty

91 Endnotes

(Based on original code by JOHN LAVAGNINO.)

Pkg endnotes

```
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{\textsuperscript{\normalfont\theenmark}}}
11 \def\makeenmark{\@makeenmark}
```

Package 21

lwarp-enumerate.sty

92 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}`

Package 22

lwarp-enumitem.sty

93 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}
```

Package 23

lwarp-epigraph.sty

94 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]{flushnormal}
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] [] {}

```

Package 24

lwarp-eso-pic.sty

95 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] [] {}

```

Package 25

lwarp-everypage.sty

96 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]{}

```

Package 26

lwarp-everyshi.sty

97 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]{}

```

Package 27

lwarp-extramarks.sty

98 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}{}

```

Package 28

lwarp-fancybox.sty

99 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`.

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{0valbox}{#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*{-\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*{-\baselineskip}%
69 \LWRFB@LUseVerbatim{#1}%
70 \LWR@afterendverbatim%
71 }
72
73 \def\@BUseVerbatim[#1]#2{%
74 \LWR@atbeginverbatim{BVerbatim}\unskip\LWR@origvspace*{-\baselineskip}%
75 \LWRFB@UseVerbatim{#2}%
76 \LWR@afterendverbatim%
77 }

78 \end{warpHTML}

```

Package 29

lwarp-fancyhdr.sty

100 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*\fancyheadstyle}[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[1] {}
13 \newcommand*\footrulewidth[1] {}
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}
```

Package 30

lwarp-fancyvrb.sty

101 Fancyvrb

(Based on original code by TIMOTHY VAN ZANDT.)

`Pkg fancyvrb` fancyvrb is supported with some patches.

for HTML output:

```
1 \RequirePackage{xcolor}% for \convertcolorspec
2 \LWR@ProvidesPackagePass{fancyvrb}
```

for HTML output:

```
3 \begin{warpHTML}
```

Initial default patch for fancyvrb:

```
4 \fvset{frame=none}%
```

For `\VerbatimFootnotes`:

```
5 \renewcommand{\VerbatimFootnotes}{
6 \PackageError{lwarp}
7 {Verbatim footnotes are not yet supported by lwarp.}
8 {This may be improved some day.}
9 }
```

After the preamble is loaded, after any patches to `Verbatim`:

```
10 \AfterEndPreamble{
11 \LWR@traceinfo{Patching Verbatim.}
```

Remember the original definition of `Verbatim`:

```
12 \let\LWRFV@origVerbatim\Verbatim
```

`Env Verbatim` Patched to place the environment in a `fancyvrb` div, and the label in a `fancyvrblabel` div. Also corrects the left margin for line numbers. Also uses `VerbatimHTMLWidth` to control placement of line numbers on the right. Aligning the right margin requires knowing the width.

```
13 \renewcommand*{\Verbatim}{%
14 \LWR@forcenewpage
15 \LWRFV@origVerbatim%
16 }
```

`\LWR@FVstyle` Holds the style of the verbatim.

```
17 \newcommand*\LWR@FVstyle{-}
```

The following patches to `Verbatim` are executed at the start and end of the environment, depending on the choice of `frame`. Original code is from the `fancyvrb` package.

```
18 \newcommand*\LWR@fvstartnone}{%
19 \LWR@traceinfo{fvstartnone}%
20 \hbox to\z@{\LWR@atbeginverbatim[\LWR@FVstyle]{verbatim}}%
21 }
22
23 \newcommand*\LWR@fvendnone}{%
24 \LWR@traceinfo{fvendnone}%
25 \hbox to\z@{\LWR@afterendverbatim}%
26 }
27
28 \newcommand*\LWR@fvstartsingle}{%
29 \LWR@traceinfo{fvstartsingle}%
30 \LWR@fvstartnone%
31 \FV@BeginListFrame@Single%
32 }
33
34 \newcommand*\LWR@fvendsingle}{%
35 \LWR@traceinfo{fvendsingle}%
36 \FV@endListFrame@Single%
37 \LWR@fvendnone%
38 }
39
40 \newcommand*\LWR@fvstartline}{%
41 \LWR@traceinfo{fvstartline}%
42 \LWR@fvstartnone%
43 \FV@BeginListFrame@Lines%
44 }
45
46 \newcommand*\LWR@fvendline}{%
47 \LWR@traceinfo{fvendline}%
48 \FV@endListFrame@Lines%
49 \LWR@fvendnone%
50 }
```

The following patches select the start/left/right/end behaviors depending on `frame`. Original code is from the `fancyvrb` package.

```
51
52 \newcommand*\LWR@FVfindcurrentcolor}{%
53 \protect\colorlet{LWR@current@color}{.}%
54 \protect\convertcolorspec{named}{LWR@current@color}{HTML}\LWR@tempcolor%
```

```

55 }
56
57 \newcommand*{\LWR@FVtextstyle}{%
58 \LWR@FVfindcurrentcolor
59 \ifdefstring{\LWR@tempcolor}{000000}%
60 {}%
61 {color: \#\LWR@tempcolor ; }%
62 }
63
64
65 \newcommand*{\LWR@FVfindbordercolor}{%
66 \FancyVerbRuleColor%
67 \LWR@FVfindcurrentcolor
68 \color{black}
69 }
70
71 % border width of \FV@FrameRule
72 \newcommand*{\LWR@FVborderstyle}[1]{%
73 padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ;
74 \LWR@FVfindbordercolor
75 border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt solid \#\LWR@tempcolor ;
76 }
77
78 \def\FV@Frame@none{%
79 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle}%
80 \let\FV@BeginListFrame\LWR@fvstartnone%
81 \let\FV@LeftListFrame\relax%
82 \let\FV@RightListFrame\relax%
83 \let\FV@endListFrame\LWR@fvendnone}
84
85 \FV@Frame@none% default values
86
87 \def\FV@Frame@single{%
88 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{}}%
89 \let\FV@BeginListFrame\LWR@fvstartsingle%
90 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
91 \let\FV@RightListFrame\FV@RightListFrame@Single%
92 \let\FV@endListFrame\LWR@fvendsingle}
93
94 \def\FV@Frame@lines{%
95 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-top}\LWR@FVborderstyle{-bottom}}%
96 \let\FV@BeginListFrame\LWR@fvstartline%
97 \let\FV@LeftListFrame\relax%
98 \let\FV@RightListFrame\relax%
99 \let\FV@endListFrame\LWR@fvendline}
100
101 \def\FV@Frame@topline{%
102 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-top}}%
103 \let\FV@BeginListFrame\LWR@fvstartline%
104 \let\FV@LeftListFrame\relax%

```

```

105 \let\FV@RightListFrame\relax%
106 \let\FV@EndListFrame\LWR@fvendnone}
107
108 \def\FV@Frame@bottomline{%
109 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-bottom}}%
110 \let\FV@BeginListFrame\LWR@fvstartnone%
111 \let\FV@LeftListFrame\relax%
112 \let\FV@RightListFrame\relax%
113 \let\FV@EndListFrame\LWR@fvendline}
114
115 \def\FV@Frame@leftline{%
116 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-left}}%
117 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
118 \ifx\FancyVerbFillColor\relax%
119 \let\FV@FrameFillLine\relax%
120 \else%
121 \@tempdima\FV@FrameRule\relax%
122 \multiply\@tempdima-\tw@%
123 \edef\FV@FrameFillLine{%
124 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}%
125 \kern-\number\@tempdima sp}}%
126 \fi%
127 \let\FV@BeginListFrame\LWR@fvstartnone%
128 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
129 \let\FV@RightListFrame\relax%
130 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the fancyvrb package.

```

131 \def\FV@SingleFrameLine#1{%
132 \hbox to\z@{%
133 % \kern\leftmargin
134 \ifnum#1=\z@\relax
135 \let\FV@Label\FV@LabelBegin
136 \else
137 \let\FV@Label\FV@LabelEnd
138 \fi
139 \ifx\FV@Label\relax
140 % \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
141 \else
142 \ifnum#1=\z@
143 % \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
144 \ifx\FV@LabelPositionTopLine\relax
145 \else
146 \LWR@FVfindbordercolor
147 \LWR@htmltagc{div class="fancyvrblabel" style="color:\#\LWR@tempcolor"}
148 \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
149 \LWR@htmltagc{/div}

```

```

150 \fi
151     \else
152 %         \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
153         \ifx\FV@LabelPositionBottomLine\relax
154 \else
155 \LWR@FVfindbordercolor
156         \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
157 \LWR@origtextrm{\FV@LabelEnd}
158 \LWR@htmltagc{/div}
159 \fi
160     \fi
161
162     \fi
163     \hss
164 }
165 }

```

Processes each line, adding optional line numbers. Original code is from the fancyvrb package.

```

166 \def\FV@ListProcessLine#1{%
167     \hbox to \hsize{%
168 %         \kern\leftmargin
169         \hbox to \VerbatimHTMLWidth {%
170             \ifcvoid{FV@LeftListNumber}{}{\kern 2.5em}%
171 \FV@LeftListNumber%
172 %         \FV@LeftListFrame
173         \FancyVerbFormatLine{#1}%
174         \hss%
175 %         \FV@RightListFrame
176         \FV@RightListNumber%
177 }%
178     \hss% required to avoid underfull hboxes
179 }
180 }

```

Env BVerbatim

```

181 \AtBeginEnvironment{BVerbatim}
182 {
183 \LWR@forcenewpage
184 \LWR@atbeginverbatim{bverbatim}
185
186 }
187
188 \AfterEndEnvironment{BVerbatim}
189 {
190 \leavevmode\par\LWR@origvspace{-\baselineskip}
191 \LWR@afterendverbatim

```

192 }

End of the modifications to make at the end of the preamble:

193 } % \AfterEndPreamble

194 \end{warpHTML}

Package 31

lwarp-figcaps.sty

102 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}
```

Package 32

lwarp-float.sty

103 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 57.2 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 }
```

Package 33

lwarp-floatflt.sty

104 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}}
```

Package 34

lwarp-floatpag.sty

105 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]{}

```

Package 35

lwarp-floatrow.sty

106 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@infloatrow` 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 +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}{}

```

Preamble and default width are ignored.

```

57 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
58 \@namedef{#1}{
59 \floatbox{#2}
60 }
61 }

```

Preamble and default width are ignored.

```

62 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
63 \@namedef{#1}{%
64 \floatbox{#2}
65 }
66 }

```

```

67 \newfloatcommand{ffigbox}{figure}[\nocapbeside] []

```

```
68 \newfloatcommand{ttabbox}{table}[\captop] [\FBwidth]
```

```
69 \newfloatcommand{fcapside}{figure}[\capbeside] []
```

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}{}%
```

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}{}

```

To be used inside a minipage or parbox.

```

127 \newcommand{\RawCaption}[1]{#1}

```

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@infloatrow}
134 \boolfalse{LWR@infloatrow}
```

```
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@infloatrow` is set true, which tells `\floatbox` to use `\subfigure` or `\subtable`.

```
139 \begingroup
140 \booltrue{LWR@infloatrow}
141 }
142 {
143 \endgroup
144 \endBlockClass
145 \boolfalse{LWR@infloatrow}
146 }
```

Package 36

lwarp-flushend.sty

107 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}{}

```

Package 37

lwarp-fontenc.sty

108 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}

Package 38

lwarp-fontspec.sty

109 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}

Package 39

lwarp-footmisc.sty

110 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 }
```

Package 40

lwarp-footnote.sty

111 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}{\LWR@htmltagc{/\LWR@tagregularparagraph}}
```

Removed print-version formatting:

```

10 \def\fn@startfntext{%
11 \setbox\z@\vbox\bgroup%
12 \fn@startnote%
13 \fn@prefntext%
14 \ignorespaces%
15 }
```

Removed print-version formatting, added closing paragraph tag:

```

16 \def\fn@endfntext{%
17 \LWR@htmltagc{/\LWR@tagregularparagraph}%
18 \fn@postfntext%
19 \egroup%
20 \begingroup%
21 \let\@makefntext\@empty%
22 \let\@finalstrut\@gobble%
23 \LetLtxMacro\rule\@gobbletwo% *8* also the optional argument?
24 \@footnotetext{\unvbox\z@}%
25 \endgroup%
26 }
```

These have been redefined, so re-\let them again:

```
27 \let\endfootnote\fn@endfntext
28 \let\endfootnotetext\endfootnote
```

Package 41

lwarp-footnotehyper.sty

112 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}
```

Package 42

lwarp-framed.sty

113 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 [settings]

89 \let\MakeFramed\relax
90 \let\endMakeFramed\relax
91
92 \NewEnviron{MakeFramed}[1]{%
93 \FrameCommand{\begin{minipage}{\linewidth}\BODY\end{minipage}}%
94 }

\fb@put@frame [frame cmd no split] [frame cmd split]

95 \renewcommand*\fb@put@frame[2]{%
96 \relax%
97 \@tempboxa%
98 }
```

Package 43

lwarp-ftnright.sty

114 Ftnright

Pkg `ftnright` ftnright is ignored.

for **HTML output**: Discard all options for lwarp-ftnright:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

Package 44

lwarp-fullpage.sty

115 Fullpage

Pkg `fullpage` Emulated.

for **HTML output**: Discard all options for lwarp-fullpage:

```
1 \LWR@ProvidesPackageDrop{fullpage}
```

Package 45

lwarp-geometry.sty

116 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]{}

```

Package 46

lwarp-glossaries.sty

117 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.

`lwarpmk` has the commands `printglossary` and `htmlglossary` to process the glossaries created by `glossaries` using `xindy`.

