

lwarf

LATEXHTML5

The lwarf package

LATEX to HTML5

v0.29 — 2017/04/15

© 2016–2017 Brian Dunn
bd@BDTechConcepts.com

Abstract

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

Documents may be produced by LATEX, LuaLATEX, or XeLATEX. 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 **lwarf** version of a package for HTML when available. Several dozen LATEX packages are supported with these high-level source compatibility replacements.

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 **lwarf**, 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	l warp.sty	20
1	Updates	20
2	Introduction	24
2.1	Supported packages and features	25
3	Alternatives	30
3.1	Internet class	30
3.2	TeX4ht	30
3.3	Translators	30
3.4	AsciiDoc	31
3.5	Pandoc	31
3.6	Word Processors	31
3.7	Commercial Systems	32
3.8	Comparisons	32
4	Installation	33
4.1	Installing the <code>l warp</code> package	33
4.2	Installing the <code>l warpmk</code> utility	36
4.2.1	Using a local copy of <code>l warpmk</code>	37
4.3	Installing additional utilities	38
5	Tutorial	41
5.1	Starting a new project	41
5.2	Compiling the print version with <code>l warpmk</code>	45
5.3	Compiling the HTML version with <code>l warpmk</code>	46
5.4	Generating the SVG images	47
5.5	Using MathJax for math	48
5.6	Changing the CSS style	49
5.7	Customizing the HTML output	50
5.8	Using <code>latexmk</code>	54
5.9	Using XeLaTeX or LuaLaTeX	55
5.10	Using a glossary	56
5.11	Cleaning auxiliary files	57
5.12	Cleaning auxiliary and output files	57
5.13	Processing multiple projects in the same directory	57
5.14	Using the <code>make</code> utility	57
6	Additional details	58
6.1	Font and UTF-8 support	58
6.2	<code>l warp</code> package loading and options	59
6.3	Selecting the operating system	59
6.4	Selecting actions for print or HTML output	59

6.5	Commands to be placed into the <code>\warpprint</code> environment	61
6.6	Commands for a successful HTML conversion	61
6.7	Title page	62
6.8	HTML page meta descriptions	63
6.9	HTML page meta author	64
6.10	CSS	64
6.11	Special cases and limitations	65
6.11.1	Text formatting	65
6.11.2	Footnotes and page notes	65
6.11.3	Math	65
6.11.4	Ntheorem	67
6.11.5	Graphics	67
6.11.6	Xcolor	68
6.11.7	Tabular	68
6.11.8	Longtable	69
6.11.9	Save Boxes	69
6.11.10	Minipages	70
6.11.11	Mdframed	70
6.11.12	<code>float</code> , <code>trivfloat</code> , and/or <code>algorithmicx</code> together	71
6.11.13	<code>caption</code> and <code>subcaption</code> packages	71
6.11.14	<code>subfig</code> package	71
6.11.15	<code>floatrow</code> package	71
6.11.16	<code>abstract</code> package	72
6.11.17	<code>cleveref</code> and <code>varioref</code> packages	72
6.11.18	<code>verse</code> and <code>memoir</code>	72
6.11.19	<code>siunitx</code> package	73
6.11.20	<code>newclude</code> package	73
6.11.21	<code>newtxmath</code> package	73
6.11.22	<code>babel</code> package	74
6.11.23	<code>glossaries</code> package	74
6.11.24	<code>enumitem</code> package	74
6.11.25	<code>enumerate</code> package	74
7	EPUB conversion	76
8	Word-processor conversion	78
9	Modifying l warp	79
9.1	Creating an <code>l warp</code> version of a package	79
9.2	Testing <code>l warp</code>	80
9.3	Modifying <code>l warpmk</code>	80
9.4	Modifying <code>xindy</code> index processing	81
10	Troubleshooting	82
10.1	Using the <code>l warp.sty</code> package	82
10.1.1	Debug tracing output	83

10.2	Compiling the <code>l warp.dtx</code> file	84
11	Implementation	84
12	Stack depths	85
13	Source Code	86
14	Detecting the <code>T<small>E</small>X</code> Engine — <code>pdflatex</code> , <code>lualatex</code> , <code>xelatex</code>	87
15	Unicode Input Characters	87
16	Handling package options	87
17	Misplaced packages	90
18	Required packages	93
19	Loading packages	98
20	Copying a file	101
21	Debugging messages	101
22	HTML-conversion output modifications	102
23	Remembering original formatting macros	103
24	Operating-System portability	104
24.1	Common portability code	104
24.2	Unix, Linux, and Mac OS	104
24.3	MS-Windows	104
25	Stacks	105
25.1	Assigning depths	105
25.2	Closing actions	106
25.3	Closing depths	106
25.4	Pushing and popping the stack	107
26	Data arrays	108
27	HTML entities	109
28	HTML filename generation	109
29	Homepage link	112
30	\PrintStack diagnostic tool	112

31	Closing stack levels	113
32	Forcing a new PDF page	113
33	HTML tags, spans, divs, elements	114
33.1	Mapping L ^A T _E X Sections to HTML Sections	114
33.2	HTML tags	114
33.3	Block tags and comments	116
33.4	Div class and element class	117
33.5	Single-line elements	118
33.6	HTML5 semantic elements	118
33.7	High-level block and inline classes	119
33.8	Closing HTML tags	120
34	Paragraph handling	120
35	Paragraph start/stop handling	124
36	Page headers and footers	126
37	CSS	127
38	HTML meta description and author	128
39	Footnotes	129
39.1	Regular page footnotes	129
39.2	Minipage footnotes	129
39.3	Titlepage thanks	129
39.4	Regular page footnote implementation	129
40	Marginspars	131
41	Splitting HTML files	131
42	Sectioning	137
42.1	Book class commands	137
42.2	Sectioning support macros	138
42.3	\section and friends	143
43	Starting a new file	145
44	Starting HTML output	147
45	Ending HTML output	150
46	Titles and the <i>titling</i> package	152
46.1	Setting the title, etc.	153
46.2	Changes to HTML titlepage and titlingpage	155

46.3	Printing the title, etc. in HTML	156
46.4	Printing the title, etc. in print form	159
46.5	\maketitle for print output	160
46.6	\maketitle for HTML output	162
47	Abstract	165
48	Quote and verse	166
48.1	Citations and attributions	166
48.2	Quotes, quotations	167
48.3	Verse	167
49	Verbatim	168
50	Fancyvrb	168
51	Theorems	174
52	Lists	175
52.1	Itemize	175
52.2	Enumerate	177
52.3	Description	177
53	Tabular	178
53.1	Token lookahead	180
53.2	Booleans	180
53.3	Handling & and !	181
53.4	Handling \\	183
53.5	Variables	184
53.6	Parsing @, >, <, ! columns	185
53.7	Parsing 'l', 'c', or 'r' columns	187
53.8	Parsing 'p', 'm', or 'b' columns	188
53.9	Parsing 'D' columns	188
53.10	Parsing the column specifications	188
53.11	Starting a new row	192
53.12	Data opening tag	193
53.13	Midrules	194
53.14	Multicolumns	195
53.14.1	Parsing multicolumns	195
53.14.2	High-level multicolumn interface	198
53.14.3	Longtable captions	199
53.14.4	\tabledatamulticolumntag	201
53.15	Multirow	202
53.16	Utility macros inside a table	202
53.17	Checking for a new table cell	202
53.18	\mrowcell	204
53.19	New \tabular definition	204

53.20	Array	206
54	Cross-references	207
54.1	Setup	207
54.2	Zref setup	209
54.3	Labels	210
54.4	References	211
54.5	Hyper-references	214
55	Floats	217
55.1	Float captions	218
55.1.1	Caption inside a float environment	219
55.1.2	Caption and LOF linking and tracking	220
56	Table of Contents, LOF, LOT	223
56.1	Reading and printing the TOC	223
56.2	High-level TOC commands	226
56.3	Side TOC	226
56.4	Low-level TOC line formatting	227
57	Index and glossary	230
58	Math	232
58.1	Inline and display math	234
58.2	MathJax support	236
58.3	Equation environment	239
58.4	AMS Math environments	240
58.4.1	Support macros	240
58.4.2	Environment patches	242
59	Lateximages	246
60	center, flushleft, flushright	252
61	Siunitx	253
62	Graphics	253
62.1	\graphicspath	254
62.2	Length conversions and graphics options	254
62.3	\includegraphics	257
62.4	\rotatebox, \scalebox, \reflectbox	261
62.5	Null functions	264
63	Cleverref	264
64	Picture	265
65	Boxes and Minipages	266

65.1	Counters and lengths	267
65.2	Footnote handling	267
65.3	Minipage handling	267
65.4	Parbox, makebox, framebox, fbox, raisebox	271
66	Direct formatting	272
67	Skips, spaces, font sizes	275
68	\phantomsection	282
69	\LaTeX{} and other logos	282
69.1	HTML logos	282
69.2	Print logos	284
70	\AtBeginDocument, \AtEndDocument	284
71	lwarfpmk.conf	285
72	Trademarks	286
2	lwarf-newproject.sty	287
73	lwarf-newproject	287
73.1	project_html.tex	289
73.2	project.lwarfpmkconf	289
73.3	lwarf.css	290
73.4	lwarf_sagebrush.css	314
73.5	lwarf_formal.css	319
73.6	sample_project.css	323
73.7	lwarf_html.xdy	324
73.8	lwarf_mathjax.txt	324
73.9	Lwarfpmk option	326
3	lwarf-abstract.sty	337
74	Abstract	337
4	lwarf-afterpage.sty	339
75	Afterpage	339

5	l warp-algorithmicx.sty	340
76	Algorithmicx	340
6	l warp-alltt.sty	340
77	Alltt	341
7	l warp-amsthm.sty	342
78	AMSthm	342
8	l warp-bookmark.sty	344
79	Bookmark	344
9	l warp-booktabs.sty	346
80	Booktabs	346
10	l warp-ccaption.sty	346
81	Ccaption	346
11	l warp-changepage.sty	347
82	Changepage	347
12	l warp-cutwin.sty	348
83	Cutwin	348
13	l warp-dcolumn.sty	348
84	Dcolumn	349
14	l warp-draftwatermark.sty	350
85	Draftwatermark	350