Opt `IndexLanguage` The package `lwarp` 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.

for HTML output:

```
1 \PassOptionsToPackage{xindy}{glossaries}
2 \LWR@ProvidesPackagePass{glossaries}
3 \setupglossaries{nonumberlist}
4 \setglossarystyle{index}
```

Package 47

lwarp-graphics.sty

118 Graphics

Pkg graphics graphics is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphics}

Package 48

lwarp-graphicx.sty

119 Graphicx

Pkg graphicx graphicx is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphicx}

Package 49

lwrap-hyperref.sty

120 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 lwrap html version of package 'hyperref' -- discarding options.}
3 \typeout{   Are not using ProvidesPackage, so that other packages}
4 \typeout{   do not attempt to patch lwrap's version of 'hyperref'.}
5 % \ProvidesPackage{lwrap-#1-#2}
6 \DeclareOption*{}
7 \ProcessOptions\relax

8 \newcommand*\hypersetup}[1]{}
9 \newcommand*\hyperbaseurl}[1]{}

```

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 }

```

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 }

```

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 }

```

Creates text as an HTML link to the L^AT_EX label.

```

32 \NewDocumentCommand{\LWR@hyperrefc}{O{label} +m}{
33 \LWR@startref{#1}%
34 #2%
35 \LWR@htmltag{/a}%
36 \endgroup%
37 }

```

```

38 \newcommand{\hyperref}{%
39 \LWR@ensuredoingapar%
40 \begingroup\catcode'\_ =12
41 \@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
42 }

```

Creates an anchor to name with the given text.

```

43 \NewDocumentCommand{\hypertarget}{m +m}{%
44 \label{#1}%
45 #2%
46 }

```

Creates a link to the anchor created by `hypertarget`, with the given link text.

```

47 \NewDocumentCommand{\hyperlink}{m +m}{%
48 \hyperref{#1}{#2}%
49 }

```

For HTML, `\cleverref` is used instead.

```

50 \NewDocumentCommand{\autoref}{s m}{%
51 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
52 }

```

For HTML, `\cleverref` is used instead.

```

53 \NewDocumentCommand{\autopageref}{s m}{%
54 \IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
55 }

```

```

56 \newcommand{\pdfstringdef}[2]{}

```

```

57 \newcommand{\pdfbookmark}[3] [] {}

```

```
58 \newcommand{\currentpdfbookmark}[2]{}
```

```
59 \newcommand{\subpdfbookmark}[2]{}
```

```
60 \newcommand{\belowpdfbookmark}[2]{}
```

```
61 \newcommand{\texorpdfstring}[2]{#2}
```

From hyperref.

```
62 \def\hypercalcbp#1{%  
63 \strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax  
64 }%
```

```
65 \newcommand{\Acrobatmenu}[2]{}
```

```
66 \newcommand*{\TextField}[2] [] {}
```

```
67 \newcommand*{\CheckBox}[2] [] {}
```

```
68 \newcommand{\ChoiceMenu}[3] [] {}
```

```
69 \newcommand*{\PushButton}[2] [] {}
```

```
70 \newcommand*{\Submit}[2] [] {}
```

```
71 \newcommand*{\Reset}[2] [] {}
```

```
72 \newcommand*{\LayoutTextField}[2] {}
```

```
73 \newcommand*{\LayoutChoiceField}[2] {}
```

```
74 \newcommand*{\LayoutCheckField}[2] {}
```

```
75 \newcommand*{\MakeRadioField}[2] {}
```

```
76 \newcommand*{\MakeCheckField}[2] {}
```

```
77 \newcommand*{\MakeTextField}[2] {}
```

```
78 \newcommand*{\MakeChoiceField}[2] {}
```

```
79 \newcommand{\MakeFieldButton}[1] {}
```

Package 50

lwarp-hyperxmp.sty

121 Hyperxmp

Pkg hyperxmp Emulated.

for **HTML output:** Discard all options for lwarp-hyperxmp:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```

Package 51

lwarp-idxlayout.sty

122 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@indexsection
7 \LWR@startpars
8
9 \LWR@indexprenote
10
11 \LWR@origprintindex
12 }
13
14 \newcommand{\setindexprenote}[1]{\renewcommand{\LWR@indexprenote}{#1}}
15 \newcommand*\noindexprenote{\renewcommand{\LWR@indexprenote}{} }
16
17 \newcommand{\idxlayout}[1]{}
18 \newcommand*\indexfont{}
19 \newcommand*\indexjustific{}
20 \newcommand*\indexsubsdelim{}
21 \newcommand*\indexstheadcase{}
```

Package 52

lwarp-indentfirst.sty

123 Indentfirst

Pkg `indentfirst` `indentfirst` is ignored.

Discard all options for `lwarp-indentfirst`:

for HTML output: `1 \LWR@ProvidesPackageDrop{indentfirst}`

Package 53

lwarp-inputenc.sty

124 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}`

Package 54

lwarp-keyfloat.sty

125 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@maybeincthisfloat%
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
```

Package 55

lwarp-layout.sty

126 Layout

Pkg layout layout is ignored.

for HTML output: Discard all options for lwarp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}
2 \NewDocumentCommand{\layout}{s}{}

```

Package 56

lwarp-letterspace.sty

127 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}

```

Package 57

lwarp-lettrine.sty

128 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
27 \newcommand*{\LettrineFontHook}{}
28
29 \newcommand*{\LettrineFont}[1]{\InlineClass{lettrine}{#1}}
30 \newcommand*{\LettrineFontEPS}[1]{\includegraphics [height=1.5ex] {#1}}
```

Package 58

lwarp-lips.sty

129 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 {}
```

Package 59

lwarp-listings.sty

130 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}{1ol}{#1}{#2}}
```

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 }

25 \renewcommand*{\lst@DeInit}{%
26 \lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the `</pre>`, then reenable line numbers:

```
27 \let\lsthk@EveryPar\relax%
28
29 \LWR@afterendverbatim%
30 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
31 \else%
```

Inline, so create the closing ``:

```
32 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
33 \fi%
```

Final listings deinit:

```
34 \LWR@origlst@DeInit%
35 }
```

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\lst@name\lst@name
50 \lst@ifno101\else
51 \ifx\lst@@caption\empty
```

```

52             \ifx\lst@caption\@empty
53             \ifx\lst@intname\@empty \else \def\lst@temp{ }%
54             \ifx\lst@intname\lst@temp \else

```

This code places a contents entry for a non-float. This would have to be modified for lwarp:

```

55 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
56 %             \addcontentsline{lol}{\lstlisting}{\lst@name}
57             \fi\fi
58             \fi
59             \else

```

This would have to be modified for lwarp:

```

60 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
61             \addcontentsline{lol}{\lstlisting}%
62 {\protect\numberline{\thelstlisting}%
63 {\protect\ignorespaces \lst@@caption \protect\relax}}%
64             \fi
65             \fi
66             \fi
67             \ifx\lst@caption\@empty\else
68 \LWR@traceinfo{\lst@caption not empty-}%
69             \lst@ifsubstring #1\lst@captionpos
70             {\begingroup
71 \LWR@traceinfo{at the selected position}%

```

These space and box commands are not needed for HTML output:

```

72 %             \let\@@vskip\vskip
73 %             \def\vskip{\afterassignment\lst@vskip \@tempskipa}%
74 %             \def\lst@vskip{\nobreak\@@vskip\@tempskipa\nobreak}%
75 %             \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
76 %             \ifx #1t\allowbreak \fi
77             \ifx\lst@title\@empty

```

New lwarp code to create a caption:

```

78             \lst@makecaption\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@originormalfont%
110 \lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
111 % }
112 }
113 \\%
114     right&\def\lst@PlaceNumber{\rlap{\LWR@originormalfont
115         \kern\VerbatimHTMLWidth \kern\lst@numbersep
116         \lst@numberstyle{\thelstnumber}}}%
117     }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}

118 \end{warpHTML}

```

Package 60

lwarp-longtable.sty

131 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` * `[(horizontal)] {<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{\LTcapttype}%
17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{\LTcapttype}%
21 \let\captionlistentry\LWR@LTcaptionlistentry%
22 \LWR@tabular{#2}
23 }
24 {\endLWR@tabular\LWR@floatend}
25
```

Provided for compatibility, but ignored:

```
26 \newcounter{LTchunksize}
27 \def\endhead{\LWR@tabularendofline}% throws away options //[dim] and //*
28 \def\endfirsthead{\LWR@tabularendofline}
29 \def\endfoot{\LWR@tabularendofline}
30 \def\endlastfoot{\LWR@tabularendofline}
31 \newcommand\tabularnewline{\LWR@tabularendofline}
32 \newcommand{\setlongtables}{}% Obsolete command, does nothing.
33 \newlength{\LTleft}
34 \newlength{\LTRight}
35 \newlength{\LTpre}
36 \newlength{\LTpost}
37 \newlength{\LTcapwidth}

38 \renewcommand*{\kill}{\LWR@tabularendofline}
```

Package 61

lwarp-lscape.sty

132 Lscape

Pkg lscape lscape is nullified.

for HTML output: Discard all options for lwarp-lscape.

```
1 \LWR@ProvidesPackageDrop{lscape}
2 \newenvironment*{landscape}{}{}
```

Package 62

lwarp-ltcaption.sty

133 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}{}

```

Package 63

lwarp-marginfit.sty

134 Marginfit

Pkg marginfit Emulated.

for HTML output: Discard all options for lwarp-marginfit:

```
1 \LWR@ProvidesPackageDrop{marginfit}
```

Package 64

lwarp-marginfix.sty

135 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] [] {}
10 \newcommand*{\unblockmargin}[1] [] {}
11 \newcommand*{\marginphantom}[2] [] {}
```

Package 65

lwarp-marginnote.sty

136 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}{}

```

Package 66

lwarp-mcaption.sty

137 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}

```

Package 67

lwarp-mdframed.sty

138 Mdfamed

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 }
```

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 }
```

Given the `mdframed` key, print the length.

```
15 \newcommand*\LWR@mdfprintlength}[1]{%
16 \rndprintlength{\csuse{mdf@#1@length}}
17 }
```

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 modified `\hspace` and `\rule`, so restore them to their originals while inside `mdframed`:

```
38 \LetLtxMacro{\hspace}{\LWR@orighspace}%
39 \LetLtxMacro{\rule}{\LWR@origrule}%
40 }
```

Actions after an `mdframe` ends.

After closing the `<div>`, globally restore to the default environment type:

```
41 \newcommand*{\LWR@mdframedend}{
```

Close the custom `<div>`:

```
42 \LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
43 \gdef\LWR@mdthisenv{mdframed}
```

Resume paragraph handling:

```
44 \LWR@startpars%
45 }
```

Encapsulation of the original which places the title inside a `` of class `mdframedtitle`:

```
46 \LetLtxMacro\LWR@origmdframedtitleenv\mdframedtitleenv
47
48 \newlength{\LWR@titleroundcorner}
49
50 \renewrobustcmd\mdframedtitleenv[1]{%
51 \LWR@origmdframedtitleenv{%
```

Below, print HTML pt lengths:

```
52 \uselengthunit{PT}%
```

Open a `` with a custom class and custom style:

```
53 \LWR@htmltagc{span class="mdframedtitle" \LWR@orignewline
54 style=" \LWR@orignewline
```

Convert and print the title background color:

```
55 background:
56 \LWR@mdfprintcolor{frametitlebackgroundcolor}
57 ; \LWR@orignewline
```

Convert and print the title rule:

```
58 \ifbool{mdf@frametitlerule}{%
59 border-bottom:
60 \LWR@mdfprintlength{frametitlerulewidth}
61 solid
62 \LWR@mdfprintcolor{frametitlerulecolor}
63 ; \LWR@orignewline
64 }{}}%
```

The title's top border radius is adjusted for the line width:

```
65 border-radius:
66 \setlength{\LWR@titleroundcorner}
67 {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{Opt}}
68 \rndprintlength{\LWR@titleroundcorner}
69 \rndprintlength{\LWR@titleroundcorner}
70 Opt Opt
71 \LWR@orignewline
```

Finish the custom style and the opening span tag:

```
72 " \LWR@orignewline
73 }% span
```

Restrict paragraph tags inside a span:

```
74 \begin{LWR@nestspan}%
```

Print the title inside the span:

```
75 #1%
```

Close the span and unnest the paragraph tag restriction:

```
76 \LWR@htmltagc{/span}%
```

```
77 \end{LWR@nestspan}%
```

```
78 }
```

```
79 }
```

Common code for `\LWR@mdfsubtitle` and `\LWR@mdfsubsubtitle`.

Encapsulate the subtitle inside a `` of class `mdframedsubtitle`:

```
80 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
```

```
81 {% the following empty line is required
```

```
82
```

Special handling for `mdframed`: Subtitles have `\pars` around them, so temporarily disable them here.

```
83 \let\par\LWR@origpar%
```

Open a `` with a custom class and custom style:

```
84 \LWR@htmltagc{span class="mdframed#1title"
```

```
85 style=" \LWR@orignewline
```

Convert and print the background color:

```
86 background:
```

```
87 \LWR@mdfprintcolor{#1titlebackgroundcolor}
```

```
88 ; \LWR@orignewline
```

Convert and print the above line:

```
89 \ifbool{mdf@#1titleaboveline}{%
```

```
90 border-top:
```

```
91 \LWR@mdfprintlength{#1titleabovelinewidth}
```

```
92 solid
```

```
93 \LWR@mdfprintcolor{#1titleabovelinecolor}
```

```
94 ; \LWR@orignewline
```

```
95 }{}%
```

Convert and print the below line:

```

96 \ifbool{mdf@#1titlebelowline}{%
97 border-bottom:
98 \LWR@mdfprintlength{#1titlebelowlinewidth}
99 solid
100 \LWR@mdfprintcolor{#1titlebelowlinecolor}
101 ; \LWR@orignewline
102 }{}%
```

Finish the custom style and the opening span tag:

```
103 "% span
```

Restrict paragraph tags inside a span:

```
104 \begin{LWR@nestspan}%
```

Perform the original subtitle action:

```

105 \IfNoValueTF{#2}
106 {\csuse{LWR@origmdf#1title}{#3}}%
107 {\csuse{LWR@origmdf#1title}[#2]{#3}}%
```

Close the span and unnest the paragraph tag restriction:

```

108 \LWR@htmltagc{/span}% the following empty line is required
109 \end{LWR@nestspan}% must follow the /span or an extra <p> appears
110
111 }
```

```

112 \let\LWR@origmdfsubtitle\mdfsubtitle
113
114 \newcommand*{\LWR@mdfsubtitle}{%
115 \LWR@mdfsubtitlecommon{sub}%
116 }
117 \let\mdfsubtitle\LWR@mdfsubtitle
```

```

118 \let\LWR@origmdfsubsubtitle\mdfsubsubtitle
119
120 \newcommand*{\LWR@mdfsubsubtitle}{%
121 \LWR@mdfsubsubtitlecommon{subsub}%
122 }
123 \let\mdfsubsubtitle\LWR@mdfsubsubtitle
```

Stores the environment of the frame about to be created:

```
124 \newcommand*{\LWR@mdthisenv}{mdframed}
```

Modified from the original to remember the environment.