15	lwarf-ellipsis.sty	350
86	Ellipsis	350
16	lwarf-emptypage.sty	351
87	Emptypage	351
17	lwarf-endnotes.sty	352
88	Endnotes	352
18	lwarf-enumerate.sty	353
89	Enumerate	353
19	lwarf-epigraph.sty	354
90	Epigraph	354
20	lwarf-eso-pic.sty	355
91	Eso-pic	355
21	lwarf-everypage.sty	356
92	Everypage	356
22	lwarf-extramarks.sty	356
93	Extramarks	356
23	lwarf-fancyhdr.sty	357
94	Fancyhdr	357
24	lwarf-float.sty	358
95	Float and \newfloat	358

25 l warp-floatflt.sty	360
96 Floatflt	360
26 l warp-floatrow.sty	361
97 Floatrow	361
27 l warp-fontenc.sty	367
98 Fontenc	367
28 l warp-fontspec.sty	367
99 Fontspec	367
29 l warp-footmisc.sty	367
100 Footmisc	367
30 l warp-footnote.sty	369
101 Footnote	369
31 l warp-footnotehyper.sty	370
102 Footnotehyper	370
32 l warp-framed.sty	371
103 Framed	371
33 l warp-ftnright.sty	374
104 Ftnright	374
34 l warp-geometry.sty	374
105 Geometry	374

35 lwarf-glossaries.sty	375
106 Glossaries	375
36 lwarf-graphics.sty	376
107 Graphics	376
37 lwarf-graphicx.sty	376
108 Graphicx	376
38 lwarf-hyperref.sty	377
109 Hyperref	377
39 lwarf-indentfirst.sty	380
110 Indentfirst	380
40 lwarf-inputenc.sty	380
111 Inputenc	380
41 lwarf-keyfloat.sty	381
112 Keyfloat	381
42 lwarf-layout.sty	381
113 Layout	382
43 lwarf-letterspace.sty	382
114 Letterspace	382
44 lwarf-lettrine.sty	383
115 Lettrine	383

45 l warp-lips.sty	384
116 Lips	384
46 l warp-listings.sty	385
117 Listings	385
47 l warp-longtable.sty	389
118 Longtable	389
48 l warp-lscape.sty	391
119 Lscape	391
49 l warp-ltcaption.sty	391
120 Ltcaption	391
50 l warp-marginfix.sty	392
121 Marginfix	392
51 l warp-marginnote.sty	393
122 Marginnote	393
52 l warp-mcaption.sty	393
123 Mcaption	393
53 l warp-mdframed.sty	394
124 Mdframed	394
54 l warp-microtype.sty	401
125 Microtype	401

55 l warp-mparhack.sty	402
126 Mparhack	402
56 l warp-multicol.sty	403
127 Multicol	403
57 l warp-multirow.sty	405
128 Multirow	405
58 l warp-nameref.sty	405
129 Nameref	405
59 l warp-needspace.sty	406
130 Needspace	406
60 l warp-newinclude.sty	406
131 Newinclude	406
61 l warp-newunicodechar.sty	407
132 Newunicodechar	407
62 l warp-nextpage.sty	407
133 Nextpage	407
63 l warp-nowidow.sty	407
134 Nowidow	408
64 l warp-ntheorem.sty	409
135 Ntheorem	409

65 lwarf-pagenote.sty	420
136 Pagenote	420
66 lwarf-parskip.sty	421
137 Parskip	421
67 lwarf-placeins.sty	421
138 Placeins	421
68 lwarf-ragged2e.sty	422
139 Ragged2e	422
69 lwarf-rotating.sty	423
140 Rotating	423
70 lwarf-setspace.sty	424
141 Setspace	424
71 lwarf-showidx.sty	425
142 Showidx	425
72 lwarf-showkeys.sty	425
143 Showkeys	425
73 lwarf-sidecap.sty	426
144 Sidecap	426
74 lwarf-sidenotes.sty	427
145 Sidenotes	427

75 l warp-soul.sty	429
146 Soul	429
76 l warp-subfig.sty	431
147 Subfig	431
77 l warp-tabularx.sty	437
148 Tabularx	437
78 l warp-tabulary.sty	438
149 Tabulary	438
79 l warp-textpos.sty	439
150 Textpos	439
80 l warp-theorem.sty	440
151 Theorem	440
81 l warp-threeparttable.sty	444
152 Threeparttable	444
82 l warp-tikz.sty	445
153 Tikz	445
83 l warp-titleps.sty	446
154 Titleps	446
84 l warp-titlesec.sty	448
155 Titlesec	448

85 lwarf-tletoc.sty	449
156 Tletoc	449
86 lwarf-titling.sty	450
157 Titling	450
87 lwarf-tocloft.sty	451
158 Tocloft	451
88 lwarf-trivfloat.sty	456
159 Trivfloat	456
159.1 Combining \newfloat, \trivfloat, and algorithmicx	456
89 lwarf-ulem.sty	458
160 Ulem	458
90 lwarf-verse.sty	460
161 Verse	460
91 lwarf-wallpaper.sty	462
162 Wallpaper	462
92 lwarf-wrapfig.sty	463
163 Wrapfig	463
93 lwarf-xcolor.sty	465
164 Xcolor	465

94 l warp-xfrac.sty **468**

165 Xfrac **468**

Change History and Index **471**

List of Figures

1	tutorial.tex listing	42
---	----------------------	----

List of Tables

1	L ^A T _E X–HTML generation — l warp package — Supported functions	25
2	Additional supported packages	29
3	Required software programs	34
4	Files created in a new project	44
5	Section depths and HTML headings	85
6	Tabular baseline	189
7	Tabular HTML column conversions	190
8	Cross-referencing data structures	208
9	Float data structures	217

Package 1

lwarf.sty

1 Updates

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

For a detailed list of changes, see the Change History on page [471](#).

v0.29:

- Add: `lwarpmklang` option for `lwarf-newproject` and `lwarf`. Sets the language to use while processing the glossary.
- 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.

v0.27:

- Improved documentation for MacOS install.
- Fix for `microtype` with X_EL^AT_EX and L^Au_LAT_EX.
- Fix for table footnote paragraph tags.
- Adds `lettrine`, `ulem`, and `soul`.

v0.26:

- Improved installation instructions for MiK_TE_X regarding generating the `lwarpmk` executable.
- Footnotes are now supported by L^AT_EX boxes instead of pagenotes. `pagenote` now works as per the print version. `footnote`, `footnotehyper`, `footmisc`, `endnotes`, `marginnote`, and `sidenotes` are also supported.
- L^AT_EX labels now are used to track the page numbers of `latexamges`. This allows the correct inclusion of `lateximages` in footnotes, pagenotes, and endnotes.
- `cutwin` and `floatflt` are also supported.

v0.25:

- Fix: Allows `graphicx` and `graphicsx` before `lwarp` because X_EL^AT_EX and L^Au_LAT_EX use `xunicode` which uses `graphics`.
- Package support for `framed`, several theorem packages, and ellipses.

v0.24:

- `tikz`'s `babel` library is load automatically as needed.
- `subfig` has been added, along with `lofdepth` and `lotdepth`.
- `picture` and `tikzpicture` now may be inline.

v0.22:

- Support has been added for tabular column types D, !, and X. Unknown column types are converted to l.
- Additional packages are supported, including `abstract`, `dcolumn`, `tabularx`, and `variorref`.

v0.21:

- Documentation for installing on Windows has been updated and improved.
- For Windows compatibility, the `latexamges` shell script has been replaced with a `lateximages.txt` file, which is parsed by `lwarpmk` to generate `latexamges`. This does not require any changes in the user's code.

- Windows `lwarpmk` again now functions.
- For improved error handling, `lwarp` now verifies the order in which packages are loaded, and signals an error for misplaced packages. `inputenc`, `fontenc`, `newunicode`, and `fontspec` must be loaded before `lwarp`, and the other packages which `lwarp` knows about must be loaded after.
- `lwarp` no longer requires a `\title` be assigned.

v0.20:

- The `makefile` and related infrastructure has been replaced by the `lwarpmk` utility. This provides increased portability, reduced dependencies, and much simpler installation and setup.
- The `lwarp-newproject` package is now used to locally create support files.
- The print and HTML versions of a document may co-exist with their own sets of auxiliary files.
- Package handling is now controlled by a modular system which looks for and loads an `lwarp-<package>` version if available.
- High-level source compatibility is provided for all supported packages, almost totally eliminating the need for `warpprint` and `warpHTML` environments.
- A large number of additional packages are supported.
- A new tutorial is included in the documentation, and many obsolete sections have been removed.
- `\NewHTMLdescription` sets the HTML meta description tag for each file. See section 6.8.



- `\HTMLFileName` may now be empty, allowing filenames without a prefix. Lwarp no longer automatically appends a - character. For existing projects, add a - to the end of `\HTMLFileName`.



- `\HomeHTMLFileName` and `\HTMLFileName` no longer use escaped underscore characters. Underscores may be used in filenames as-is.
- `lwarp` now tries to auto-detect the operating system, and `\warpOSwindows` is only needed if the auto-detection fails to detect Windows.
- Tabular column types @, >, and < are now supported.
- `BlockClass` and `\InlineClass` add an optional style.
- The `sidebar` and `example` environments have been moved to the test suite, and are no longer included in `lwarp`.

v0.19:

- MATHJAX now may be used to display math via the `mathjax` option. See sections 6.2 and 6.11.3. To use MathJax with a pre-existing project, copy or link the file `lwarp_mathjax.txt` to the project's directory.

- `\rule` added, supporting width, height, raise, `\textcolor`.
- `\LateximageFontSizeName` provides user-adjustable font size for math and `latexitimages`.
- `\minipagefullwidth` requests that the next minipage be full-width in HTML, but still the assigned width in print.
- `minipage` improved side-by-side rendering.
- CSS class `tablenotes` is provided for table note items.
- ⚠ • `\warpprintonly` replaces `\rowprintedonly`, and `\warpHTMLonly` is added. These behave like the `warpprint` and `warpHTML` environments, and are generally useful, so they replace the previously table-specific syntax.
- `cleveref` is loaded `\AtEndPreamble` for improved reliability. See section 63.
- `\xfracHTMLfontsize` controls `xfrac` font size in HTML.
- `Tikz` improved catcode handling.

v0.18:

- The `verse` package and the `verse`-related commands from the `memoir` package are now supported.
- Responsive web design has been improved for the `sideTOC`.
- `\includegraphics` now maintains relative sized for em, ex, and %.

v0.17:

- `mdframed` package is supported.

v0.16:

- Font and input encoding are now controlled by the user, and `lwarp` is loaded after fonts have been selected.
- Support for X_EL^AT_EX and LuaL^AT_EX. See section 6.1.