```

125 \renewrobustcmd*\newmdenv[2] [] {%
126 \newenvironment{#2}%
127 {%
128 \mdfsetup{#1}%
129 \renewcommand*\LWR@mdthisenv}{md#2}%
130 \begin{mdframed}%
131 }
132 {\end{mdframed}}%
133 }

```

Modified from the original to remember the environment.

```

134 \renewrobustcmd*\surroundwithmdframed[2] [] {%
135 \BeforeBeginEnvironment{#2}{%
136 \renewcommand*\LWR@mdthisenv}{md#2}%
137 \begin{mdframed}[#1]}%
138 \AfterEndEnvironment{#2}{\end{mdframed}}%
139 }

```

$[\langle numberedlike \rangle] \{\langle caption \rangle\} [\langle within \rangle]$

Modified from the original to remember the environment.

```

140 \DeclareDocumentCommand{\mdtheorem}{ 0{} m o m o }%
141 {\ifcsdef{#2}%
142   {\mdf@PackageWarning{Environment #2 already exists\MessageBreak}}%
143   {%
144     \IfNoValueTF {#3}%
145     {%#3 not given -- number relationship
146       \IfNoValueTF {#5}%
147       {%#3+#5 not given
148         \@definecounter{#2}%
149         \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
150         \newenvironment{#2}[1] [] {%
151           \refstepcounter{#2}%
152           \ifstrempy{##1}%
153             {\let\@temptitle\relax}%
154             {%
155               \def\@temptitle{\mdf@theoremseparator%
156                 \mdf@theoremspace%
157                 \mdf@theoremtitlefont%
158                 ##1}%
159               \mdf@thm@caption{#2}{#{4}{\csname the#2\endcsname}{##1}}%
160             }%
161             \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
162               \@temptitle}}}%
163           {\end{mdframed}}%
164         \newenvironment{#2*}[1] [] {%

```

```

165     \ifstrempy{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
166     \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]}%
167     {\end{mdframed}}%
168 }%
169 {%#5 given -- reset counter
170 \@definecounter{#2}\@newctr{#2}[#5]%
171 \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
172 \expandafter\xdef\csname the#2\endcsname{%
173     \expandafter\noexpand\csname the#5\endcsname \@thmcountersep%
174     \@thmcounter{#2}}%
175 \newenvironment{#2}[1][ ]{%
176     \refstepcounter{#2}%
177     \ifstrempy{##1}%
178     {\let\@temptitle\relax}%
179     {%
180         \def\@temptitle{\mdf@theoremseparator%
181             \mdf@theoremspace%
182             \mdf@theoremtitlefont%
183             ##1}%
184         \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%
185     }
186     \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
187         \@temptitle}]}%
188     {\end{mdframed}}%
189 \newenvironment{#2*}[1][ ]{%
190     \ifstrempy{##1}%
191     {\let\@temptitle\relax}%
192     {%
193         \def\@temptitle{\mdf@theoremseparator%
194             \mdf@theoremspace%
195             \mdf@theoremtitlefont%
196             ##1}%
197         \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%
198     }%
199     \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]}%
200     {\end{mdframed}}%
201 }%
202 }%
203 {%#3 given -- number relationship
204 \global\@namedef{the#2}{\@nameuse{the#3}}%
205 \newenvironment{#2}[1][ ]{%
206     \refstepcounter{#3}%
207     \ifstrempy{##1}%
208     {\let\@temptitle\relax}%
209     {%
210         \def\@temptitle{\mdf@theoremseparator%
211             \mdf@theoremspace%
212             \mdf@theoremtitlefont%
213             ##1}%
214         \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%

```

```

215         }
216         \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
217             \@temptitle}]]%
218         {\end{mdframed}}%
219         \newenvironment{#2*}[1] [] {%
220             \ifstrempy{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
221             \begin{mdframed}[#1,frametitle={\strut#4\@temptile}]]%
222             {\end{mdframed}}%
223         }%
224 \BeforeBeginEnvironment{#2}{\renewcommand*{\LWR@mdthisenv}{md#2}}% new
225 \BeforeBeginEnvironment{#2*}{\renewcommand*{\LWR@mdthisenv}{md#2}}% new
226     }%
227 }

```

[*numberedlike*] [*caption*] [*within*]

Modified from the original to remember the environment.

```

228 \DeclareDocumentCommand\newmdtheoremenv{0}{ m o m o }{%
229 \ifboolexpr{ test {\IfNoValueTF {#3}} and test {\IfNoValueTF {#5}} }%
230     {\newtheorem{#2}{#4}}{%
231     \IfValueTF{#3}{\newtheorem{#2}[#3]{#4}}{%
232     \IfValueTF{#5}{\newtheorem{#2}{#4}[#5]}{}}%
233     }%
234 \BeforeBeginEnvironment{#2}{%
235 \renewcommand*{\LWR@mdthisenv}{md#2}}%
236 \begin{mdframed}[#1]}%
237 \AfterEndEnvironment{#2}{%
238 \end{mdframed}}%
239 }

```

Package 68

lwarp-microtype.sty

139 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

```

Package 69

lwarp-moreverb.sty

140 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=\@ne%
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}
```

Package 70

lwarp-mparhack.sty

141 Mparhack

Pkg mparhack Emulated.

for **HTML output:** Discard all options for lwarp-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```

Package 71

lwarp-multicol.sty

142 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 `multicolshheading`.

The content is placed inside a `<div>` of class `multicols`.

```
2 \begin{warpHTML}
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 \IfValueTF{#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}
```

Package 72

lwarp-multirow.sty

143 Multirow

`Pkg multirow` multirow is emulated during HTML output, and the multirow package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{multirow}`

Package 73

lwarp-nameref.sty

144 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
```

Package 74

lwarp-needspace.sty

145 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}{}

```

Package 75

lwarp-newclude.sty

146 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}

```

Package 76

lwarp-newunicodechar.sty

147 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}
```

Package 77

lwarp-nextpage.sty

148 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] [] {}
```

Package 78

lwarp-nowidow.sty

149 Nowidow

Pkg `nowidow` `nowidow` is not used during HTML conversion.

Discard all options for `lwarp-nowidow`:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{nowidow}
2 \newcommand*{\nowidow}[1] [] {}
3 \newcommand*{\setnowidow}[1] [] {}
4 \newcommand*{\noclub}[1] [] {}
5 \newcommand*{\setnoclub}[1] [] {}
```

Package 79

lwarp-ntheorem.sty

150 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 L^AT_EX 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@ntheoremmarks}
5 \boolfalse{LWR@ntheoremmarks}
6
7 \DeclareOption{thmmarks}{

```

```

8 \booltrue{LWR@theoremmarks}
9 \newif\ifsetendmark\setendmarktrue
10 }
11
12
13 \newbool{LWR@theoremamsthm}
14 \boolfalse{LWR@theoremamsthm}
15
16 \DeclareOption{amsthm}{\booltrue{LWR@theoremamsthm}}
17
18
19 \DeclareOption{amsmath}{}
20 \DeclareOption{hyperref}{}
21
22
23 \LWR@ProvidesPackagePass{theorem}

```

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@thmstyle{\csuse{LWR@thmstyle#1}}}% new
52   \fi
53 }
54
55 \gdef\@ynthm#1#2{%
56   \ifthm@tempif
57     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
58     \expandafter\@ifundefined{c@#1}%
59       {\@definecounter{#1}}{}%
60     \expandafter\xdef\csname the#1\endcsname
61       {\noexpand\csname\the\theoremnumbering\endcsname{#1}}%
62     \expandafter\gdef\csname mkheader@#1\endcsname
63       {\csname setparms@#1\endcsname
64         \@thm{#1}{#1}{#2}
65 }%
66   \global\@namedef{end#1}{\@endtheorem}
67   \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\csuse{LWR@thmstyle#1}}}% new
68   \fi
69 }
70
71 \gdef\@othm#1[#2]#3{%
72   \@ifundefined{c@#2}{\@nocounterr{#2}}%
73   {\ifthm@tempif
74     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
75     \global\@namedef{the#1}{\@nameuse{the#2}}%
76     \expandafter\protected@xdef\csname num@addtheorem#1\endcsname{%
77       \noexpand\@num@addtheorem#1{#1}{#3}}%
78     \expandafter\protected@xdef\csname nonum@addtheorem#1\endcsname{%
79       \noexpand\@nonum@addtheorem#1{#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@intheorem% new
104 #2}%
105   \def\@opargbegintheorem####1####2####3{%
106 \LWR@forcenewpage% new
107 \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
108 \LWR@intheorem% new
109 #3}%
110 }%
111 }%
112 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
113 }
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{nonnumberplain}%
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{nonnumberbreak}%
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@ntheoremamsthm}{-}{-}
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@intheorem% 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@intheorem% 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@intheorem% 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@intheorem% 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@intheorem% 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@intheorem% 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@intheorem% 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@intheorem% 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@intheorem% 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@htmlmlelementclass{nav}{lothm}%
422 \begingroup
423 \c@tocdepth=-2%
424 \def\thm@list{#1}\thm@processlist
425 \endgroup
426 \LWR@htmlmlelementclassend{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}}
```

Package 80

lwarp-pagenote.sty

151 Pagenote

Pkg pagenote pagenote works as-is.

It is only included as an `lwarp-pagenote.sty` file because past versions of `lwarp` used `pagenote` to emulate footnotes, and so the file may exist on current installations, and should be over-written by this newer version.

for HTML output: `1 \LWR@ProvidesPackagePass{pagenote}`

Package 81

lwarp-paralist.sty

152 Paralist

Pkg paralist paralist is supported with minor changes.

for HTML output: `1 \LWR@ProvidesPackagePass{paralist}`

for HTML output: `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}}}`

Package 82

lwarp-parskip.sty

153 Parskip

Pkg `parskip` parskip is ignored.

for **HTML output**: Discard all options for lwarp-parskip.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

Package 83

lwarp-pdfscape.sty

154 Pdfscape

Pkg `pdfscape` Emulated.

for **HTML output**: Discard all options for lwarp-pdfscape:

```
1 \LWR@ProvidesPackageDrop{pdfscape}
```

Package 84

lwarp-pdfsync.sty

155 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
```

Package 85

lwarp-placeins.sty

156 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}{}
```

Package 86

lwarp-prelim2e.sty

157 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}{}

```

Package 87

lwarp-ragged2e.sty

158 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}

```

Package 88

lwarp-rotating.sty

159 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
```

Package 89

lwarp-rotfloat.sty

160 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 }

```

Package 90

lwarp-savetrees.sty

161 Savetrees

Pkg `savetrees` Emulated.

for **HTML output**: Discard all options for `lwarp-savetrees`:

```
1 \LWR@ProvidesPackageDrop{savetrees}
```

Package 91

lwarp-setspace.sty

162 Setspace

Pkg `setspace` `setspace` is not used during HTML conversion.

Discard all options for `lwarp-setspace`:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{setspace}
2
3 \newcommand*\setstretch[1]{}
4 \newcommand*\SetSinglespace[1]{}
5 \newcommand*\singlespacing{}
6 \newcommand*\onehalfspacing{}
7 \newcommand*\doublespacing{}
8
9 \newenvironment{singlespace}
10 {
11 \LWR@forcenewpage
12 \BlockClass{singlespace}
13 }
14 {\endBlockClass}
15
16 \newenvironment{singlespace*}
17 {
18 \LWR@forcenewpage
19 \BlockClass{singlespace}
20 }
21 {\endBlockClass}
22
23 \newenvironment{spacing}[1]{
24
25 }{
26
27 }
28
29 \newenvironment{onehalfspace}
30 {
31 \LWR@forcenewpage
32 \BlockClass{onehalfspace}
33 }
34 {\endBlockClass}
35
36 \newenvironment{doublespace}
37 {

```

```
38 \LWR@forcenewpage
39 \BlockClass{doublespace}
40 }
41 {\endBlockClass}
```

Package 92

lwarp-shadow.sty

163 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 }
```

Package 93

lwarp-showidx.sty

164 Showidx

Pkg `showidx` `showidx` is ignored.

for **HTML output:** Discard all options for `lwarp-showidx`:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

Package 94

lwarp-showkeys.sty

165 Showkeys

Pkg `showkeys` `showkeys` is ignored.

for **HTML output:** Discard all options for `lwarp-showkeys`:

```
1 \LWR@ProvidesPackageDrop{showkeys}
```

```
2 \NewDocumentCommand{\showkeys}{s}{} 
```

Package 95

lwarp-sidecap.sty

166 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}{}{}
```

Package 96

lwarp-sidenotes.sty

167 Sidenotes

(Based on original code by ANDY THOMAS, OLIVER SCHEBAUM.)

Pkg sidenotes Patched for lwarp.

for HTML output: Load the original package:

```
1 \LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with lwarp:

Stop paragraph handling while creating the caption:

```
2 \RenewDocumentCommand \sidecaption {s o o m}
3 {
4 \LWR@stoppars
5 \captionsetup{style=sidecaption}
6 \IfBooleanTF{#1}
7 { % starred
8 \IfNoValueOrEmptyTF{#2}
9 {\marginnote{\caption*{#4}}}
10 {\marginnote{\caption*{#4}}[#2]}
11 }
12 { % unstarred
13 \IfNoValueOrEmptyTF{#2}
14 {\def\@sidenotes@sidecaption@tof{#4}}
15 {\def\@sidenotes@sidecaption@tof{#2}}
16 \IfNoValueOrEmptyTF{#3}
17 {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}}
18 {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}[#3]}
19 }
20 \LWR@startpars
21 }
```

Borrowed from the lwarp version of keyfloat:

```
22 \NewDocumentEnvironment{KFLT@sidenotes@marginfloat}{0{-1.2ex} m}
23 {% start
24 \LWR@maybeinthisfloat%
25 \LWR@forcenewpage
26 \LWR@stoppars%
27 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}}
```

```
28 \LWR@startpars%
29 \captionsetup{type=#2}%
30 }
31 {
32 \LWR@htmldivclassend{div}
33 }
34
35 \RenewDocumentEnvironment{marginfigure}{o}
36   {\begin{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 L^AT_EX 2_ε `article.cls` source:

```
42 \renewenvironment{figure*}
43     {\@dblfloat{figure}}
44     {\end@dblfloat}
45
46 \renewenvironment{table*}
47     {\@dblfloat{table}}
48     {\end@dblfloat}
```

Package 97

lwarp-soul.sty

168 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 \ifcsempy{#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*\capsselect}[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

```

Package 98

lwarp-subfig.sty

169 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           \@tempkipb\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 {\langle caption label \rangle} {\langle caption text \rangle}

165 \renewcommand\caption@@@make[2]{%
166 \LWR@startpars% new

```

```

167 \sbox\@tempboxa{#1}%
168 \ifdim\wd\@tempboxa=\z@
169   \let\caption@lsep\relax
170 \fi
171 \caption@ifempty{#2}{%
172   \let\caption@lsep\@empty
173   \let\caption@tfmt\@firstofone
174 }%
175 % \@setpar{\@par\caption@@par}\caption@@par
176 \renewcommand{\@par}{\LWR@closeparagraph\LWR@orig@@par}% new
177 \caption@applyfont
178 \caption@fmt
179   {\ifcaption@star\else
180     \begingroup
181       \captionlabelfont
182       #1%
183     \endgroup
184   \fi}%
185   {\ifcaption@star\else
186     \begingroup
187       \caption@iflf\captionlabelfont
188       \relax\caption@lsep
189     \endgroup
190   \fi}%
191   {\captiontextfont
192     \caption@ifstrut
193     {\vrule\@height\ht\strutbox\@width\z@}%
194     {}%
195     \nobreak\hskip\z@skip % enable hyphenation
196     \caption@tfmt{#2}
197 \LWR@ensuredoingapar% new
198   \caption@ifstrut
199     {\ifhmode\@finalstrut\strutbox\fi}%
200     {}%
201   \par}}
202 \LWR@stoppars% new
203 }

```

Patches for \sf@sub@label:

```

204 \def\subfloat@label{%
205 \LWR@ensuredoingapar% new
206 \ifnextchar(% %) match left parenthesis
207   {\sf@sub@label}
208   {\sf@sub@label(Sub\@capytype\space
209     \@ifundefined{thechapter}{\@nameuse{thechapter}\space}%
210     \@nameuse{p@sub\@capytype}%
211     \@nameuse{thesub\@capytype}.)}}

```

Patches for \subref.

The unstarred version uses a `\ref` link whose printed text comes from the `sub@<label>`:

```
212 \renewcommand{\sf@subref}[1]{%
213 \LWR@subnewref{#1}{sub@#1}%
214 }
```

The starred version uses the printed `sub@<label>` which is stored as if it were a page number:

```
215 \renewcommand{\sf@subref}[1]{\LWR@origpageref{sub@#1}}
```

Defining new subfloats. The `l@sub<type>` for each is redefined.

```
216 \LetLtxMacro\LWR@orig@newsubfloat\@newsubfloat
217
218 \def\@newsubfloat[#1]#2{%
219 \LWR@orig@newsubfloat[#1]{#2}%
220 \renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}
221 }
```

Pre-defined for figures and tables:

```
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}
```

Package 99

lwarp-syntonly.sty

170 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
```

Package 100

lwarp-tabularx.sty

171 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}
```

Package 101

lwarp-tabulary.sty

172 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{}
```

Package 102

lwarp-textpos.sty

173 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] {}

```

Package 103

lwarp-theorem.sty

174 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}

```

Package 104

lwarp-threeparttable.sty

175 Threeparttable

Pkg `threeparttable` `threeparttable` is emulated during HTML output, and the `threeparttable` package is ignored.

Table note are contained inside a CSS `<div>` of class `tnotes`. If `enumitem` is used, the note item labels are also individually highlighted with an additional CSS `` of class `tnoteitemheader`, otherwise they are plain text.

for HTML output: `1 \LWR@ProvidesPackageDrop{threeparttable}`

Prints the table note item header inside a CSS class of `tnoteitemheader`.

```
2 \newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}
```

To emulate `threeparttable`:

```
3 \newenvironment*{threeparttable}[1][b]{}{}
```

```
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 }
```

```
17 \newcommand{\tnote}[1]{\textsuperscript{#1}}
```

Package 105

lwarp-tikz.sty

176 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}

Package 106

lwarp-titles.sty

177 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.

```

2 \NewDocumentCommand{\newpagestyle}{m o m}{}
3 \NewDocumentCommand{\renewpagestyle}{m o m}{}

4 \NewDocumentCommand{\sethead}{o o o m m m}{}
5 \NewDocumentCommand{\setfoot}{o o o m m m}{}

6 \NewDocumentCommand{\setttitlemarks}{s m}{}

7 \newcommand*{\headrule}{}
8 \newcommand*{\footrule}{}

9 \newcommand*{\setheadrule}[1]{}
10 \newcommand*{\setfootrule}[1]{}

11 \newcommand*{\makeheadrule}{}
12 \newcommand*{\makefootrule}{}

13 \newcommand{\setmarkboth}[1]{}

14 \NewDocumentCommand{\widenhead}{s o o m m}{}

15 \newcommand*{\bottitlemarks}{}
16 \newcommand*{\toptitlemarks}{}
17 \newcommand*{\firsttitlemarks}{}
18 \newcommand*{\nexttoptitlemarks}{}
19 \newcommand*{\outertitlemarks}{}
20 \newcommand*{\innertitlemarks}{}

21 \NewDocumentCommand{\newtitlemark}{s m}{}

```

```
22 \NewDocumentCommand{\pretitlemark}{s m m}{}

23 \newcommand{\ifsamemark}[4]{}

24 \NewDocumentCommand{\setfloathead}{s o o m m m m}{}
25 \NewDocumentCommand{\setfloatfoot}{s o o m m m m}{}

26 \NewDocumentCommand{\nextfloathead}{s o o m m m m}{}
27 \NewDocumentCommand{\nextfloatfoot}{s o o m m m m}{}

28 \newcommand{\newmarkset}[1]{}

29 \NewDocumentCommand{\newextramarkset}{s m m}{}

30 \newcommand{\botextramarks}[1]{}
31 \newcommand{\topextramarks}[1]{}
32 \newcommand{\firstextramarks}[1]{}
33 \newcommand{\nexttoextramarks}[1]{}
34 \newcommand{\outerextramarks}[1]{}
35 \newcommand{\innerextramarks}[1]{}

```

Package 107

lwarp-titleref.sty

178 Titleref

Pkg titleref titleref is superceded by hyperref and nameref.

for **HTML output:** 1 \LWR@loadnever{titleref}{hyperref and nameref}

Package 108

lwarp-titlesec.sty

179 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}

2 \newcommand*{\titlelabel}[1]{}

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}{}

8 \@ifundefined{chapapp}{\let\chapapp\chaptername}{}
9 \newcommand\chaptertitlename{\@chapapp}

10 \NewDocumentCommand{\titlespacing}{s m m m m o}{}

11 \newcommand*{\filright}{}
12 \newcommand*{\filcenter}{}
13 \newcommand*{\filleft}{}
14 \newcommand*{\fillast}{}
15 \newcommand*{\filinner}{}
16 \newcommand*{\filouter}{}

17 \newcommand\wordsep{\fontdimen\tw@\font \@plus
18   \fontdimen\thr@\font \@minus \fontdimen4\font}

19 \NewDocumentCommand{\titleline}{s o m}{}

20 \providecommand\titlerule{\@ifstar{\ttl@row}{\ttl@rule}}
21 \newcommand*{\ttl@rule}[1] [] {}
22 \newcommand*{\ttl@row}[2] [] {}

23 \newcommand{\iftitlemeasuring}[2] {#2}

24 \newcommand{\assignpagestyle}[2] {#2}

25 \NewDocumentCommand{\titleclass}{m o m o}

```

Package 109

lwarp-titletoc.sty

180 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}

2 \NewDocumentCommand{\dottedcontents}{m o m m m}{\}

3 \newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
4 \NewDocumentCommand{\ttl@tcstar}{m o m m m o o}{\}
5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m o}{\}

6 \newcommand{\contentsmargin}[2] [] {}

7 \newcommand*{\thecontentslabel}{\thecontentslabel}
8 \newcommand*{\thecontentspage}{\thecontentspage}

9 \newcommand{\contentslabel}[2] [] {\thecontentslabel}
10 \newcommand{\contentspage}[1] [] {\thecontentspage}

11 \newcommand{\contentspush}[1] {}

12 \newcommand{\contentsuse}[2] {}

13 \newcommand*{\startcontents}[1] [] {}
14 \newcommand*{\stopcontents}[1] [] {}
15 \newcommand*{\resumecontents}[1] [] {}

16 \newcommand{\printcontents}[4] [] {}

17 \newcommand{\startlist}[2] [] {}
18 \newcommand{\stoplist}[2] [] {}
19 \newcommand{\resumelist}[2] [] {}

20 \newcommand{\printlist}[4] [] {}

```

Package 110

lwarp-titling.sty

181 Titling

Pkg titling titling is used by lwarp. The following patches are not needed by lwarp, but are required if the user requests titling.

lwarp uses page notes for footnotes, so the various titling footnote restyling commands have no effect.

Pass all options to lwarp-titling:

for HTML output: `1 \LWR@ProvidesPackagePass{titling}`

Patch `\@bsmtitleempty`:

```
2 \let\LWR@orig@bsmtitleempty\@bsmtitleempty
3 \renewcommand*\@bsmtitleempty}{%
4 \LWR@orig@bsmtitleempty%
5 \global\let\published\relax%
6 \global\let\subtitle\relax%
7 }
```

Patch `\keepthetitle`:

```
8 \let\LWR@origkeepthetitle\keepthetitle
9 \renewcommand*\keepthetitle}{%
10 \LWR@orig@keepthetitle%
11 \global\let\@published\@empty%
12 \global\let\@subtitle\@empty%
13 }
```

Patch `\killtitle`:

```
14 \let\LWR@origkilltitle\killtitle
15 \renewcommand*\killtitle}{%
16 \LWR@orig@killtitle%
17 \global\let\thepublished\relax%
18 \global\let\thesubtitle\relax%
19 }
```

Package 111

lwarp-tocloft.sty

182 Tocloft

Pkg `tocloft` `tocloft` is emulated. Most user options and macros are ignored and disabled. `\newlistof` and `\cftchapterprecis` are supported.

Discard all options for `lwarp-tocloft`:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{tocloft}

2 \newcommand{\tocloftpagestyle}[1]{}

3 \newcommand*\cftmarktoc{}
4 \newcommand*\cfttoctitlefont{}
5 \newcommand*\cftaftertoctitle{}

6 \newlength{\cftbeforetoctitleskip}
7 \newlength{\cftaftertoctitleskip}

8 \newcommand*\cftmarklof{}
9 \newcommand*\cftlofttitlefont{}
10 \newcommand*\cftafterlofttitle{}

11 \newlength{\cftbeforelofttitleskip}
12 \newlength{\cftafterlofttitleskip}

13 \newcommand*\cftmarklot{}
14 \newcommand*\cftlottitlefont{}
15 \newcommand*\cftafterlottitle{}

16 \newlength{\cftbeforelottitleskip}
17 \newlength{\cftafterlottitleskip}

18 \newcommand*\cftdot}{.}
19 \providecommand*\cftdotsep}{1}
20 \newcommand*\cftnodots}{5000}
21
22 \providecommand{\cftdotfill}[1]{}

23 \newcommand*\cftsetpnumwidth}[1]{}
24 \newcommand*\cftsetrmarg}[1]{}

```

```
25 \newcommand*\cftpnumalign}[1]{}

26 \newlength{\cftparskip}

27 \newlength{\cftbeforepartskip}
28 \newlength{\cftpartindent}
29 \newlength{\cftpartnumwidth}
30 \newcommand*\cftpartfont{}
31 \newcommand*\cftpartpresnum{}
32 \newcommand*\cftpartaftersnum{}
33 \newcommand*\cftpartaftersnumb{}
34 \newcommand*\cftpartleader{}
35 \newcommand*\cftpartdotsep}{1}
36 \newcommand*\cftpartpagefont{}
37 \newcommand*\cftpartafterpnum{}

38 \newlength{\cftbeforechapskip}
39 \newlength{\cftchapindent}
40 \newlength{\cftchapnumwidth}
41 \newcommand*\cftchapfont{}
42 \newcommand*\cftchappresnum{}
43 \newcommand*\cftchapaftersnum{}
44 \newcommand*\cftchapaftersnumb{}
45 \newcommand*\cftchapleader{}
46 \newcommand*\cftchapdotsep}{1}
47 \newcommand*\cftchappagefont{}
48 \newcommand*\cftchapafterpnum{}

49 \newlength{\cftbeforesecskip}
50 \newlength{\cftsecindent}
51 \newlength{\cftsecnumwidth}
52 \newcommand*\cftsecfont{}
53 \newcommand*\cftsecpresnum{}
54 \newcommand*\cftsecaftersnum{}
55 \newcommand*\cftsecaftersnumb{}
56 \newcommand*\cftsecleader{}
57 \newcommand*\cftsecdotsep}{1}
58 \newcommand*\cftsecpagefont{}
59 \newcommand*\cftsecafterpnum{}

60 \newlength{\cftbeforesubsecskip}
61 \newlength{\cftsubsecindent}
62 \newlength{\cftsubsecnumwidth}
63 \newcommand*\cftsubsecfont{}
64 \newcommand*\cftsubsecpresnum{}
65 \newcommand*\cftsubsecaftersnum{}
66 \newcommand*\cftsubsecaftersnumb{}
67 \newcommand*\cftsubsecleader{}
68 \newcommand*\cftsubsecdotsep}{1}
```

```
69 \newcommand*\cftsubsecpagefont}{  
70 \newcommand*\cftsubsecafterpnum}{  
  
71 \newlength{\cftbeforesubsubsecskip}  
72 \newlength{\cftsubsubsecindent}  
73 \newlength{\cftsubsubsecnumwidth}  
74 \newcommand*\cftsubsubsecfont}{  
75 \newcommand*\cftsubsubsecpresnum}{  
76 \newcommand*\cftsubsubsecaftersnum}{  
77 \newcommand*\cftsubsubsecaftersnumb}{  
78 \newcommand*\cftsubsubsecleader}{  
79 \newcommand*\cftsubsubsecdotsep}{1}  
80 \newcommand*\cftsubsubsecpagefont}{  
81 \newcommand*\cftsubsubsecafterpnum}{  
  
82 \newlength{\cftbeforeparaskip}  
83 \newlength{\cftparaindent}  
84 \newlength{\cftparanumwidth}  
85 \newcommand*\cftparafont}{  
86 \newcommand*\cftparapresnum}{  
87 \newcommand*\cftparaaftersnum}{  
88 \newcommand*\cftparaaftersnumb}{  
89 \newcommand*\cftparaleader}{  
90 \newcommand*\cftparadotsep}{1}  
91 \newcommand*\cftparapagefont}{  
92 \newcommand*\cftparaafterpnum}{  
  
93 \newlength{\cftbeforesubparaskip}  
94 \newlength{\cftsubparaindent}  
95 \newlength{\cftsubparanumwidth}  
96 \newcommand*\cftsubparafont}{  
97 \newcommand*\cftsubparapresnum}{  
98 \newcommand*\cftsubparaaftersnum}{  
99 \newcommand*\cftsubparaaftersnumb}{  
100 \newcommand*\cftsubparaleader}{  
101 \newcommand*\cftsubparadotsep}{1}  
102 \newcommand*\cftsubparapagefont}{  
103 \newcommand*\cftsubparaafterpnum}{  
  
104 \newlength{\cftbeforefigskip}  
105 \newlength{\cftfigindent}  
106 \newlength{\cftfignumwidth}  
107 \newcommand*\cftfigfont}{  
108 \newcommand*\cftfigpresnum}{  
109 \newcommand*\cftfigaftersnum}{  
110 \newcommand*\cftfigaftersnumb}{  
111 \newcommand*\cftfigleader}{  
112 \newcommand*\cftfigdotsep}{1}  
113 \newcommand*\cftfigpagefont}{
```

```

114 \newcommand*\cftfigafterpnum}{ }

115 \newlength{\cftbeforesubfigskip}
116 \newlength{\cftsubfigindent}
117 \newlength{\cftsubfignumwidth}
118 \newcommand*\cftsubfigfont}{ }
119 \newcommand*\cftsubfigpresnum}{ }
120 \newcommand*\cftsubfigaftersnum}{ }
121 \newcommand*\cftsubfigaftersnumb}{ }
122 \newcommand*\cftsubfigleader}{ }
123 \newcommand*\cftsubfigdotsep}{1}
124 \newcommand*\cftsubfigpagefont}{ }
125 \newcommand*\cftsubfigafterpnum}{ }

126 \newlength{\cftbeforetabskip}
127 \newlength{\cfttabindent}
128 \newlength{\cfttabnumwidth}
129 \newcommand*\cfttabfont}{ }
130 \newcommand*\cfttabpresnum}{ }
131 \newcommand*\cfttabaftersnum}{ }
132 \newcommand*\cfttabaftersnumb}{ }
133 \newcommand*\cfttableader}{ }
134 \newcommand*\cfttabdotsep}{1}
135 \newcommand*\cfttabpagefont}{ }
136 \newcommand*\cfttabafterpnum}{ }

137 \newlength{\cftbeforesubtabskip}
138 \newlength{\cftsubtabindent}
139 \newlength{\cftsubtabnumwidth}
140 \newcommand*\cftsubtabfont}{ }
141 \newcommand*\cftsubtabpresnum}{ }
142 \newcommand*\cftsubtabaftersnum}{ }
143 \newcommand*\cftsubtabaftersnumb}{ }
144 \newcommand*\cftsubtableader}{ }
145 \newcommand*\cftsubtabdotsep}{1}
146 \newcommand*\cftsubtabpagefont}{ }
147 \newcommand*\cftsubtabafterpnum}{ }

148 \newcommand{\cftsetindents}[3]{ }

149 \newcommand{\pagenumbersoff}[1]{ }
150 \newcommand{\pagenumberon}[1]{ }

Emulated through the \newfloat mechanism.

151 \NewDocumentCommand{\newlistof}{o m m}
152 { %
153 \IfValueTF{#1}