2 Introduction

The lwarf project aims to allow a rich L^AT_EX document to be converted to a reasonable HTML interpretation. No attempt has been made to force L^AT_EX to provide for every HTML-related possibility, and HTML cannot exactly render every possible L^AT_EX 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 lwarf to process the usual image formats. While generating HTML output, SVG files are used in place of PDF. Other formats such as JPG are used as-is.

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

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

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. lwarf maintains L^AT_EX control for cross-referencing and equation numbering / formatting.

The lwarf package allows L^AT_EX to directly generate HTML5 tags from a L^AT_EX 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 L^AT_EX for sharing math. The author has no particular opinion on the matter, except to say that in this case L^AT_EX 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.

A special `l warp-newproject` package is used to set up the auxiliary files necessary for the HTML conversion. 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, `l warp` first looks to see if it has its own modified version to use instead of the usual L^AT_EX version. These `l warp-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 25.)

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

2.1 Supported packages and features

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

Table 1 lists some of the various 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 — `l warp` package — Supported functions

Category	Status
Engines:	pdfL ^A T _E X, XeL ^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.

l warp 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.
Table of Contents, Figures, Tables:	Supported, with hyperlinks.
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.
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.
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 L ^A T _E X source for the math expression. MathJax supported as an alternative. <i>AMS</i> environments are supported. User-defined macros are available during conversion, due to native L ^A T _E X processing.
Theorems:	Support for native L ^A T _E X theorems, plus <code>theorem</code> , <code>amsthm</code> , <code>ntheorem</code> .
Floats:	Appear where declared. <code>float</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> .

l warp Supported Functions — continued

Category	Status
<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, tabulary, threeparttable, multirow:</code>	Emulated.
<code>longtable:</code>	Emulated. Converted to a tabular. 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 width or trim options.
<code>graphics, 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.
<code>Lists:</code>	Supported
<code>enumitem:</code>	Supported, although spacing is still controlled by CSS.
<code>Environments:</code>	Standard L ^A T _E X environments are supported.
<code>picture and 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.
<code>fancyvrb:</code>	Supported except for verbatim footnotes.
<code>framed, mdframed:</code>	Supported

l warp Supported Functions — continued

Category	Status
multicol:	Emulated, with <code>css3</code> . Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported.
siunitx:	Supported except for <code>per-mode=fraction</code> .
xfrac:	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 f, q, t are intentionally turned off because many simpler browsers do not display them correctly. Modern full-featured browsers re-create these ligatures on-the-fly.
Horizontal space:	HTML output for thin-unbreakable, unbreakable, <code>\enskip</code> , <code>\quad</code> , <code>\qquad</code> , <code>\hspace</code> .
Rules:	<code>\rule</code> with width, height, raise, text color.
HTML reserved characters:	<code>\&</code> , <code>\textless</code> , and <code>\textgreater</code> are converted to HTML entities.
xcolor:	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 L ^A T _E X package is used.
Emulated:	The L ^A T _E X 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 each of the following in table 2, and probably others which have not yet been tested. Many

Table 2: Additional supported packages

`babel`, `bm`, `calc`, `cleveref`, `csquotes`, `enumitem`, `fancyvrb`, `fileerr`, `newtxmath`, `siunitx`, `somedefs`, `tikz`, `trace`, `variorref`, `xspace`

are simply nullified 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 L^AT_EX or other document to HTML. Where an existing L^AT_EX document is to be converted to HTML, l warp 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 l warp 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 L^AT_EX 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, l warp 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 L^AT_EX 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** **L^AT_EXML:** <http://dlmf.nist.gov/LaTeXML/>

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

Prog **LaTeX2HTML** **L^AT_EX2HTML:** <http://www.latex2html.org/>
and <http://ctan.org/pkg/latex2html>.

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

Finally, Glad_{TEX} may used to directly insert L_AT_EX math into HTML:

Prog Glad_{TeX} Glad_{TEX}: <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 AsciiDoctor AsciiDoc: <http://asciidoc.org/> (The original version.)

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

Asciidoctor-LaTeX:

<http://www.noteshare.io/book/asciidoctor-latex-manual>

Prog Asciidoctor-LaTeX <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

Prog LibreOffice years in their abilities to represent math with a L_AT_EX-ish input syntax, unicode
Prog OpenOffice math fonts, and high-quality output, and also generate HTML with varying success.

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, L^AT_EX 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 L^AT_EX is comparably easy to learn, while L^AT_EX 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 L^AT_EX 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 L^AT_EX back end, yielding high-quality results especially when the L^AT_EX template is adjusted, but they lose the ability to use L^AT_EX macros and other L^AT_EX source-document features.

The effort required to customize the output of each markup system varies. For print output, L^AT_EX 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 lwarf, CSS is used, and much HTML output is adjusted through the usual L^AT_EX optional macro parameters, but further customization may require patching L^AT_EX 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 l warp package

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

Pre-installed: Try entering into a command line:

Enter ⇒ kpsewhich l warp.sty

If a path to l warp.sty is shown, then l warp is already installed.

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

Enter ⇒ tlmgr install l warp

MiK_TE_X: If using MiK_TE_X, try using the package installer to install the package l warp. Also update the package miketex-misc, which will install the l warpmk executable.

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

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

1. See <http://ctan.org/pkg/l warp> for the l warp package.
2. Download the TDS archive: l warp.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: pdflatex, xelatex, or lualatex.

The lwarf package: This package.

The lwarf-newproject package: Accompanies lwarf, and used to create configuration files.

The lwarpmk utility: Provided along with this package. This should be an operating-system executable in the same way that pdflatex or latexmk is. It is possible to have the lwarf-newproject 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 lwarf 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 ⇒ `mktextlsr`

— or —

Enter ⇒ `texhash`

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

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

1. See <http://ctan.org/pkg/lwarp> for the l warp package.
2. Download the zip archive **lwarp.zip** into your own **lwarp** directory.
3. Unpack **lwarp.zip**.
4. Locate the contents **lwarp.dtx** and **lwarp.ins**
5. Create the documentation:

Enter ⇒ `pdflatex lwarp.dtx`

(several times)

6. Create the **.sty** files:

Enter ⇒ `pdflatex lwarp.ins`

7. Copy the **.sty** files somewhere such as the T_EX Live local tree found in the previous CTAN TDS section, under the subdirectory:

`<texlocal>/tex/latex/local/lwarp`

8. Copy the documentation **lwarp.pdf** to a **source** directory in the local tree, such as:

`<texlocal>/doc/local/lwarp`

9. Renew the cache:

Enter ⇒ `mktextlsr`

— or —

Enter ⇒ `texhash`

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

10. See section [4.2.1](#) to generate your local copy of **lwarpmk**.
11. Once the local version of **lwarpmk.lua** is installed, it may be made available system-wide as per section [4.2](#).

Project-local ctan .dtx and .ins files: The .dtx and .ins files may be downloaded to a project directory, then compiled right there, alongside the document source files. The resultant *.sty and l warpmk.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 l warp before deciding whether to permanently install it.

4.2 Installing the l warpmk utility

(Note: If l warpmk 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 l warp package is installed, you may need to setup the l warpmk utility:

1. At a command line, try executing l warpmk. If the l warpmk help message appears, then l warpmk 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 l warpmk executable for you.

Otherwise, continue with the following:

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

Enter ⇒ kpsewhich l warpmk.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 l warpmk:

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 l warpmk.lua:

Enter ⇒ ln -s <pathtolwarpmk.lua> l warpmk

- (c) Make the link executable:

```
Enter ⇒ chmod 0755 lwarpmk
```

Windows TeX Live: Create a new `lwarpmk.exe` file:

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

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

- (a) Locate the binaries. These will be in a directory such as:

```
C:\Program Files\MiKTeX 2.9\miktex\bin\x64
```

In this directory you will find programs such as `pdflatex.exe` and `makeindex.exe`.

- (b) Create a new file named `lwarpmk.bat` containing:

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

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

4.2.1 Using a local copy of `lwarpmk`

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

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

```
\usepackage[lwarpmk]{larp-newproject}
```

2. When the tutorial is compiled with `pdflatex`, the file `lwarpmk.lua` will be generated along with the other configuration files.

3. `lwarpmk.lua` may be used for this project:

Unix:

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

```
Enter ⇒ chmod 0755 lwarpmk.lua
```

- (b) Compile documents with

```
Enter ⇒ ./lwarpmk.lua html
```

```
Enter ⇒ ./lwarpmk.lua print
```

etc.

- (c) It may be useful to rename or link to a version without the `.lua` suffix.

Windows:

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

Enter ⇒ `texlua lwarpmk.lua html`

Enter ⇒ `texlua lwarpmk.lua print`

etc.

Or:

Enter ⇒ `lwarpmk html`

Enter ⇒ `lwarpmk print`

etc.

4.3 Installing additional utilities

To test for the existence of the additional utilities:

Enter the following in a command line. If each programs' version is displayed, then that utility is already installed. See table 3 on page 34.

Enter ⇒ `luatex -version`

Enter ⇒ `xindy -version`

Enter ⇒ `latexmk -version`

Enter ⇒ `perl -version`

Enter ⇒ `pdfcrop -version`

Enter ⇒ `pdftotext -v`

Enter ⇒ `pdfseparate -version`

Enter ⇒ `pdftocairo -v`

To install `xindy`, `latexmk`, and `pdfcrop`:

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

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

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 system's package manager.
 Prog `pdftocairo`

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 34.
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\<myname>\Desktop\Poppler\poppler-0.5_x86\poppler-0.5\bin`

Be sure to include `\bin`.