```

```
154 {\newfloat{#2}{tbp}{#3}[#1]}
155 {\newfloat{#2}{tbp}{#3}}
156 \@namedef{listof#2}{\listof{#2}{#4}}
157 \@namedef{#2depth}{1}
158 \expandafter\newlength\csuse{cftbefore#2skip}
159 \expandafter\newlength\csuse{cft#2indent}
160 \expandafter\newlength\csuse{cft#2numwidth}
161 \@namedef{cft#2font}{}
162 \@namedef{cft#2presnum}{}
163 \@namedef{cft#2aftersnum}{}
164 \@namedef{cft#2aftersnumb}{}
165 \@namedef{cft#2leader}{}
166 \@namedef{cft#2dotsep}{1}
167 \@namedef{cft#2pagefont}{}
168 \@namedef{cft#2afterpnum}{}
169 }

\cftchapterprecis from tocloft:

170 \newcommand{\cftchapterprecis}[1]{%
171   \cftchapterprecishere{#1}
172   \cftchapterprecistoc{#1}}
173 \newcommand{\cftchapterprecishere}[1]{%
174   \begin{quote}\textit{#1}\end{quote}}
175 \newcommand{\cftchapterprecistoc}[1]{
176   \addtocontents{toc}{%
177     {
178       \protect\begin{quote}#1\protect\end{quote}}
179   }
180 }
```

Package 112

lwarp-trivfloat.sty

183 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}
```

183.1 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}
```

Package 113

lwarp-ulem.sty

184 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}
```

Package 114

lwarp-verse.sty

185 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
Pkg memoir \flagverse command. Do not allow them to go into the left margin, which would
\flagverse cause pdfcrop to crop the entire page further to the left:

Len \leftskip
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

```

Package 115

lwarp-wallpaper.sty

186 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[2]{}
17 \newlength{\wpXoffset}
18 \newlength{\wpYoffset}
```

Package 116

lwarp-wrapfig.sty

187 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
4 \newcommand*\LWR@wrapposition{}
5
6 \newcommand*\LWR@subwrapfigure[2]{%
7 \LWR@maybeinthisfloat%
8 \renewcommand*\LWR@wrapposition{}%
9 \ifthenelse{%
10 \equal{#1}{r}\OR\equal{#1}{R}\OR%
11 \equal{#1}{o}\OR\equal{#1}{O}%
12 }%
13 {\renewcommand*\LWR@wrapposition}{float:right}}%
14 {\renewcommand*\LWR@wrapposition}{float:left}}%
15 \setlength{\LWR@wrapwidth}{#2}%
16 \addtolength{\LWR@wrapwidth}{4em}%
17 \uselengthunit{PT}%
18 \LWR@forcenewpage
19 \LWR@stoppars%
20 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{\LWR@thisfloat}"
21 style="width:\rndprintlength{\LWR@wrapwidth} ; %
22 \LWR@wrapposition"%
23 }
24 \LWR@startpars
25 }
26
27
28 \NewDocumentEnvironment{wrapfigure}{o m o m}
29 {%
30 \LWR@subwrapfigure{#2}{#4}%
31 \captionsetup{type=figure}%
32 }
33 {
34 \LWR@htmldivclassend{div}
```

```
35 }
36
37
38 \NewDocumentEnvironment{wraptable}{o m o m}
39 {%
40 \LWR@subwrapfigure{#2}{#4}%
41 \captionsetup{type=table}%
42 }
43 {
44 \LWR@htmldivclassend{div}
45 }
46
47
48 \NewDocumentEnvironment{wrapfloat}{m o m o m}
49 {%
50 \LWR@subwrapfigure{#3}{#5}%
51 \captionsetup{type=#1}%
52 }
53 {
54 \LWR@htmldivclassend{div}
55 }
56
57 \newlength{\wrapoverhang}
```

Package 117

lwarp-xcolor.sty

188 Xcolor

Pkg `xcolor` `xcolor` is supported by `lwarp`.

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.

for HTML output:

```
1 \LWR@ProvidesPackagePass{xcolor}
```

```
2 \newcommand*\LWR@tempcolor{}
```

defaulting to black.

```
3 \newcommand*\LWR@currenttextcolor{black}
```

```
\LWR@colorstyle {<1: styletext>} {<2: model>} {<3: color>} {<4: spancontents>}
```

Creates a styled span with a color converted to HTML hex colorspace. Uses `LWR@spandepth` to prevent paragraph tags inside the span. If used for `\textcolor`, with a `styletext` of `color:`, then the new color is copied into `\LWR@currenttextcolor` for possible re-use in `\rule`.

```
4 \NewDocumentCommand{\LWR@colorstyle}{m m m m}{%
```

Use the `xcolor` package to convert to an HTML color space:

```
5 \convertcolorspec{#2}{#3}{HTML}\LWR@tempcolor%
```

If is a `\textcolor`, save a copy of this color for use by `\rule`:

```
6 \ifthenelse{\equal{#1}{color:}}{%
```

```
7 {\renewcommand*\LWR@currenttextcolor}{\#\LWR@tempcolor}}{}}%
```

Create the HTML `` with the styled color:

```
8 \LWR@htmltagc{span style="#1\#\LWR@tempcolor"{}}%
9 \begin{LWR@nestspan}%
```

Prevent additional paragraph tags inside this span:

Print the contents then close the span:

```
10 #4%
11 \LWR@htmltagc{/span}%
12 \end{LWR@nestspan}%
```

For paragraph-tag handling:

```
13 \LWR@ensuredoingapar%
14 }
```

`\color` appears in the L^AT_EX PDF output, but is ignored by `pdftotext` and thus is ignored in the HTML file. Text styling by local group is not yet supported.

Each of the following macros is given a temporary name, and is `\let` to the final name once the HTML conversion starts.

`\textcolor` [*model*] {*color*} {*text*} is converted into an HTML hex color span.

```
15 \NewDocumentCommand{\LWR@textcolor}{0{named} m m}{%
16 \begingroup%
17 \LWR@colorstyle{color:}{#1}{#2}{#3}%
18 \endgroup%
19 }
```

`\pagecolor` [*model*] {*color*} is ignored. Use `\CSSFilename` instead.

```
20 \newcommand*{\LWR@pagecolor}[2][named]{}
```

`\colorbox` [*model*] {*color*} {*text*} is converted into an HTML hex background color span.

```
21 \NewDocumentCommand{\LWR@colorbox}{0{named} m m}{%
22 \begingroup%
23 \LWR@colorstyle{background:}{#1}{#2}{#3}%
24 \endgroup%
25 }
```

`\fcolorbox` [*framemodel*] {*framecolor*} [*boxmodel*] {*boxcolor*} {*text*} is converted into a framed HTML hex background color span.

A background color of "none" creates a colored frame without a background color.

```
26 \NewDocumentCommand{\LWR@fcolorbox}{O{named} m O{named} m m}{%
27 \begingroup%
28 \ifthenelse{\equal{#4}{none}}{% no background color
29 \LWR@colorstyle{border:1px solid }{#1}{#2}{#5}%
30 }{% yes background color
31 \LWR@colorstyle{border:1px solid }{#1}{#2}%
32 {\LWR@colorstyle{background:}{#3}{#4}{#5}}%
33 }%
34 \endgroup%
35 }
```

Redirect to new definitions:

```
36 \let\textcolor\LWR@textcolor
37 \let\pagecolor\LWR@pagecolor
38 \LetLtxMacro\colorbox\LWR@colorbox
39 \LetLtxMacro\fcolorbox\LWR@fcolorbox
```

Package 118

lwarp-xfrac.sty

189 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}`

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 \DeclareInstance{xfrac}{lmr}{text}{
27 numerator-format = {%
28 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
29 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend},
30 denominator-format = {%
31 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
32 \LWR@htmlsmallfontstart{\, #1\LWR@htmlsmallfontend},

```

For `pdftotext`, do not scale the text:

```

33 scaling = false
34 }
35 \DeclareInstance{xfrac}{lmss}{text}{
36 numerator-format = {%
37 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
38 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend},
39 denominator-format = {%
40 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
41 \LWR@htmlsmallfontstart{\, #1\LWR@htmlsmallfontend},

```

For pdftotext, do not scale the text:

```
42 scaling = false
43 }
44 \DeclareInstance{xfrac}{lmtt}{text}{
45 numerator-format = {%
46 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
47 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend},
48 denominator-format = {%
49 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
50 \LWR@htmlsmallfontstart{\, #1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
51 scaling = false
52 }

53 \end{warpHTML}
```

Package 119

lwarp-xmpincl.sty

190 Xmpincl

Pkg xmpincl Emulated.

for **HTML output**: Discard all options for lwarp-xmpincl:

```
1 \LWR@ProvidesPackageDrop{xmpincl}
```

```
2 \newcommand*{\includemp}[1]{}
```

Change History and Index

Change History

v0.10		Docs: Commands for a successful HTML conversion.	68
General: 2016/03/08 Initial version	1		
v0.11		Docs: Commands into a warpprint environment.	65
General: 2016/03/11	1	Docs: Newclude limitations.	81
Added section:		Docs: Table: Cross-referencing data structures.	271
Operating-System portability.	97	Docs: Table: Float data structures.	281
Added section: Selecting the operating system.	64	Docs: Trademarks section.	357
Test Suite: limages and index in README.txt	1	Docs: Troubleshooting cross-references.	89
Test Suite: MS-Windows in README.txt	1	Test Suite: Assigned cleveref name for Test Float.	1
v0.12		Test Suite: Floatrow	1
\LRW@newhtmlfile: Bugfix: TOC with numbered files.	199		
General: 2016/03/14	1	v0.15	
Global: Uses \p@(type) in float captions.	1	\printpublished: No div if empty.	221
Test Suite: Sub-figures	1	\printsubtitle: No div if empty.	222
v0.13		General: 2016/04/06	1
\CaptionSeparator: Fix for newer babel package.	283	Added	403
\LRW@LwarpStart: \up and \fup	212	Ampersand (&): Fixed handling when passed as an argument.	243
\published: Default to empty published.	218	Docs: Added warning icons for items needing special attention.	94
\subtitle: Default to empty subtitle.	219	Docs: Clarify print/HTML output.	64
General: 2016/03/24	1	Docs: Moved the supported functions table to the introduction.	27
Removed package: subfig	1	Files: lwarp_formal.css added.	1
Test Suite: Ordinals, Subcaption	1	Fix: steps counter	402
tikzpicture: Fix dollar-redefined bug for newer package.	498	Fixed & handling.	400
v0.14		Test Suite: test_suite_formal.css file added.	1
\LRW@htmlsectionfilename: Fix: Links to home page.	172		
General: 2016/03/31	1	v0.16	
floatrow: Added.	398	General: 2016/04/11	1

<code>\tittingpage</code> : Improved	
print-output spacing.	223
Added XeLaTeX, LuaLaTeX	
support.	105
Docs: Font and UTF-8 support.	61
Docs: Moved location of	
<code>\usepackage{lwarp}</code>	62
Docs: Text not converting. . . .	89
Fix: amsmath options clash . .	109
Fix: newtxmath compatibility.	109
Lwarp no longer selects	
fonts.	61, 105
Removed package: suffix	1
Test Suite: Improved	
titlingpage.	223
Test Suite: Lwarp no longer	
selects fonts.	1
Test Suite: Supports XeLaTeX,	
LuaLaTeX.	1
xfrac: Adjusted for the use of	
any font:	524
v0.17	
<code>\LWR@htmlsectionfilename</code> : Fix:	
Links when entire doc is one	
HTML page.	172
General: 2016/04/14	1
mdframed: Added.	437
Test Suite: Fix: Print-version	
front-matter page numbers. . . .	1
Test Suite: Mdfamed	1
v0.18	
<code>\LWR@hspace</code> : <code>\hspace</code> supported.	349
<code>\LWR@includegraphicsb</code> : Add:	
svgz file extension.	325
em, ex, %, px dimensions	
preserved.	325
Fix: <code>\linewidth</code> , <code>\textwidth</code> ,	
<code>\textheight</code> inside a	
minipage.	325
Improved HTML output	
linebreaks.	325
<code>\LWR@myshorttoc</code> : Reorganize	
<code>\HomeHTMLFilename</code> logic. . .	287
<code>\LWR@newhtmlfile</code> : sideTOC after	
title, improving responsive	
design.	198
<code>\LWR@requesttoc</code> : Reorganize	
<code>\HomeHTMLFilename</code> logic. . .	214
<code>\LWR@subhyperref</code> : Improved	
HTML output linebreaks. . . .	279
<code>\LWR@subhyperrefclass</code> : Improved	
HTML output linebreaks. . . .	279
<code>\LWR@subinlineimage</code> : Surpress	
extra space.	280
General: 2016/05/19	1
File: lwarp.css: Improved TOC	
outline display.	1
Files: lwarp.css and	
lwarp_formal.css: Improved	
responsive design.	1
Microtype disabled during HTML	
generation	105
PDF Unicode input characters.	95
Test Suite: Verse package	1
<code>lateximage</code> : pdfcrop: --hires	
added.	316
Reorganize <code>\HomeHTMLFilename</code>	
logic.	316
Surpress extra space.	316
<code>verse</code> : Supports verse, memoir	
packages.	515
<code>minipage</code> : Fix: <code>\linewidth</code> ,	
<code>\textwidth</code> , <code>\textheight</code>	
inside a minipage.	336
v0.19	
<code>\HTMLFilename</code> : Docs: Escape	
filename underscores.	171
<code>\HomeHTMLFilename</code> : Docs: Escape	
filename underscores.	171
<code>\LWR@LwarpStart</code> : Enabled <code>\\</code>	
equal to <code>\newline</code>	211
<code>\LWR@doubledollar</code> : MathJax	
support.	299
<code>\LWR@filestart</code> :	
lwarp_mathjax.txt loaded. . .	209
<code>\LWR@hspace</code> : Fix: <code>\hspace</code> length	
computations.	349
<code>\LWR@minipagestartpars</code> :	
Surpresses paragraph tags	
between minipages.	348
<code>\LWR@singledollar</code> : MathJax	
support.	299
<code>\LWR@tabledatamultirowtag</code> :	
Added optional args.	265
<code>\LateximageFontSizeName</code> : Add:	
User-adjustable	
math/lateximage font size. . .	315
<code>\minipagefullwidth</code> : Added: No	
width tag for the next minipage	
in HTML.	336

<code>\rule</code> : Added	353	<code>\LWR@hspace</code> : Add: Supports HTML thin breakable space.	349
<code>\warpHTMLonly</code> : Added.	98	<code>\LWR@htmldivclass</code> : Added optional style.	181
<code>\warpprintonly</code> : Replaces <code>\rowprintedonly</code> .	98	<code>\LWR@htmlclass</code> : Added optional style.	181
General: 2016/06/08	1	<code>\LWR@htmlsectionfilename</code> : HTMLFilename: removed additional trailing '-', and may be empty.	172
Added.	523	Sections called "Index" or "index" have an underscore prepended to their filenames if no prefix.	172
Avoids MathJax.	298	<code>\LWR@includegraphicsb</code> : Fix: <code>\linewidth</code> in a floatrow.	325
cleveref: Loaded		Fix: Expands filename.	325
<code>\AtEndPreamble</code> .	332	<code>\LWR@longtabledatacaptiontag</code> : Fix: Pars in captions.	261
CSS for table note item.	497	<code>\LWR@section</code> : Combined higher-level sections together into files.	202
Docs: <code>multirow</code> browser bug workaround.	74, 239	<code>\LWR@setOSWindows</code> : Auto-detects operating system.	97
Docs: Math options.	62	<code>\LWR@subhtmlclass</code> : Factored code.	180
Docs: Table: Cross-referencing data structures, updated.	271	<code>\SetHTMLFileNumber</code> : Add: Control file numbers.	171
File: <code>lwarp_mathjax.txt</code> added.	1	<code>\cpagerefFor</code> : User-redefinable word for page references.	333
File: <code>lwarp.css</code> : <code>tnoteitemheader</code> added.	1	<code>\dotfill</code> : Inserts an ellipsis.	346
Introduction: MathJax support mentioned.	26	<code>\hfill</code> : Inserts a <code>\qqquad</code> .	346
MathJax support added.	301, 306, 307	<code>\hrulefill</code> : Inserts a short rule.	346
Options: <code>mathsvg</code> and <code>mathjax</code>	99	<code>\hyperindexref</code> : Print mode provided in case hyperref not used.	295
Supports colored <code>\rule</code> .	520	<code>\pageref</code> : Added.	278
titles: <code>null</code> <code>\pagestyle</code> and <code>\thispagestyle</code> for HTML.	500	<code>\tracinglwarp</code> : Added.	113
<code>tikzpicture</code> : Adapts to <code>tikz</code> version.	498	General: 2017/02/09	1
<code>equation</code> : MathJax support.	304	<code>afterpage</code> : Added.	360
v0.20		<code>alltt</code> : Added.	362
<code>\BlockClassSingle</code> : Renamed from "LWR@htmldivclassline".	183	<code>bookmark</code> : Added.	367
<code>\HTMLDescription</code> : Added <code>\NewHTMLdescription</code> . (Renamed in v0.30.)	192	<code>caption</code> and <code>subcaption</code> supported.	1
<code>\HTMLFilename</code> : No longer escape underscores.	171	cleveref and referencing patches: Applied <code>\AfterEndPreamble</code> .	332
<code>\HomeHTMLFilename</code> : No longer escape underscores.	171	<code>draftwatermark</code> : Added.	373
<code>\InlineClass</code> : Renamed from "inlineclass".	183	<code>eso-pic</code> : Added.	379
<code>\LWR@LwarpStart</code> : Fix: math cross references.	212	<code>everypage</code> : Added.	380
<code>\LWR@closeparagraph</code> : <code>\unskip</code> extra spaces.	186	<code>extramarks</code> : Added.	381
No break tags in the start/end of a tabular.	186	<code>fancyhdr</code> : Added.	385
<code>\LWR@endoffline</code> : Fix: <code>\\</code>	347		
<code>\LWR@filestart</code> : Adds meta description.	209		

hyperref: Additional user macros.	417	Test Suite: HTML meta descriptions.	1
keyfloat: Added.	423	Verbatim: Added.	387
letterspace: User-interface emulated.	425	verbatim: Added.	234
listings: Added.	428	BlockClass: Added optional style.	182
lrcaption: Added.	434	Renamed from "blockclass".	182
lwarp-newproject: Added.	116	LWR@nestspan: Fix: Minipages inside a span.	177
microtype: User-interface emulated.	446	v0.21	
needspace: Added.	453	\LWR@LwarpStart: Changed lateximages to a .txt file.	211
nowidow: Added.	455	\LWR@filestart: Skip title if not given.	209
placeins: Added.	469	\LWR@newhtmlfile: Skip title if not given.	198
ragged2e: Added.	471	\marginpar: Fixed source listing.	195
setspace: Improved support.	475	General: 2017/02/23	1
textpos: Added.	492	fontenc: Added.	405
titles: Added.	500	fontspec: Added.	405
titlesec: Added.	503	inputenc: Added.	422
titletoc: Added.	504	newclude: Added.	453
titling: Improved compatibility.	505	newunicodechar: Added.	454
tocloft: Added.	506	lwarpmk: Fix: lwarpmk again for Windows.	155
wallpaper: Added.	517	lwarpmk: Fix: lwarpmk limages for Windows.	155
wrapfig: Added.	518	lwarpmk: Fix: lwarpmk uses lateximages text file instead of shell script.	155
Added @, <, > columns.	240	Add: Errors for misplaced packages.	101
Added single-expansion data arrays.	170	Docs: Added internet class.	32
Code factored into independent lwarp_html files.	357	Docs: Added TeX2page, GladTeX.	32
Docs: Examples for generating HTML file names.	54	Docs: Installing on Windows.	38
Docs: Improved index.	1	File lwarp_tutorial.txt added.	43
Enhanced titling support.	220	v0.22	
File: lwarp.css: Minor fixes for validation.	1	\LWR@parseDcolumn: Added tabular D column.	249
File: lwarpmk used to compile print, HTML, indexes, and lateximages.	1	\LWR@parsebangcolumn: Added tabular ! column.	247
Fix: \linewidth in a floatrow.	401	\LWR@parsetablecols: Unknown table column types become l. Added tabular D, !, X columns.	250
Improved float caption type handling.	394	\LWR@printmccoldata: Added tabular D, !, and X columns.	259
Moved sidebar and example code to test suite.	1	General: 2017/03/02	1
Page geometry set to 6in wide with large margins.	106	abstract: Added.	358
Parallel versions of aux files for print/HTML.	1	changepage: Added.	369
Removed reliance on make, grep, gawk.	1	dcolumn: Added.	372
Tabular: \unskip extra spaces.	240		

ftnright: Added.	413	emptypage: Added.	374
geometry: Nullified commands.	414	framed: Added.	410
indentfirst: Added.	422	lips: Added.	427
layout: Added.	425	mdframed: Help avoid	
lscap: Added.	434	hyphenation.	438
mcaption: Added.	436	ntheorem: Added.	456
nameref: Added.	452	showidx: Added.	477
nextpage: Added.	454	theorem: Added.	493
parskip: Added.	468	Basic L ^A T _E X theorems: improved	
showkeys: Added.	477	CSS.	234
sidecap: Added.	478	Docs: Adds credits for patched	
tabularx: Added.	490	code.	1
varioref: Supported.	70	Docs: Testing <code>lwarp</code>	88
verse: Added.	515	Fix: Allows XeL ^A T _E X and	
v0.23		LuaL ^A T _E X to preload graphics	
\LWR@parsetablecols: Fix for vert		and graphicx.	102
bar column type.	250	v0.26	
\LWR@printmccoldata: Fix for vert		General: 2017/03/31	1
bar column type.	259	cutwin: Added.	371
General: 2017/03/02	1	endnotes: Added.	375
v0.24		floatflt: Added.	396
\LWR@hspace: Add: <code>\hspace \fill</code>		footmisc: Added.	406
converts to 2em	349	footnotehyper: Added.	409
\LWR@htmlfileref: Fix: Index		footnote: Added.	408
links while <code>\tracinglwarp</code>	274	marginfix: Added.	435
\hypertocfloat: List of floats		marginnote: Added.	436
responds to <code>lofdepth</code> ,		mparhack: Added.	449
<code>lotdepth</code>	293	pagenote: Supported as-is.	467
General: 2017/03/15	1	sidenotes: Added.	479
floatrow: Support for <code>subfig</code>	398	<code>lwarp.css</code> : Improved responsive	
subfig: Added.	483	<code>marginpar</code> and <code>marginblock</code>	118
tikz: For tikz v3.0.0 or later,		Docs: Improved MiK _T E _X install	
auto-loads tikz babel library if		instructions.	35, 38
necessary.	498	Dollar span avoided in a	
Docs: Filename underscore.	52, 69	<code>lateximage</code>	298
No longer preloads <code>subcaption</code> ;		Footnotes now are L ^A T _E X boxes	
conflicted with <code>subfig</code>	108	instead of pagenotes.	193