9. Click "Ok" when done.

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

1. Download and install a version of Perl, such as Strawberry Perl, to a directory without a space in its name, such as `C:\Strawberry`.
2. Edit the Path as seen above for the Poppler utilities.

3. Enter the **bin** directory of the Perl utility, such as:

C:\Strawberry\perl\bin

Be sure to include \bin.

4. Click "Ok" when done.

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

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`.
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`

⚠ Note: `.txt` suffix!

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`.

⚠ 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.)

A number of new files are created when `tutorial.tex` is compiled, as shown in table 4. These files are created by the `lwarp-newproject` package.

(Two of the new files are configuration files for the helper program `lwarpmk`. Whenever a print version of the document is created while the `lwarp-newproject` package is used, 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 lwarf 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

% --- HTML FILENAME AND LATEXMK SETTINGS ---
% \newcommand{\HomeHTMLfilename}{index} % Filename of the homepage.
% \newcommand{\HTMLfiveName}{node-} % Filename prefix of other pages.
% \newcommand{\UseLatexmk}{true}% Uncomment to use latexmk

% --- LWARP IS LOADED NEXT ---
\usepackage{lwarf-newproject}   % Possibly with the [lwarpmk] option.
\usepackage{lwarf}
% \boolfalse{FileSectionNames} % If false, numbers the files.

% --- OTHER PACKAGES ARE LOADED AFTER LWARP ---
\usepackage{makeidx} \makeindex
\usepackage{xcolor}

\usepackage{hyperref,cleveref} % LOAD THESE LAST!

% --- LATEX AND HTML CUSTOMIZATION ---
\title{The Lwarf 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
\SetFirstPageTop{Name and \fbox{HOMEPAGE LOGO}}
\SetPageTop{\fbox{LOGO}}
\SetPageBottom{Contact Information and Copyright}
\NewCSS{lwarf_sagebrush.css}
```

```
\begin{document}
\maketitle % (or a titlepage environment)

% --- An abstract may be placed here. ---

\tableofcontents \listoffigures % --- MUST BE BEFORE THE FIRST SECTION.

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

\section{Some math}

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

\printindex

\end{document}
```

Table 4: Files created in a new project

tutorial.pdf: The PDF output from L^AT_EX.

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 `tutorial.aux`, `tutorial.log`. 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: `l warp.css`, `l warp_formal.css`, `l warp_sagebrush.css` These files are standard for `l warp`, 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 `l warp` defaults.

l warp_html.xdy: Used by `l warp` while creating an index. This file should not need to be modified by the user.

l warp_mathjax.txt: Inserted into the HTML files when MathJax is used to display math. This file should not need to be modified by the user.

comment.cut: A temporary file used by `l warp` to conditionally process blocks of text. This file may be ignored.

When the `lwarpmk` option is given to the `l warp-newproject` package:

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

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

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

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

5.2 Compiling the print version with `lwarpmk`

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

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

1. Re-compile the print version:

Enter ⇒ `lwarpmk print`

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

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

Enter ⇒ `lwarpmk print`

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

4. Force a recompile to occur.

Enter ⇒ `lwarpmk again`

Enter ⇒ `lwarpmk print`

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

5. Process the index.⁴

Enter ⇒ `lwarpmk printindex`

6. Recompile again to include the index.

Enter ⇒ `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. Set its language using the `lwarpmklang` option for `lwarf-newproject` and `lwarf`.

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 L^AT_EX to process `tutorial_html.tex` to create `tutorial_html.pdf`.
- (b) `pdftotext` is then used to convert into 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 L^AT_EX source until the images of the math expressions have not yet 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 L^AT_EX 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. Set its language using the `lwarpmklang` option for `lwarpmk` and `lwarpmkhtml`.

5.4 Generating the SVG images

math as SVG images

By default l warp represents math as SVG images with the L^AT_EX source included in `alt` tags. In this way, the math displays as it was drawn by L^AT_EX, and the L^AT_EX source may be copied and pasted into some other document.

picture and Tikz

l warp uses the same mechanism for `picture` and `Tikz` environments.

1. Create the SVG images:

Enter ⇒ `lwarpmk limages`

Enter ⇒ `lwarpmk html`

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

⚠ Adding/removing

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

⚠ Lots of files!

Expressing math as SVG images has the advantage of representing the math exactly as L^AT_EX 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, change

```
\usepackage{l warp}
```

to

```
\usepackage[mathjax]{l warp}
```

2. Recompile

```
Enter ⇒ l warpmk 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 L^AT_EX source.

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

5.6 Changing the CSS style

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

The styles provided by lwarf include:

lwarf.css: A default style if \NewCSS is not used. This style is comparable to a plain L^AT_EX document. To set this style, you may use \NewCSS{lwarf.css}, or no \NewCSS call at all.

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

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

To see each style in use, change the \NewCSS entry in the tutorial lwarpmk.html, 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 \NewCSS{<project>.css}.

⚠ Rename it! Note that sample_project.css is overwritten whenever lwarf-newproject 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 lwarf.css, and this entry may be changed to load lwarf_formal.css or lwarf_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

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

⚠ Placement! Note that if changes are made, it is best to first use `lwarpmk clearall` to clear all the HTML, PDF, and auxiliary files. Also, if `\HomeHTMLFilename`, `\HTMLFileName`, or `\UseLatexmk` are changed, it is necessary to reenable the `lwarp-newproject` package and then recompile the print version in order to recreate the configuration files for `lwarpmk`.

Placed in the preamble just before `lwarp` is loaded:

<code>\HomeHTMLFilename</code>	<code>\HomeHTMLFilename</code> : Filename of the homepage, without the “.html” suffix. Defaults to the <code>\BaseJobname</code> . A common setting is: <code>\newcommand{\HomeHTMLFilename}{index}</code> causing the homepage to be the file <code>index.html</code> . Underscores are allowed in <code>\HomeHTMLFilename</code> and <code>\HTMLFileName</code> , but usually must be escaped elsewhere, such as <code>\item [\href{file_name.pdf}{text}] \</code>
<code>\HTMLfiveName</code>	<code>\HTMLfiveName</code> : A filename prefix for the rest of the HTML web pages. Useful for numbered web pages with a common prefix. May be empty.
<code>\UseLatexmk</code>	<code>\UseLatexmk</code> : A macro which 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. Set to true with: <code>\newcommand*{\UseLatexmk}{true}</code>

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

<code>Ctr tocdepth</code>	<code>tocdepth</code> : Sectioning depth of the table of contents. See section 12 for a list of L ^A T _E X stack depths.
<code>Ctr SideTOCDepth</code>	<code>SideTOCDepth</code> : Sectioning depth of the sideTOC. Defaults to 1, causing the sideTOC to show sections but not subsections.
<code>sidetoc</code>	Each subpage of the website has its own small table of contents on the side (the “sideTOC”). Its depth is set by <code>SideTOCDepth</code> . 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: <code>SideTOCDepth = FileDepth</code> or

`SideTOCDepth = FileDepth+1`



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

Ctr `FileDepth`

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

- To place the entire file into one HTML page, use:

`\setcounter{FileDepth}{-5}`

- To split the HTML file at `\section` depth, use:

`\setcounter{FileDepth}{1}`



- To ensure that the HTML pages/files are accessible:
Place a `\tableofcontents` somewhere before the first section break (therefore in the “home page”), and set

`tocdepth >= FileDepth`

Bool `CombineHigherDepths`

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

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

Inaccesible 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` is prefixed.

[HTML filenames](#)

Example HTML filenames:

Numbered html nodes:

Example: Homepage `index.html`, and `node-1`, `node-2`. (See `\SetHTMLFileName` to number grouped by chapter, for example.)

```
\newcommand{\HomeHTMLFileName}{index}
\newcommand{\HTMLfiveName}{node-}
\usepackage{lwarp}
\boolfalse{FileSectionNames}
```

Named html sections, no prefix:

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

```
\newcommand{\HomeHTMLFileName}{index}
\newcommand{\HTMLfiveName}{}
\usepackage{lwarp}
\booltrue{FileSectionNames}
```

Named html sections, with prefix:

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

```
\newcommand{\HomeHTMLFileName}{mywebsite}
\newcommand{\HTMLfiveName}{mywebsite-}
\usepackage{lwarp}
\booltrue{FileSectionNames}
```

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

`\MetaLanguage`: The HTML language meta header. Defaults to en-US.

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

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

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

Placed in the home page before the first section break:

`\tableofcontents`: Used to place a table of contents on the home page.

This command must be used before the first file split, so that a way is available to navigate to other files from the homepage.

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

Placed before any sectioning command which causes a file break:

\NewCSS	\NewCSS: { <i>filename.css</i> } 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 \NewCSS command. If \NewCSS is not used, a default plain style is used to mimic printed L ^A T _E X output. <code>lwarp_sagebrush.css</code> is a semi-fancy colored style as shown in this tutorial. Change it to <code>lwarp_formal.css</code> for a more formal look, or comment out the \NewCSS command to see the default. \NewCSS may be used before each file break to set the css for individual pages of the website.
\NewHTMLdescription	\NewHTMLdescription: { <i>description</i> } Sets the HTML description tag for the following files. May be changed before each each sectioning command which would cause a file split.

Placed in the document wherever necessary:

Env \warpprint	warpprint: An environment which is only used while generating print output. Place here anything which does not apply to HTML and which may cause problems with lwarp. If lwarp knows about and emulates or supports a package then its related macros, lengths, counters, etc. probably won't have to be placed inside a warpprint environment, but unknown packages may cause problems which may be isolated from lwarp using this environment.
Env \warpHTML	warpHTML: An environment which is only used while generating HTML output. This is useful for website logos and other items which have no purpose in printed output.
\warpprintonly	\warpprintonly: { <i>contents</i> } A macro version of the warpprint environment.
\warpHTMLonly	\warpHTMLonly: { <i>contents</i> } A macro version of the warpHTML environment.

5.8 Using `latexmk`

`latexmk` is a L^AT_EX utility used to monitor changes in source files and recompile as needed.

1. In the tutorial's source code, at "HTML FILENAME AND LATEXMK SETTINGS", insert the line:

```
\newcommand{\UseLatexmk}{true}
```

2. Directly recompile the printed version of the document.

Enter ⇒ `pdflatex tutorial.tex`

(Or `xelatex` or `lualatex`)

lwarf 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_ELaTeX or LuaLaTeX may be used instead of L^AT_EX.