picture: Fix for inline images.	333	<code>lateximage</code> : Labels track page	
tikzpicture: Fix for inline		numbers of <code>lateximages</code>	316
images.	498	Print mode now uses a <code>minipage</code>	
v0.25		of <code>\linewidth</code>	316
\LWR@loadnever: Added the ability		picture: Fix for <code>\makebox in</code>	
to prevent conflicting		picture.	333
packages.	102	v0.27	
\addcontentsline: Handles		General: 2017/04/04	1
theorems.	286	lettrine: Added.	426
General: 2016/03/22	1	microtype: Fix with XeL ^A T _E X,	
amsthm: Added.	363	LuaL ^A T _E X.	446
ccaption: Prevented.	369	soul: Added.	481
ellipsis: Added.	373	ulem: Added.	513

Docs: Installing utilities for MacOS.	40	Docs: Modfying lwarpmk and index processing.	88
Docs: Limitations of saveboxes.	75	File <code>lwarp_mathjax.txt</code> : Updated CDN repository.	153
Fix for table footnote par tags.	194	Forced oneside to maintain large right margin.	106
Page geometry modified to reduce line overflow.	106		
v0.28		v0.29	
<code>\@wrindex</code> : Improved indexing.	295	<code>\LWR@includegraphicsb</code> : Fix: Error when no optional arguments.	325
<code>\HTMLAuthor</code> : Added <code>\HTMLauthor</code> . (Renamed in v0.30.)	192	General: 2017/04/15	1
<code>\LWR@LwarpEnd</code> : If <code>FormatEPUB</code> or <code>FormatWordProcessor</code> , no bottom nav.	214	<code>lwarpmk</code> : Add: <code>language</code> option for config files.	155
<code>\LWR@LwarpStart</code> : <code>FormatWordProcessor</code> forces single-file output.	211	Add: <code>lwarpmklang</code> option for <code>lwarp</code>	99
<code>\LWR@filestart</code> : Adds HTML meta author.	209	Docs: Using a glossary	59
<code>\LWR@forcenewpage</code> : Forces new PDF page before major environments.	175	File <code>*.lwarpmkconf</code> : Add: <code>language</code> option for config files.	117
<code>\LWR@htmlcomment</code> : Breaks ligatures in HTML comments.	180	File <code>lwarpmk.conf</code> : Add: <code>language</code> option for config files.	116
<code>\LWR@includegraphicsb</code> : Adapts to graphics syntax.	325	v0.30	
<code>\LWR@newhtmlfile</code> : If <code>FormatEPUB</code> or <code>FormatWordProcessor</code> : skips headers, footers, nav.	198	<code>\CSSFilename</code> : Renamed from <code>\NewCSS</code>	191
<code>\LWR@parsetablecols</code> : Added L, C, R, J column types.	250	<code>\HTMLAuthor</code> : Renamed from <code>\HTMLauthor</code>	192
<code>\LWR@startref</code> : Removed space.	276	<code>\HTMLDescription</code> : Renamed from <code>\NewHTMLdescription</code>	192
<code>\chapter</code> : If EPUB, prints footnotes before each section.	207	<code>\HTMLFirstPageTop</code> : Renamed from <code>\SetFirstPageTop</code>	190
<code>\hyperindexref</code> : Improved indexing.	295	<code>\HTMLLanguage</code> : Renamed from <code>\MetaLanguage</code>	208
<code>\textup</code> : Fixed span class.	342	<code>\HTMLPageBottom</code> : Renamed from <code>\SetPageBottom</code>	191
General: 2017/04/14	1	<code>\HTMLPageTop</code> : Renamed from <code>\SetPageTop</code>	191
glossaries: Added.	415	General: 2017/04/29	1
graphics: Added.	416	<code>lwarp-newproject</code> removed, and combined with <code>lwarp</code>	116
<code>tabularx</code> : Fix for optional pos.	490	<code>lwarpmk</code> : Add: <code>xdyfile</code> configuration option.	155
<code>tabulary</code> : Added.	491	<code>lwarpmk</code> : Fix: <code>xindy</code> and <code>texindy</code> adjusted for <code>pdflatex</code> , <code>xelatex</code> and <code>lualatex</code>	155
<code>lwarpmk</code> : Add: <code>printglossary</code> and <code>htmlglossary</code> commands.	155	<code>lwarpmk</code> : Fix: <code>xindy</code> now used for print index generation with <code>latexmk</code>	155
Added boolean <code>FormatEPUB</code>	113		
Added boolean <code>FormatWordProcessor</code>	114		
Added boolean <code>HTMLDebugComments</code>	113		
Added boolean <code>HTMLMarkFloats</code>	114		

lwarpmk: language now used for both index and glossary generation.	155		
File: lwrap_html.xdy renamed to lwrap.xdy.	153		
Fix: *.css files only written in print mode.	118		
Fix: lwrap.xdy only written in print mode.	153		
Fix: lwrap_mathjax.txt only written in print mode.	153		
Option OSWindows replaces macro \warpOSWindows.	100		
Option latexmk replaces macro \UseLatexmk.	100		
Option lwrapmklang changed to IndexLanguage.	99		
Option xdyFilename added.	100		
Options HomeHTMLFilename and HTMLFilename replace macros \HomeHTMLFilename and \HTMLFilename.	100		
v0.31			
General: 2017/05/15	1		
keyfloat: Improved compatibility.	423		
v0.32			
\RequirePackage: Fix: Ignores blanks in package list.	110		
General: 2016/06/09	1		
glossaries: Prevent error with \glo@name not defined.	296		
lwarpmk: Fix: io.lines() changed to file:lines() due to luatex changes.	155		
v0.33			
\HTMLAuthor: Fix: Provides empty default author if none given.	192		
\LWR@loadbefore: Fix: No \PackageError if already loaded.	102		
\LWR@parseatcolumn: Fix: Column alignment with leftmost @.	246		
\LWR@tabledatasinglecolumn: Fix: Macros in tabular could cause extra data cell.	255		
\LWR@vspace: Add: \vspace nullified.	350		
		\StartDefiningTabulars: Add:	
		Avoids error: misplaced alignment tab character &.	243
		General: 2017/07/10	1
		amsmath: Removed fleqn option.	109
		fancyhdr: Fix: Optional args for \thead, etc.	385
		Add: Tabular at and bang columns now have their own HTML columns.	240
		cleveref: Fix: Loaded \AtEndPreamble.	332
		Fix: Incorrectly-inline math environments.	307
		New handling of & to localize catcode changes.	240
	v0.34		
		\@fnsymbol: Text symbols instead of math.	227
		\InlineClass: Moved optional argument in front of mandatory.	183
		\LWR@htmldivclass: Moved optional argument in front of mandatory.	181
		\LWR@htmlclass: Moved optional argument in front of mandatory.	181
		\LWR@htmlclassline: Moved optional argument in front of mandatory.	182
		\LWR@htmlspanclass: Moved optional argument in front of mandatory.	179
		\LWR@nullfonts: Improved font control.	343
		\LWR@restoreorigformatting: booktabs: Works inside lateximage.	302
		Improved font control.	302
		\LWR@subhtmlclass: Moved optional argument in front of mandatory.	180
		\LWR@tabledatacolumn: booktabs: Works inside lateximage.	266
		\fbboxBlock: Added.	339
		\makebox: Fix: Handles paren arg.	339
		General: 2017/08/08	1

babel-french: Adds fixed-width HTML spaces to punctuation.	176	shadow: Added.	476
balance: Added.	366	syntonly: Added.	489
booktabs: Works inside lateximage.	268, 367	titles: No longer required.	500
boxedminipage2e: Added.	368	titleref: Prevented.	502
boxedminipage: Prevented.	368	xmpincl: Added.	526
crop: Added.	370	Docs: Horizontal space limitations.	1
enumerate: Added.	376	Docs: Misplaced alignment character.	89
enumitem: Added, no longer required.	377	File: README.txt: updated.	1
everyshi: Added.	381	File: lwrap_mathjax.txt version change.	153
fancybox: Added.	383	Fix: Added the eqnarray environments.	307
fancyvrb: Added, no longer required.	387	Improved font control.	341
figcaps: Added.	393	Lists refactored to remove enumitem requirement.	235
filecontents: Required. Patched for morewrites.	107	Verbatim refactored to remove fancyvrb requirement.	232
floatpag: Added.	397	lateximage: Fix: lateximage with minipage, \parbox, \makebox, \fbox, \framebox, \raisebox, \scalebox, \reflectbox.	316
flushend: Added.	404	BlockClass: Moved optional argument in front of mandatory.	182
fullpage: Added.	413	fminipage: Added.	340
hyperxmp: Added.	420	LWR@nestspan: Fix: Minipages, BlocksClass, and lists inside a span.	177
idlayout: Added.	421	LWR@tabular: booktabs: Works inside lateximage.	268
marginfit: Added.	435		
mdframed: Improved mdtheorem patch.	443		
moreverb: Added.	447		
paralist: Added.	467		
pdfscape: Added.	468		
pdfsync: Added.	469		
prelim2e: Added.	470		
rotfloat: Added.	473		
savetrees: Added.	474		

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols			
\\$	<i>298</i>	\@@setcref	6587
\&	<i>171, 4641</i>	\@@setcrefrange	<u>6588</u>
\(.	<u>5759</u>	\@author	<i>217</i>
\)	<u>5759</u>	\@begintheorem	<u>4487</u>
\,	<i>69</i>	\@capytype	<u>5449</u>
\@@setcpageref	<u>6591</u>	\@currentlabelname	<u>5230</u>
		\@date	<i>217</i>

C	
calc (package)	108
Calibre	84
caption (package)	108
\caption@begin	5482
\caption@end	5482
\captionlistentry	5486
\captionof	5516
\CaptionSeparator	5450
ccaption (package)	369
center (environment)	6300
\centering	212
changepage (package)	369
\chapter	3846
\citetitle	4413
classes:	
internet	32
cleveref (package)	332
cmap (package)	62
CombineHigherDepths (boolean)	53, 196
comment (package)	96
Computer Modern	61
\ConTeXt	7053
counters:	
FileDepth	53, 195
lofdepth	292
lotdepth	292
LWR@externalfilecnt	298
LWR@htmlfilenumber	172
LWR@latestautopage	284
LWR@lateximagedepth	314
LWR@lateximagenumber	314
LWR@LIpage	315
LWR@midrulecounter	256
LWR@minipagedepth	334
LWR@nextautofloat	284
LWR@nextautopage	284
LWR@nextequation	301
LWR@prevFileDepth	202
LWR@spandepth	184
LWR@tablecolspos	245
LWR@tablecolswidth	245
LWR@tabletotalcols	246
LWR@tabletotalcolsnext	246
LWR@thisfloat	283
SideTOCDepth	53, 290
tocdepth	52
\cpagerefFor	6590
crop (package)	370
cross-references	
missing or incorrect	89
CSS	
file selection	68
lwarp.css	67
per HTML page	68
project-specific changes	67
\CSSFilename	51, 54, 68, 3492
cutwin (package)	371
D	
danger icon	94
\date	66
dcolumn (package)	372
debugging	89
HTML debug comments	113
\DeclareGraphicsExtensions	6330
\DeclareGraphicsRule	6330
Deja Vu	61
description	
HTML meta tag	66, 192
description (environment)	4551
displaymath (environment)	5777
\dotfill	6884
draftwatermark (package)	373
E	
ellipsis (package)	373
\emph	6778
emptypage (package)	374
\end@dlbfloat	5441
\end@float	5441
\EndDefiningTabulars	4636
endnotes (package)	375
\enlargethispage	6991
\enskip	348, 6952
enumerate (environment)	4523
enumerate (package)	376
enumitem (package)	377
environ (package)	108
environments:	
abstract	4396
align	6072
align*	6095
BlockClass	3295
BVerbatim	181
center	6300
description	4551
displaymath	5777
enumerate	4523

selection	61	between minipages	348
size — lateximage	71, 296, 314	<code>\href</code>	5372
size — math, SVG	71, 296, 314	<code>\hrulefill</code>	6883
size — xfrac	523	<code>\hskip</code>	69
<code>fontenc</code> (package)	61, 405	<code>\hspace</code>	69, 348, 6982
<code>fontspec</code> (package)	61, 405	HTML	
<code>footmisc</code> (package)	406	commands for a successful conver-	
<code>footnote</code> (package)	408	sion	68
<code>footnotehyper</code> (package)	409	conversion settings	52
footnotes	193	debug comments	113
for		EPUB	84, 113
HTML & PRINT	94	word processor	86, 114
HTML output	94	filename generation	54
PRINT output	94	headings	92
<code>FormatEpub</code> (boolean)	84, 113	in place of SVG images	49, 90
<code>FormatWordProcessor</code> (boolean)	86, 114	meta tag	
<code>\framebox</code>	6721	author	67, 192
<code>framed</code> (package)	410	description	66, 192
framed objects	76	generator	209
<code>FrameMaker</code> (program)	34	viewport	210
Frequently Asked Questions	89	tabular column conversion	251
<code>\frontmatter</code>	3718	<code>\HTMLAuthor</code>	55, 67, 3506
<code>ftnright</code> (package)	413	<code>HTMLDebugComments</code> (boolean)	113
<code>fullpage</code> (package)	413	<code>\HTMLDescription</code>	55, 66, 3512
<code>\fup</code>	6881	<code>\HTMLentity</code>	2990
		<code>\HTMLFilename</code>	3008
		<code>HTMLFilename</code> (option)	52, 100
		<code>\HTMLFirstPageTop</code>	55, 3469
		<code>\HTMLLanguage</code>	55, 3904
		<code>HTMLleftmargini</code> (length)	
			81, 231, 232, 515
		<code>HTMLMarkFloats</code> (boolean)	86, 114
		<code>\HTMLPageBottom</code>	55, 217, 3485
		<code>\HTMLPageTop</code>	55, 3477
		<code>\HTMLunicode</code>	2998
		<code>HTMLvleftskip</code> (length)	81, 231, 515
		<code>\hyperindexref</code>	5707
		<code>hyperref</code> (package)	278, 417
		<code>\hypertoc</code>	5618
		<code>\hypertocfloat</code>	5631
		<code>hyperxmp</code> (package)	420
		I	
		icon	
		warning	94
		<code>idxlayout</code> (package)	421
		<code>ifplatform</code> (package)	96
		images	
		graphicx package	321
		in strange places	90
		H	
		<code>Hevea</code> (program)	32
		<code>\hfill</code>	6882
		<code>\HomeHTMLFilename</code>	3009
		<code>HomeHTMLFilename</code> (option)	52, 100
		horizontal space	69
		G	
		<code>gather</code> (environment)	6026
		<code>gather*</code> (environment)	6049
		<code>GELLMU</code> (program)	32
		generator	
		HTML meta tag	209