1. Remove the auxiliary files for the project:

Enter ⇒ lwarfmk cleanall

2. Be sure that

\usepackage{lwarf-newproject}

is still enabled in `tutorial.tex`.

3. Use `xelatex` or `lualatex` to recompile the printed version.

Enter ⇒ xelatex `tutorial.tex`

-or-

Enter ⇒ lualatex `tutorial.tex`

When the recompile occurs and `lwarf-newproject` is loaded, the configuration files for `lwarfmk` are modified to remember which T_EX engine was used. X_ELaTeX or LuaLaTeX will be used for future runs of `lwarfmk`.

4. To recompile the document:

Enter ⇒ lwarfmk print

-and-

Enter ⇒ lwarfmk html

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

5.10 Using a glossary

lwarf supports the `glossaries` package.

Opt `lwarf-newproject` To assign a language to be used while processing the glossary, use the `lwarpmklang` option:

`lwarf lwarpmklang`

```
\usepackage[lwarpmklang=english]{lwarf-newproject}
\usepackage[lwarpmklang=english]{lwarf}
```

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

`lwarf` 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 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.

 `XELATEX` and `LuALTEX` 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{lwarf}`:

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

- For `XELATEX` or `LuALTEX`:

- `fontspec` and font choices

Pkg `fontspec`

`ligatures`

`lwarf` 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[\sfamily]{Ligatures={NoCommon,TeX}}
\defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
```

- For `pdflatex`:

- `lmodern` or other font-related packages

- `fontenc`

- `inputenc`

- `newunicodechar`

- `\input glyphhtounicode.tex`

- `\input glyphhtounicode-cmr.tex%` from the `pdfx` package

Pkg `lmodern`

Pkg `fontenc`

Pkg `inputenc`

Pkg `newunicodechar`

File `glyphhtounicode`

- `\pdfgentounicode=1`
- Pkg `cmap` – `cmap`
- Pkg `textcomp` – `textcomp`
- Pkg `microtype`
`ligatures` – `microtype` is automatically used by `lwarf` to turn off f,q,t,T,Q ligatures for the same browser-related reasons shown above. Also, the monospaced font is used during HTML tag generation to turn off `TEX` ligatures.

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

6.2 lwarf package loading and options

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

Pkg `lwarf` Load the `lwarf` package immediately after the font and UTF-8 setup commands.

Opt `lwarf` `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 `lwarf-newproject` is loaded in print mode, it creates `<project>.html.tex`, which sets the `warpHTML` option before calling the user's source code `<project>.tex`. In this way, `<project>.tex` can `\usepackage{lwarf}` without any options to create a printed version, while `<project>.html.tex` will create an HTML version.

Opt `lwarf` `mathsvg` For math display, select `mathsvg` (default), or `mathjax`. For more information about the math options, see section 6.11.3.

6.3 Selecting the operating system

`lwarf` tries to detect which operating system is being used.

Prog `MS-WINDOWS` If `MS-WINDOWS` is not correctly detected, use the command `\warpOSwindows` in the document preamble after `lwarf` is loaded. This modifies the operating-system path separator used by `lwarf`.

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

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

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

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

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

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

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

`\warpprintonly {<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`
- Variable spaces such as `\vspace`. `\hfill` is turned into a `\quad`. Fixed spaces such as `\quad` are emulated correctly.
- 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 Commands for a successful HTML conversion

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

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

`\bfseries`, etc: Use `\textbf` instead.

`\centering`, `\raggedright`, `\raggedleft`:

Use the environments `center`, `flushright`, `flushleft` instead.

Superscripts and other non-math uses of math mode:

Use `x` instead of `x`

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

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

Filenames in lists:

filename underscore

Escape underscores in the filenames:

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

Side-by-side minipages:

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

\fbox around a minipage:

`\fbox` can only be used around inline items during HTML output.

For an `\fbox` around a minipage, you may:

- Place the `\fbox` command and its closing brace inside `\warpprint` environments.
- Use `\mdframed` instead.
- 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}
```

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

6.7 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 {<title>}`

⚠ 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 {<author>}`

⚠ 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 {<date>}`

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

`\subtitle {<subtitle>}`

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 {<published>}`

A new command which sets a publisher. The default is empty. In HTML, this will appear in a `<div>` class `published`.

`\thanks {<text>}`

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

6.8 HTML page meta descriptions

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

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

placement Use `\NewHTMLdescription` 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 `\NewHTMLdescription` 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 `\NewHTMLdescription`. It is best to use a unique description for each HTML file.

disabling To disable the generation of HTML description meta tags, use:

```
\NewCSSdescription{}
```

6.9 HTML page meta author

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

6.10 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:

```
/* ( --- 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`.

- \NewCSS For each section at which HTML files are split, \NewCSS 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.

6.11 Special cases and limitations

6.11.1 Text formatting

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

6.11.2 Footnotes and page notes

lwarf uses native L^AT_EX footnote code, although with its own \box to avoid the L^AT_EX output routine. The usual functions work as-is.

6.11.3 Math

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 L^AT_EX 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 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 tag 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.

⁸See section 165 regarding fonts and fractions.

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

MathML Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 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 L^AT_EX math expression to manually insert tags, but this has not yet been done.

footnotes in math

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

lateximage

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

siunitx

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

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

⚠ siunitx inside an equation

- `LATEX macros` `\includegraphics` 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 234.

6.11.4 Ntheorem

- ⚠ `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 AMS environments when the option `thref` is used. `l warp` does not share this bug, so equations with `\split`, etc, are numbered correctly with `l warp`'s `HTML` output, but not with the `print` output. It is recommended to use `cleveref` instead of `ntheorem`'s `thref` option.

6.11.5 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.
- ⚠ `\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.

6.11.6 Xcolor

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.
<code>\color</code> and <code>\pagecolor</code>	\color and \pagecolor are ignored. Use css or \textcolor where possible.

6.11.7 Tabular

column types	<ul style="list-style-type: none"> 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 l. tabularx ignores the width, but X columns do produce paragraph columns or multicolumns.
 <code>\multirow</code> & <code>\multicolumn</code>	<ul style="list-style-type: none"> Multirow and multicolumn cannot be used at the same time. (No rectangular holes wider than one column or taller than one row.) 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`

`\multirow with rules`

`rule at last row`

 `paragraphs`

- If a multirow reaches to the bottom of a table, and \bottomrule does not go over to that edge, try adding a line of empty cells below the \bottomrule. This may be a browser bug.
- If a \midrule is desired after the last row, an additional row of blank cells must be used.
- Multiple paragraphs in one cell of a p, b, m column must have \newline between paragraphs.

`\cmidrule width, trim`

`longtable` headings

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

For `longtable`, place headings and footings which do not apply to HTML inside `\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 lwarf’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 \\}
```

6.11.8 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`.

6.11.9 Save Boxes

LATEX boxes are placed inline and do not allow line breaks, so boxes with long contents may overflow the line during HTML conversion. This is mostly a problem when the boxes contain objects which themselves hold large HTML tags, such as

rotation commands with long contents. When this object overflows the line, some HTML code will be lost and the page will be corrupted.

6.11.10 Minipages

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

⚠ 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`.

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 are ignored. 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` tag, 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.

6.11.11 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, lwarf 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 `l warp` 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.

6.11.12 `float`, `trivfloat`, and/or `algorithmicx` together

 **package conflicts** If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 159.1.

6.11.13 `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`.

6.11.14 `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.

6.11.15 `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. `l warp` 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.

6.11.16 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.

6.11.17 cleveref and varioref packages

`cleveref` and `varioref` are supported, but printed page numbers do not map to HTML. See section 6.6 to redefine the message which is printed for page number references.

6.11.18 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. `l warp` 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
 Len `\leftmargini`
 Len `\TMLvleftskip`
 Len `\TMLleftmargini`

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

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

6.11.19 `siunitx` package

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

6.11.20 `newclude` package

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

Ex:

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

6.11.21 `newtxmath` package

Pkg `newtxmath` The proper load order is:

⚠ `loading sequence`

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

6.11.22 babel package

Pkg **babel** If using **babel** with French, use

 **French** `\frenchbsetup{StandardLists=true}`

to preserve the special HTML and `enumitem` list handling.