		<code>geometry</code> (package)	106, 414
		<code>getttitlestring</code> (package)	107
		<code>GladTeX</code> (program)	33
		<code>glossaries</code> (package)	82, 415
		glossary	
		language	59
		processing	59
		<code>glyphtounicode</code> (file)	61
		<code>graphics</code> (package)	321, 416
		<code>\graphicspath</code>	6328
		<code>graphicx</code> (package)	321, 416

<code>\LWR@afterendverbatim</code>	4461	<code>\LWR@htmlcomment</code>	3230
<code>\LWR@amsmultiline (boolean)</code>	306	<code>\LWR@htmldivclass</code>	3272
<code>\LWR@atbeginverbatim</code>	4448	<code>\LWR@htmldivclassend</code>	3275
<code>\LWR@beginhideamsmath</code>	5915	<code>\LWR@htmlmlelement</code>	3285
<code>\LWR@botnavigation</code>	3067	<code>\LWR@htmlmlelementclass</code>	3259
<code>\LWR@caption@begin</code>	5463	<code>\LWR@htmlmlelementclassend</code>	3264
<code>\LWR@caption@end</code>	5473	<code>\LWR@htmlmlelementclassline</code>	3278
<code>\LWR@clearmidrules</code>	4892	<code>\LWR@htmlmlelementend</code>	3288
<code>\LWR@closeparagraph</code>	3382	<code>\LWR@htmlfilenumber (counter)</code>	172
<code>\LWR@closeprevious</code>	3087	<code>\LWR@htmlfileref</code>	5256
<code>\LWR@closetabledatacell</code>	4600	<code>\LWR@htmlmathlabel</code>	5901
<code>LWR@colafterspec</code>	246	<code>\LWR@htmlopencomment</code>	3209
<code>LWR@colatspec</code>	246	<code>\LWR@htmlrefsectionfilename</code>	3052
<code>LWR@colbangspec</code>	246	<code>\LWR@HTMLSanitize</code>	6179
<code>LWR@colbeforespec</code>	246	<code>\LWR@HTMLSanitizeexpand</code>	6197
<code>\LWR@copyfile</code>	383	<code>\LWR@htmlsectionfilename</code>	3019
<code>\LWR@createautosec</code>	3727	<code>\LWR@htmlspan</code>	3184
<code>\LWR@currentcss</code>	3491	<code>\LWR@htmlspanclass</code>	3192
<code>\LWR@descitem</code>	4537	<code>\LWR@htmltag</code>	3200
<code>\LWR@docmidrule</code>	4911	<code>\LWR@htmltagc</code>	3130
<code>LWR@doingapar (boolean)</code>	185	<code>\LWR@includegraphicsb</code>	6421
<code>LWR@doinghline (boolean)</code>	241	<code>\LWR@indexitem</code>	5681, 5685, 5689
<code>LWR@doingstartpars (boolean)</code>	184	<code>\LWR@indexsection</code>	5665
<code>LWR@doingtbrule (boolean)</code>	241	<code>\LWR@infloatrow (boolean)</code>	325
<code>\LWR@domulticolumn</code>	4985	<code>\LWR@intabularmetadata (boolean)</code>	241
<code>\LWR@doubledollar</code>	5733	<code>\LWR@itemizeitem</code>	4502
<code>\LWR@endhideamsmath</code>	5923	<code>\LWR@latestautopage (counter)</code>	284
<code>\LWR@endminipage</code>	6617	<code>\LWR@lateximagedepth (counter)</code>	314
<code>\LWR@endoffline</code>	6929	<code>\LWR@lateximagedepthref</code>	5260
<code>\LWR@endsubminipage</code>	6638	<code>\LWR@lateximagenumber (counter)</code>	314
<code>\LWR@ensuredoingapar</code>	3356	<code>\LWR@lateximagenumberref</code>	5263
<code>LWR@externalfilecnt (counter)</code>	298	<code>\LWR@LIpage (counter)</code>	315
<code>\LWR@FBcancel</code>	3121	<code>\LWR@loadafter</code>	98
<code>\LWR@filenamoblanks</code>	3584	<code>\LWR@loadbefore</code>	109
<code>\LWR@filestart</code>	3911	<code>\LWR@loadnever</code>	118
<code>\LWR@findword</code>	318	<code>\LWR@longtabledatacaptiontag</code>	5004
<code>\LWR@floatbegin</code>	5411	<code>\LWR@lookforpackagename</code>	321
<code>\LWR@floatend</code>	5429	<code>\LWR@LwarpEnd</code>	4038, 7083
<code>LWR@fminipageprint (environment)</code>	6753	<code>\LWR@LwarpStart</code>	3961, 7083
<code>\LWR@forcenewpage</code>	3093	<code>\LWR@maketitlesetup</code>	4346
<code>LWR@freezethisfloat (boolean)</code>	283	<code>\LWR@maybeinthisfloat</code>	5446
<code>\LWR@futurenonspacelct</code>	4577	<code>\LWR@maybeinthisfloatrow</code>	4825
<code>\LWR@FVstyle</code>	17	<code>\LWR@midrulecounter (counter)</code>	256
<code>\LWR@getexparray</code>	2986	<code>\LWR@midrules</code>	256
<code>\LWR@getmynexttoken</code>	4584	<code>\LWR@minipagedepth (counter)</code>	334
<code>\LWR@hidelatexequation</code>	5856	<code>\LWR@minipagefullwidth (boolean)</code>	336
<code>\LWR@hspace</code>	6960	<code>\LWR@minipageheight (length)</code>	335
<code>\LWR@htmlblockcomment</code>	3237	<code>\LWR@minipagestartpars</code>	6934
<code>\LWR@htmlblocktag</code>	3245	<code>\LWR@minipagestoppars</code>	6940
<code>\LWR@htmlclosecomment</code>	3209	<code>\LWR@minipagethispar (boolean)</code>	336

LWR@minipagewidth (length)	335	\LWR@sidetoc	5597
\LWR@multicolother	4936	\LWR@singledollar	5745
\LWR@multicolpartext	4931	LWR@skipatbang (boolean)	241
\LWR@multicolskip	4942	LWR@skippingmrowcell (boolean) . .	241
\LWR@mynexttoken	4576	LWR@spandepth (counter)	184
\LWR@myshorttoc	5528	\LWR@splabel	5266
\LWR@nameref	5253	LWR@starredlongtable (boolean) . .	261
LWR@nestspan (environment)	3140	LWR@startedrow (boolean)	240
\LWR@newtmlfile	3650	\LWR@startnewdepth	3731
\LWR@newlabel	5290	\LWR@startpars	3420
\LWR@newref	5339	\LWR@startref	5303
LWR@nextautofloat (counter)	284	\LWR@stoppars	3436
LWR@nextautopage (counter)	284	\LWR@stripperperiod	5231
LWR@nextequation (counter)	301	\LWR@strresult	4669
\LWR@nohspace	6981	\LWR@subcmidrule	4903
\LWR@nullfonts	6831	\LWR@subhtmelementclass	3250
\LWR@openparagraph	3361	\LWR@subhyperref	5363
\LWR@origcolspec	4671	\LWR@subhyperrefclass	5367
\LWR@originname	6525	\LWR@subinlineimage	5404
\LWR@originnames	6532	\LWR@subminipage	6632
\LWR@parseaftercolumn	4731	\LWR@subnewref	5333
\LWR@parseatcolumn	4677	\LWR@subsublabel	5269
\LWR@parsebangcolumn	4701	\LWR@subtableofcontents	5543
\LWR@parsebeforecolumn	4722	\LWR@syncmathjax	5779
\LWR@parseDcolumn	4759	\LWR@tablecolspec	4668
\LWR@parsenormalcolumn	4744	LWR@tablecolspos (counter)	245
\LWR@parsepcolumn	4755	LWR@tablecolswidth (counter) . . .	245
\LWR@parseskipcolumn	4741	\LWR@tabledatacolumnntag	5111
\LWR@parsetablecols	4764	\LWR@tabledatamulticolumnntag . .	5089
\LWR@patchlists	4565	\LWR@tabledatamultirowntag	5097
LWR@prevFileDepth (counter)	202	\LWR@tabledatasinglecolumnntag . .	4855
\LWR@printatbang	4841	LWR@tableparcell (boolean)	241
\LWR@printmccoldata	4945	LWR@tabletotalcols (counter) . . .	246
\LWR@printmccoltype	4915	LWR@tabletotalcolsnxt (counter) . .	246
\LWR@printthetitle	4197	LWR@tabular (environment)	5168
\LWR@ProvidesPackageDrop	373	\LWR@tabularendofline	4652
\LWR@ProvidesPackagePass	361	\LWR@tabularhtmlcolumns	5070
\LWR@pushoneclose	3730	LWR@tempheight (length)	349
\LWR@requesttoc	4031	LWR@tempraise (length)	349
\LWR@requirepackagenames	317	LWR@tempwidth (length)	349
\LWR@restoreorigformatting	5799	\LWR@thisfilename	3582
\LWR@rotboxorigin	6524	LWR@thisfloat (counter)	283
\LWR@rotstyle	6411	\LWR@thisnewfilename	3583
\LWR@rule	6992	\LWR@titlingmaketitle	4384
\LWR@scalestyle	6414	\LWR@topnavigation	3064
\LWR@section	3737	\LWR@traceinfo	403
\LWR@sectionnumber	3724	LWR@tracinglwrap (boolean)	113
\LWR@setexparray	2981	LWR@validtablecol (boolean)	250
\LWR@setlatestname	5232	LWR@verbtags (boolean)	232
\LWR@setOSWindows	31	\LWR@vspace	6983

<code>\LWRsetnextfloat</code>	5454	<code>microtype</code> (package)	62, 105, 446
<code>\LyX</code>	7064	<code>\MiKTeX</code>	7063
M			
Mac OS (program)	64, 97	<code>minipage</code>	
Madcap (program)	34	framed	76
<code>\mainmatter</code>	3714	horizontal space between	348
<code>\makebox</code>	6716	<code>minipage</code> (environment)	6653
<code>makeidx</code> (package)	108	<code>\minipagefullwidth</code>	6645
<code>\MakeIndex</code>	7056	misplaced alignment tab character &	73, 238, 243
<code>\maketitle</code>	66, 224, 4359	missing sections	53
<code>\maketitlehookaa</code>	216	<code>moreverb</code> (package)	447
<code>\maketitlehookaaa</code>	216	<code>mparhack</code> (package)	449
margin		<code>\mrowcell</code>	5157
numbers	94	MS-Windows (program)	64, 97
tags	94	<code>multicol</code> (package)	450
<code>marginfit</code> (package)	435	<code>multirrow</code> (package)	265, 452
<code>marginfix</code> (package)	435	<code>multline</code> (environment)	5977
<code>marginnote</code> (package)	436	<code>multline*</code> (environment)	6001
<code>\marginpar</code>	3571	N	
markup languages	34	<code>\Nameref</code>	5360
math		<code>\nameref</code>	5351
font size — SVG	71, 296, 314	<code>nameref</code> (package)	452
mathjax option	99	<code>needspace</code> (package)	453
MathJax summary	71, 297	<code>newclude</code> (package)	81, 453
mathsvg option	99	<code>newfloat</code>	
showing as HTML	49, 90	with trivfloat, algorithmic	511
SVG summary	70, 296	<code>\newfloat</code>	2, 4
<code>math</code> (environment)	5776	<code>newfloat</code> (package)	108
MathJax		<code>\newline</code>	6927
mathjax option	99	<code>\newtheorem</code>	234
subequations	72, 297	<code>newtxmath</code> (package)	82
summary	71, 297	<code>newunicodechar</code> (package)	61, 454
tagged equations	71, 297	<code>nextpage</code> (package)	454
MathJax (program)	71, 297	<code>nicefrac</code> (package)	81
MathJax (program) [requirement] ...	36	<code>\nolinebreak</code>	6986
<code>mathjax</code> (boolean)	98	<code>\nolinkurl</code>	5384
<code>mathjax</code> (option)	52, 62, 99	<code>\nopagebreak</code>	6990
<code>mathsvg</code> (option)	52, 62, 99	<code>nowidow</code> (package)	455
<code>mcaption</code> (package)	436	<code>ntheorem</code> (package)	456
<code>mdframed</code> (package)	437	<code>\numberline</code>	5615
<code>\mdseries</code>	6867	numbers	
memoir		left margin	94
verse	516	O	
<code>memoir</code> (package)	516	<code>OpenOffice</code> (program)	33
meta tag, HTML		options:	
author	67, 192	BaseJobname	99
description	66, 192	HomeHTMLFilename	52, 100
generator	209	HTMLFilename	52, 100
viewport	210		

IndexLanguage	59, 83, 99, 415
latexmk	52, 100
lwarpmk	100
mathjax	52, 62, 99
mathsvg	52, 62, 99
OSWindows	64, 97, 100
warpHTML	62, 99
warpprint	62, 98
xdyFilename	68, 100
\OSPathSymbol	30
OSWindows (option)	64, 97, 100

P

packages	
required	105
packages:	
abstract	358
afterpage	360
algorithmicx	361
alltt	362
amsmath	109
amsthm	363
array	271
babel	82
balance	366
bookmark	367
booktabs	367
boxedminipage	368
boxedminipage2e	368
calc	108
caption	108
ccaption	369
changepage	369
cleveref	332
cmap	62
comment	96
crop	370
cutwin	371
dcolumn	372
draftwatermark	373
ellipsis	373
emptypage	374
endnotes	375
enumerate	376
enumitem	377
environ	108
epigraph	378
eso-pic	379
etoolbox	96
everyhook	107
everypage	380
everyshi	381
expl3	107
extramarks	381
fancybox	382
fancyhdr	385
fancyvrb	387
figcaps	393
filecontents	107
float	394
floatflt	396
floatpag	397
floatrow	398
flushend	404
fontenc	61, 405
fontspec	61, 405
footmisc	406
footnote	408
footnotehyper	409
framed	410
ftnright	413
fullpage	413
geometry	106, 414
getttitlestring	107
glossaries	82, 415
graphics	321, 416
graphicx	321, 416
hyperref	278, 417
hyperxmp	420
idxlayout	421
ifplatform	96
indentfirst	422
inputenc	61, 422
keyfloat	423
kvoptions	98
layout	425
letltxmacro	96
letterspace	425
lettrine	426
lips	427
listings	428
lmodern	61
longtable	432
lscape	434
ltxcaption	434
lwrap	62
makeidx	108
marginfit	435
marginfix	435
marginnote	436

mcaption	436	titletoc	504
mdframed	437	titling	108, 505
memoir	516	tocloft	506
microtype	62, 105, 446	trivfloat	511
moreverb	447	ulem	513
mparhack	449	units	81
multicol	450	verse	515, 516
multirow	265, 452	wallpaper	517
nameref	452	wrapfig	518
needspace	453	xcolor	520
newclude	81, 453	xfrac	109, 523
newfloat	108	xifthen	107
newtxmath	82	xmpinl	526
newunicodechar	61, 454	xparse	106
nextpage	454	xstring	108
nicefrac	81	zref	109
nowidow	455	page	
ntheorem	456	inaccessible	53
pagenote	467	\pagebreak	6987
paralist	467	pagenote (package)	467
parskip	468	\pagenumbering	3098
pdfscape	468	\pageref	5346
pdfsync	469	\pagerefPageFor	5345
placeins	469	\pagestyle	3096
prelim2e	470	Pandoc (program)	33
ragged2e	471	\paragraph	3886
refcount	108	paralist (package)	467
rotating	472	\parbox	6709
rotfloat	473	\parsemulticolumnalignment	4972
savetrees	474	parskip (package)	468
setspace	475	\part	3834
shadow	476	pdfcrop (program) [requirement]	36
showidx	477	pdfLaTeX (program) [requirement]	36
showkeys	477	pdfscape (package)	468
sidecap	478	pdfseparate (program) [requirement]	36, 41
sidenotes	479	pdfsync (package)	469
siunitx	81, 320	pdftocairo (program) [requirement]	36, 41
soul	481	pdftotext (program) [requirement]	36, 41
subfig	483	Perl	41
syntonly	489	perl (program) [requirement]	41
tabularx	490	\phantomsection	7033
tabulary	491	picture (environment)	333, 6608
textcomp	62	placeins (package)	469
textpos	492	Plastex (program)	32
theorem	493	\popclose	2954
threeparttable	497	Poppler	36, 41
tikz	498	\postpublished	216
titles	500	\postsubtitle	216
titleref	502		
titlesec	503		