\CaptionSeparator Also, 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}{:~}
```

6.11.23 glossaries package

Pkg **glossaries** **xindy** is required for **glossaries**.

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

The page number list in the printed form would become `\nameref`s 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 **lwarf-newproject** **lwarpmklang** The packages **lwarf-newproject** and **lwarf** each take an option `lwarpmklang=english` to set the language used by **xindy**. Set the same option for both packages. This is passed to **xindy** using its `-L` option.

Opt **lwarf** **lwarpmklang**

6.11.24 enumitem package

Pkg **enumitem** **enumitem** is pre-loaded during HTML output. Many of the spacing options are rendered irrelevant by `pdftotext` and HTML. Numbering, labels, and `\newlist` function correctly.

6.11.25 enumerate package

Pkg **enumerate** **enumerate** conflicts with **enumitem** if both are loaded at the same time, but **lwarf** does not actually load **enumerate**. While generating HTML, **lwarf** only loads **enumitem**, and **enumerate** is simulated by **enumitem** using the functionality of the `shortlabels` option.

A problem may occur during print output if `enumitem` is loaded, either manually or by some other package such as `siunitx`. If these are used, `enumerate` will conflict with `enumitem` during print output.

7 EPUB conversion

`lwarf` 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 `lwarf` HTML files, and also for the EPUB, use `\HTMLAuthor {\{name\}}`. This assigns the name to the `<meta>` author tag. It may be set empty, and it defaults to `\theauthor`.

A special boolean is provided to simplify the process of converting `lwarf` 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 `lwarf` HTML output to EPUB, add

`\booltrue{FormatEPUB}`

to the project's source preamble after `\usepackage{lwarf}`. 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.

Calibre 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 `lwarf` 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 `lwarf`'s section headings, see table 5. For example, an `article` class document would break at `\section`, which is mapped to HTML heading level `<h4>`,

whereas a book class document would break at \chapter, which is HTML heading level <h3>. For Calibre, this option is found in

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

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

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 lwarf-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.

lwarf 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 lwarf 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

`lwarf` 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. `lwarf` provides some assistance in locating these frame boundaries, as shown below.

A special boolean is provided to simplify the process of converting `lwarf` 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 `lwarf` HTML output for easier import to a word processor, add

`\booltrue{FormatWordProcessor}`

to the project's source preamble after `lwarf` 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. `lwarf` 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 l warp

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 `\teximage` then displayed with an image of the resulting L^AT_EX output. See section 64 for an example of the `picture` environment.

To create a custom HTML block or inline CSS class, see section 33.7.

9.1 Creating an l warp version of a package

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

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

Anything the user might expect of the original package must be replaced or emulated by the new `l warp-` 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 `l warp-` package should first call either `\LWR@ProvidesPackageDrop` or `\LWR@ProvidesPackagePass`. If `Dropped`, the original print-version package is ignored, and only the `l warp-` 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 `l warp-` version continues loading as well. See section 135 (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.

The package `lwarp-newproject` must be loaded to set up the configuration files, but may be commented out from then on unless the configuration changes. A small amount of time is taken each time `lwarp-newproject` is run.

Remember that the configuration files are only rewritten when compiling the printed version of the document, and when `lwarp-newproject` is enabled.

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` 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.
File `lwarpmk.lua`

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-newproject` package.
2. Recompile the printed version of the document. The `lwarpmk` option causes `lwarp-newproject` to create a local copy of `lwarpmk.lua`
3. The `lwarpmk` option may now be removed from the `lwarp-newproject` 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`”.

9.4 Modifying `xindy` index processing

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

To customize index processing:

1. Copy `lwarf_html.xdy` to a new filename such as `projectname.xdy`
2. Make changes to `projectname.xdy`
3. Make a new version of `lwarfmk`, as previously described, which uses `xindy` with `projectname.xdy` instead of `lwarf_html.xdy`.

10 Troubleshooting

10.1 Using the `lwarf.sty` package

Also see:

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

[Section 6.6: Commands for a successful HTML conversion](#)

[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](#).

Obscure error messages:

- Be sure that a print version of the document compiles and that your document's L^AT_EX 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 `lwarfmk` again followed by `lwarfmk html` or `lwarfmk print` to compile the document one more time.

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

Use X^LA_TE_X or Lua^LA_TE_X.

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.

“Leaders not followed by proper glue”: This can be caused by a missing `1@<floattype>` or `1@<sectiontype>` definition. See `lwarp`’s definitions for examples.

Plain-looking document:

- The document’s CSS stylesheet may not be available, or may be linked incorrectly. Verify any `\NewCSS` 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 lwarf.dtx file

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

1. Delete the lwarf.sty file.
2. pdflatex lwarf.ins to generate a new lwarf.sty file.
3. pdflatex lwarf.dtx to recompile the lwarf.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.

Table 5: Section depths and HTML headings

Section	L <small>A</small> T <small>E</small> 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	span class = "paragraph"
subparagraph	5	span class = "subparagraph"
listitem	7	new for this package, used for list items

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 5. “Depth” here is not depth in the traditional computer-science stack-usage sense, but rather a representation of the nesting depth inside the LATEX document structure.

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

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

It is possible to use HTML5 `section` and `H1` for all levels, but this may not be well-recognized by older browsers.

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

13 Source Code

This is where the documented source code for `lwarf` 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 `lwarf` package.

line numbers The small numbers at the left end of a line refer to line numbers in the `lwarf.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

index entries 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: Green-colored tags in the left margin show which sections of source code apply to

for PRINT output: the generation of HTML, print, or both forms of output.

for HTML & PRINT:

`lwarf` source code begins on the following page.

14 Detecting the T_EX Engine — pdflatex, lualatex, xelatex

```

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

```

15 Unicode Input Characters

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

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

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{x}{\textttimes}
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 PDFT_EX, preserve upright quotes in verbatim text:

```

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

```

16 Handling package options

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

27 `\RequirePackage{kvoptions}`
 28 `\SetupKeyvalOptions{family=LWR,prefix=LWR@}`

Bool **warpingprint**

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

29 `\newbool{warpingprint}`
 30 `\newbool{warpingHTML}`
 31 `\newbool{mathjax}`

`\warpprintonly {<contents>}`

Only process the contents if producing printed output.

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

`\warpHTMLonly {<contents>}`

Only process the contents if producing HTML output.

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

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

34 `\includecomment{warpall}`

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

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

35 `\DeclareVoidOption{warpprint}{%}`
 36 `\PackageInfo{lwarp}{Using option 'warpprint'}`

```

37 \includecomment{warpprint}%
38 \excludecomment{warpHTML}%
39 \booltrue{warpingprint}%
40 \boolfalse{warpingHTML}%
41 }

```

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

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

```

42 \DeclareVoidOption{warpHTML}{%
43 \PackageInfo{l warp}{Using option 'warpHTML'}%
44 \excludecomment{warpprint}%
45 \includecomment{warpHTML}%
46 \booltrue{warpingHTML}%
47 \boolfalse{warpingprint}%
48 }

```

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

```

49 \DeclareVoidOption{mathsvg}{%
50 \PackageInfo{l warp}{Using option 'mathsvg'}%
51 \boolfalse{mathjax}%
52 }

```

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

```

53 \DeclareVoidOption{mathjax}{%
54 \PackageInfo{l warp}{Using option 'mathjax'}%
55 \booltrue{mathjax}%
56 }

```

Opt l warp **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.

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

Opt l warp **lwarpmklang** Sets the language to be assigned in **lwarpmk**'s configuration files. This is then used by **lwarpmk** while processing the glossary.

```
58 \DeclareStringOption[english]{lwarpmklang}
```

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

```

59 \includecomment{warpprint}%
60 \excludecomment{warpHTML}%
61 \booltrue{warpingprint}%
62 \boolfalse{warpingHTML}%
63 \boolfalse{mathjax}%
64
65
66 % \ExecuteOptions{warpprint,mathsvg}
67
68 \ProcessKeyvalOptions*\relax

```

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

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

```
\@nameedef {name} {defn}
```

Expands `<defn>` before assigning to `<name>`

```
70 \def\@nameedef#1{\expandafter\edef\csname #1\endcsname}
```

17 Misplaced packages

Several packages should only be loaded before `l warp`, and others should only be loaded after.

Packages which should only be loaded before `l warp` have their own

```
l warp-<packagename>.sty
```

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

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

```

71 \newcommand*\LWR@loadafter[1]{%
72 \@ifpackageloaded{#1}%
73 {%
74 \PackageError{l warp}%
75 {Package #1, or one which uses #1, must be loaded after l warp}%
76 {Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{l warp}.}%
77 Package #1 may also be loaded by something else, which must also be moved%
78 after l warp.}%
79 }

```

```
80 {}  
81 }
```

\LWR@loadbefore {*packagename*} Error if this package is after l warp.

```
82 \newcommand*{\LWR@loadbefore}[1]{%  
83 \PackageError{l warp}{  
84 {Package #1 must be loaded before l warp}  
85 {Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{l warp}.}  
86 }}
```

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

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

```
87 \newcommand*{\LWR@loadnever}[2]{%  
88 \PackageError{l warp}{  
89 {Package #1 does not work with l warp's HTML conversion.  
90 Please use the #2 package instead}  
91 {Package #1 conflicts with l warp in some way, but package #2 probably will work instead.}  
92 }}
```

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

The following packages must be loaded after l warp:

```
93 \LWR@loadafter{abstract}  
94 \LWR@loadafter{afterpage}  
95 \LWR@loadafter{algorithmicx}  
96 \LWR@loadafter{alltt}  
97 \LWR@loadafter{amsthm}  
98 \LWR@loadafter{bookmark}  
99 \LWR@loadafter{booktabs}  
100 \LWR@loadafter{ccaption}  
101 \LWR@loadafter{changepage}  
102 \LWR@loadafter{cutwin}  
103 \LWR@loadafter{dcolumn}  
104 \LWR@loadafter{draftwatermark}  
105 \LWR@loadafter{ellipsis}  
106 \LWR@loadafter{emptypage}  
107 \LWR@loadafter{enumerate}  
108 \LWR@loadafter{epigraph}  
109 \LWR@loadafter{eso-pic}  
110 \LWR@loadafter{everypage}  
111 \LWR@loadafter{extramarks}  
112 \LWR@loadafter{fancyhdr}  
113 \LWR@loadafter{floatrow}
```

```
114 \LWR@loadafter{float}
115 \LWR@loadafter{floatflt}
116 \LWR@loadafter{ftnright}
117 \LWR@loadafter{geometry}
118 \LWR@loadafter{glossaries}
119 % \LWR@loadafter{graphics}%
120 % \LWR@loadafter{graphicx}%
121 \LWR@loadafter{hyperref}
122 \LWR@loadafter{indentfirst}
123 \LWR@loadafter{keyfloat}
124 \LWR@loadafter{layout}
125 \LWR@loadafter{letterspace}
126 \LWR@loadafter{letterine}
127 \LWR@loadafter{lips}
128 \LWR@loadafter{listings}
129 \LWR@loadafter{longtable}
130 \LWR@loadafter{lscape}
131 \LWR@loadafter{ltcaption}
132 \LWR@loadafter{marginfix}
133 \LWR@loadafter{marginnote}
134 \LWR@loadafter{mcaption}
135 \LWR@loadafter{mdframed}
136 \LWR@loadafter{microtype}
137 \LWR@loadafter{mparhack}
138 %\LWR@loadafter{multicol}%
139 \LWR@loadafter{multirow}
140 \LWR@loadafter{nameref}
141 \LWR@loadafter{needspace}
142 \LWR@loadafter{newtxmath}
143 \LWR@loadafter{nextpage}
144 \LWR@loadafter{nowidow}
145 \LWR@loadafter{ntheorem}
146 \LWR@loadafter{pagenote}
147 \LWR@loadafter{parskip}
148 \LWR@loadafter{placeins}
149 \LWR@loadafter{ragged2e}
150 \LWR@loadafter{rotating}
151 \LWR@loadafter{setspace}
152 \LWR@loadafter{showidx}
153 \LWR@loadafter{showkeys}
154 \LWR@loadafter{sidecap}
155 \LWR@loadafter{sidenotes}
156 \LWR@loadafter{soul}
157 \LWR@loadafter{subfig}
158 \LWR@loadafter{tabularx}
159 \LWR@loadafter{tabulary}
160 \LWR@loadafter{textpos}
161 \LWR@loadafter{theorem}
162 \LWR@loadafter{threeparttable}
163 \LWR@loadafter{tikz}
```

```

164 \LWR@loadaft{titleps}
165 \LWR@loadaft{titlesec}
166 \LWR@loadaft{titletoc}
167 \LWR@loadaft{tocloft}
168 \LWR@loadaft{trivfloat}
169 \LWR@loadaft{ulem}
170 \LWR@loadaft{varioref}
171 \LWR@loadaft{verse}
172 \LWR@loadaft{wallpaper}
173 \LWR@loadaft{wrapfig}
174 \LWR@loadaft{xcolor}
175 \LWR@loadaft{xfrac}

```

18 Required packages

These packages are automatically loaded by lwarf 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: 176 `\begin{warpall}`

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

Detects X_ET_EX and LuaL^AT_EX:

```

177 \RequirePackage{iftex}
178 \newif\ifxetexorluatex
179 \ifXeTeX
180   \xetexorluatexttrue
181 \else
182   \ifLuaTeX
183     \xetexorluatexttrue
184   \else
185     \xetexorluatextfalse
186   \fi
187 \fi
188 \end{warpall}

```

for HTML output: 189 `\begin{warpHTML}`

```
190 \ifxetexorluatex
191 % ^~A      \usepackage[no-math]{fontspec}
```

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

```
192 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
193 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
194 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
195 \else
```

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

Pkg `microtype`

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

```
196 \RequirePackage{microtype}
197
198 \microtypesetup{
199   protrusion=false,
200   expansion=false,
201   tracking=false,
202   kerning=false,
203   spacing=false}
204
205 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}

206 \fi
207 \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: 208 `\begin{warpHTML}`
209 `\RequirePackage[paperheight=190in,paperwidth=20in,%`

```
210 left=2in,right=12in,%
211 top=1in,bottom=1in,%
212 ]{geometry}
213 \twosidefalse
214 \mparswitchfalse
215 \end{warpHTML}
```

for HTML & PRINT: 216 \begin{warpall}

Pkg **xparse**
 $\text{\LaTeX}3$ command argument parsing
217 \RequirePackage{xparse}
218 \end{warpall}

for HTML output: 219 \begin{warpHTML}

Pkg **expl3**
 $\text{\LaTeX}3$ programming
220 \RequirePackage{expl3}

Pkg **getttitlestring**

Used to emulate \nameref.

221 \RequirePackage{getttitlestring}

Pkg **everyhook**

everyhook is used to patch paragraph handling.

222 \RequirePackage{everyhook}
223 \end{warpHTML}

for HTML & PRINT: 224 \begin{warpall}

Pkg **fancyvrb**
 Used for Verbatim, verse.
225 \RequirePackage{fancyvrb}
226 \end{warpall}

for HTML output: 227 \begin{warpHTML}

```
Pkg  xifthen
     228 \RequirePackage{xifthen}

Pkg  xstring
     229 \RequirePackage{xstring}

Pkg  makeidx
     230 \RequirePackage{makeidx}
     231 \makeindex

Pkg  calc
     232 \RequirePackage{calc}

Pkg  refcount
     233 \RequirePackage{refcount}

Pkg  newfloat
     234 \RequirePackage{newfloat}

Pkg  caption
     235 \RequirePackage{caption}

Pkg  enumitem
     enumitem is patched to support \newlist with HTML.
     236 \RequirePackage{enumitem}
     237 \setlist[itemize]{leftmargin=0em}
     238 \setlist[enumerate]{leftmargin=0em}
     239 \setlist[description]{leftmargin=0em}

     240 \end{warpHTML}

for HTML & PRINT: 241 \begin{warpall}

Pkg  titling
     Used for \maketitle and the title page. See section 46.
     242 \RequirePackage{titling}
```

```
243 \end{warpall}
```

for HTML output: 244 \begin{warpHTML}

Pkg **zref**

Used for cross-references.

```
245 \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.

```
246 \PassOptionsToPackage{leqno,fleqn}{amsmath}
```

```
247 \RequirePackage{amsmath}
```

Pkg **environ**

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

```
248 \RequirePackage{environ}
```

Pkg **titleps**

Used to place an HTML comment into the footer of a page below the footnotes.
This comment is used for `lateximage` environments, including math.

The `nopatches` option prevents `titleps` from trying to patch sectioning commands.

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

```
249 \RequirePackage[nopatches]{titleps}
```

`\pagestyle {<style>}`

```
250 \let\LWR@origpagestyle\pagestyle  
251 \renewcommand*\{\pagestyle}[1]{}{}
```

`\thispagestyle {<style>}`

```
252 \let\LWR@origthispagestyle\thispagestyle  
253 \renewcommand*\{\thispagestyle}[1]{}{}
```

```
\pagenumbering {⟨commands⟩}

254 \let\LWR@origpagenumbering\pagenumbering
255 \renewcommand*{\pagenumbering}[1]{}
```

Pkg **xfrac**

Patched for HTML use. See section 165.

```
256 \RequirePackage{xfrac}
```

Used to convert lengths for image width/height options.

```
257 \RequirePackage{printlen}
```

```
258 \end{warpHTML}
```

19 Loading packages

for HTML output: 259 \begin{warpHTML}

Remember the original \RequirePackage:

```
260 \let\LWR@origRequirePackage\RequirePackage
```

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

```
261 \newcommand*{\LWR@requirepackagenames}{}%
```

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

Note that argument 4 is passed directly to \StrBetween.

```
262 \newcommand*{\LWR@findword}[3][,]{%
263     \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}%
264 }
```

\LWR@lookforpackagename {⟨index⟩} If this is a package name, re-direct it to the lwarp version by renaming it lwarp- followed by the original name.

```
265 \newcommand*{\LWR@lookforpackagename}[1]{
```

Find the n'th package name from the list:

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

See if the package name was found:

```
267 \IfStrEq{\LWR@strresult}{}  
268 {}% no filename  
269 {}% yes filename
```

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

```
270 \IfFileExists{l warp-\LWR@strresult.sty}  
271 {}% latex_html_ file found  
272 \StrSubstitute  
273 {\LWR@requirepackagenames}  
274 {\LWR@strresult}  
275 {l warp-\LWR@strresult}[\LWR@requirepackagenames]  
276 }  
277 {}% no latex_html_* file  
278 {}% yes filename  
279 }
```

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

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

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

Redirect up to nine names:

```
281 \renewcommand*{\LWR@requirepackagenames}{#2}  
282 \LWR@lookforpackagename{1}  
283 \LWR@lookforpackagename{2}  
284 \LWR@lookforpackagename{3}  
285 \LWR@lookforpackagename{4}  
286 \LWR@lookforpackagename{5}  
287 \LWR@lookforpackagename{6}  
288 \LWR@lookforpackagename{7}  
289 \LWR@lookforpackagename{8}  
290 \LWR@lookforpackagename{9}
```

\RequirePackage depending on the options and version:

```
291 \IfValueTF{#1}  
292 {}% options given  
293 \IfValueTF{#3}% version given?
```

```

294 {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames} [#3]}
295 {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}
296 }
297 {%
298 \IfValueTF{#3}{%
299 {\LWR@origRequirePackage{\LWR@requirepackagenames} [#3]}
300 {\LWR@origRequirePackage{\LWR@requirepackagenames}}
301 }
302 }
303 \let\usepackage\RequirePackage

```

\LWR@ProvidesPackagePass {*pkgnname*} [*version*]

Uses the original package, including options.

```

304 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{%
305 \PackageInfo{lwarf}{Using package '#1' and adding lwarf modifications, including options,}%
306 \IfValueTF{#2}{%
307 {\ProvidesPackage{lwerp-#1} [#2]}%
308 {\ProvidesPackage{lwerp-#1}}%
309 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}%
310 \ProcessOptions\relax
311 }%
312 \IfValueTF{#2}{%
313 {\LWR@origRequirePackage{#1} [#2]}%
314 {\LWR@origRequirePackage{#1}}%
315 }

```

\LWR@ProvidesPackageDrop {*pkgnname*} [*version*]

Ignores the original package and uses lwarf's version instead. Drops/discards all options.

```

316 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{%
317 \PackageInfo{lwarf}{Replacing package '#1' with the lwarf version, discarding options,}%
318 \IfValueTF{#2}{%
319 {\ProvidesPackage{lwerp-#1} [#2]}%
320 {\ProvidesPackage{lwerp-#1}}%
321 \DeclareOption*{}%
322 \ProcessOptions\relax
323 }%
324 \end{warpHTML}

```

20 Copying a file

for HTML output: 325 \begin{warpHTML}

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

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

```
326 \newcommand*\LWR@copyfile[2]{%
327 \newwrite\copyfile % open the file to write to
328 \immediate\openout\copyfile=\#2
329 \newread\file % open the file to read from
330 \openin\file=\#1
331 \begingroup\endlinechar=-1
332 \makeatletter
333 \loop\unless\ifeof\file
334 \read\file to\fileline % Read one line and store it into \fileline
335 % \fileline\par % print the content into the pdf
336 % print the content:
337 \immediate\write\copyfile{\unexpanded\expandafter{\fileline}}%
338 \repeat
339 \closeout\copyfile
340 \endgroup
341 }

342 \end{warpHTML}
```

21 Debugging messages

343 \begin{warpall}

Bool LWR@tracinglwarp True if tracing is turned on.

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

\tracinglwarp Turns on the debug tracing messages.

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

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

```
346 \newcommand{\LWR@traceinfo}[1]{%
347 \ifbool{LWR@tracinglwarp}{%
348 {}}
```

```

349 \typeout{*** lwarf: #1}%
350 % \PackageInfo{lwarf}{#1 : }%
351 }%
352 {}%
353 }

```

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

```

354 \newbool{HTMLDebugComments}
355 \boolfalse{HTMLDebugComments}

```

22 HTML-conversion output modifications

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

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

```

356 \newbool{FormatEPUB}
357 \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.

```

358 \newbool{FormatWordProcessor}
359 \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.)

```

360 \newbool{HTMLMarkFloats}
361 \booltrue{HTMLMarkFloats}

```

```

362 \end{warpall}

```

23 Remembering original formatting macros

for HTML output: 363 \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.

```
364 \let\LWR@origtextit{textit}
365 \let\LWR@origtextbf{textbf}
366 \let\LWR@origtexttt{texttt}
367 \let\LWR@origtextsc{textsc}
368 \let\LWR@origtextsf{textsf}
369 \let\LWR@origtextrm{textrm}
370 \let\LWR@origbfseries{bfseries}
371 \let\LWR@origrmfamily{rmfamily}
372 \let\LWR@origttfamily{ttfamily}
373 \let\LWR@orignormalfont{normalfont}
374
375 \let\LWR@origraggedright{raggedright}
376 \let\LWR@origonecolumn{onecolumn}
377
378 \let\LWR@origtextsuperscript{textsuperscript}
379 \let\LWR@origtextsubscript{textsubscript}
380
381 \let\LWR@origscriptsize{scriptsize}
382
383 \let\LWR@orignewpage{newpage}
384
385 \let\LWR@origminipage{minipage}
386 \let\LWR@origendminipage{endminipage}
387
388 \let\LWR@orignewline{newline}
389
390 \let\LWR@origitem{item}
391
392 \let\LWR@origpar{par}
393
394
395 \let\LWR@origfootnote{footnote}
396 \let\LWR@orig@mpfootnotetext{@mpfootnotetext}
397
398 \let\LWR@origclearpage{clearpage}
399 \let\clearpage{relax}
400 \let\cleardoublepage{relax}

401 \end{warpHTML}
```

24 Operating-System portability

lwarp tries to detect which operating system is being used.

Prog MS-Windows If MS-WINDOWS is not correctly detected, use the command \warpOSwindows in the document preamble after lwarp is loaded. This modifies the operating-system path separator used by lwarp.

for HTML & PRINT: 402 \begin{warpall}

24.1 Common portability code

Bool usingOSwindows Set if \warpOSwindows.

403 \newbool{usingOSwindows}
404 \boolfalse{usingOSwindows}

24.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

405 \newcommand*{\OSPathSymbol}{/}

24.3 MS-Windows

For MS-Windows:

\warpOSwindows Set defaults for the MS-Windows operating system. lwarp attempts to auto-detect the operating system, so \warpOSwindows may not be necessary.

406 \newcommand*{\warpOSwindows}{
407 {
408 \booltrue{usingOSwindows}
409 \renewcommand*{\OSPathSymbol}{\@backslashchar}
410 }}

Test for windows during compile. The user may also specify \warpOSwindows later in case this test fails.

411 \ifwindows
412 \warpOSwindows
413 \fi

```
414 \end{warpall}
```

25 Stacks

for HTML output: 415 \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.

25.1 Assigning depths

initial depths for empty stack entries:

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

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

```
417 \newcommand*\{\LWR@depthfinished\}{-4}
418 \newcommand*\{\LWR@depthpart\}{-1}
419 \newcommand*\{\LWR@depthchapter\}{0}
420 \newcommand*\{\LWR@depthsection\}{1}
421 \newcommand*\{\LWR@depthsubsection\}{2}
422 \newcommand*\{\LWR@depthsubsubsection\}{3}
423 \newcommand*\{\LWR@depthparagraph\}{4}
424 \newcommand*\{\LWR@depth subparagraph\}{5}
```

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

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

used by `\item`:

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

25.2 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.

```
427 \newcommand*{\LWR@closeone}{}% top of the stack
428 \newcommand*{\LWR@closetwo}{}%
429 \newcommand*{\LWR@closethree}{}%
430 \newcommand*{\LWR@closefour}{}%
431 \newcommand*{\LWR@closefive}{}%
432 \newcommand*{\LWR@closesix}{}%
433 \newcommand*{\LWR@closeseven}{}%
434 \newcommand*{\LWR@closeeight}{}%
435 \newcommand*{\LWR@closenine}{}%
436 \newcommand*{\LWR@closeten}{}%
437 \newcommand*{\LWR@closeeleven}{}%
438 \newcommand*{\LWR@closetwelve}{}%
```

25.3 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}
```

```
439 \newcommand*{\LWR@closedepthonne}{}% top of the stack
440 \newcommand*{\LWR@closedepthtwo}{}{\LWR@depthnone}
441 \newcommand*{\LWR@closedepththree}{}{\LWR@depthnone}
442 \newcommand*{\LWR@closedepthfour}{}{\LWR@depthnone}
443 \newcommand*{\LWR@closedepthfive}{}{\LWR@depthnone}
444 \newcommand*{\LWR@closedepthsix}{}{\LWR@depthnone}
445 \newcommand*{\LWR@closedepthseven}{}{\LWR@depthnone}
446 \newcommand*{\LWR@closedeptheight}{}{\LWR@depthnone}
447 \newcommand*{\LWR@closedepthnine}{}{\LWR@depthnone}
448 \newcommand*{\LWR@closedepthten}{}{\LWR@depthnone}
```

```
449 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
450 \newcommand*{\LWR@closedephtwelve}{\LWR@depthnone}
```

25.4 Pushing and popping the stack

\pushclose {<action>} {<depth>}

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

```
451 \NewDocumentCommand{\pushclose}{m m}
452 {
453 \let\LWR@closetwelve\LWR@closeeleven
454 \let\LWR@closeeleven\LWR@closeten
455 \let\LWR@closeten\LWR@closenine
456 \let\LWR@closenine\LWR@closeeight
457 \let\LWR@closeeight\LWR@closeseven
458 \let\LWR@closeseven\LWR@closesix
459 \let\LWR@closesix\LWR@closefive
460 \let\LWR@closefive\LWR@closefour
461 \let\LWR@closefour\LWR@closethree
462 \let\LWR@closethree\LWR@closetwo
463 \let\LWR@closetwo\LWR@closeone
464 \let\LWR@closeone#
465 \let\LWR@closedephtwelve\LWR@closedeptheleven
466 \let\LWR@closedepthelevel\LWR@closedepthten
467 \let\LWR@closedepthten\LWR@closedepthnine
468 \let\LWR@closedepthnine\LWR@closedeptheight
469 \let\LWR@closedeptheight\LWR@closedepthseven
470 \let\LWR@closedepthseven\LWR@closedepthsix
471 \let\LWR@closedepthsix\LWR@closedepthfive
472 \let\LWR@closedepthfive\LWR@closedepthfour
473 \let\LWR@closedepthfour\LWR@closedepththree
474 \let\LWR@closedepththree\LWR@closedepthtwo
475 \let\LWR@closedepthtwo\LWR@closedepthonne
476 \let\LWR@closedepthonne#
477 }
```

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

```
478 \newcommand*{\popclose}
479 {
480 \let\LWR@closeone\LWR@closetwo
481 \let\LWR@closetwo\LWR@closethree
482 \let\LWR@closethree\LWR@closefour
483 \let\LWR@closefour\LWR@closefive
484 \let\LWR@closefive\LWR@closesix
485 \let\LWR@closesix\LWR@closeseven
```

```

486 \let\LWR@closeseven\LWR@closeeight
487 \let\LWR@closeeight\LWR@closenine
488 \let\LWR@closenine\LWR@closeten
489 \let\LWR@closeten\LWR@closeeleven
490 \let\LWR@closeeleven\LWR@closetwelve
491 \let\LWR@closedepthon\LWR@closedepthtwo
492 \let\LWR@closedepthtwo\LWR@closedepththree
493 \let\LWR@closedepththree\LWR@closedepthfour
494 \let\LWR@closedepthfour\LWR@closedepthfive
495 \let\LWR@closedepthfive\LWR@closedepthsix
496 \let\LWR@closedepthsix\LWR@closedepthseven
497 \let\LWR@closedepthseven\LWR@closedeptheight
498 \let\LWR@closedeptheight\LWR@closedepthnine
499 \let\LWR@closedepthnine\LWR@closedepthten
500 \let\LWR@closedepthten\LWR@closedeptheleven
501 \let\LWR@closedeptheleven\LWR@closedepthtwelve
502 }

503 \end{warpHTML}

```

26 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}{\relax}
```

for HTML output: 504 `\begin{warpHTML}`

```

\LWR@setexparray {<name>} {<index>} {<contents>}
505 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
506 \expandafter\edef\csname #1#2\endcsname{\expandonce#3}%
507 }

\LWR@getexparray {<name>} {<index>}
508 \newcommand*{\LWR@getexparray}[2]{\csuse{#1#2}}
509 \end{warpHTML}

```

27 HTML entities

```

for HTML output: 510 \begin{warpHTML}

    HTML entites and HTML Unicode entities:

    511 \let\LWR@origampersand\&

    \HTMLentity {<entitytag>}

    512 \newcommand*{\HTMLentity}[1]{\LWR@origampersand#1;}

    \HTMLunicode {<hex_unicode>}

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


    \&

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


    \textless
    \textgreater
    515 \let\LWR@origtextless\textless
    516 \renewcommand*{\textless}{\HTMLentity{lt}}
    517
    518 \let\LWR@origtextgreater\textgreater
    519 \renewcommand*{\textgreater}{\HTMLentity{gt}}


    520 \end{warpHTML}

```

28 HTML filename generation

The filename of the homepage is set to `\HomeHTMLFileName.html`. The filenames of additional sections start with `\HTMLfiveName`, to which is appended a section number or a simplified section name, depending on `FileSectionNames`.

```

for HTML & PRINT: 521 \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.

    522 \providecommand*{\BaseJobname}{\jobname}

```

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

```
523 \providecommand*\HTMLfiveName{}{}
```

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

```
524 \providecommand*\HomeHTMLFileName{\BaseJobname}
```

\SetHTMLFileName {*number*}

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

```
525 \newcommand*\SetHTMLFileName[1]{%
526 \setcounter{LWR@htmlfilename}{#1}%
527 }
```

Bool FileSectionNames Selects how to create HTML file names.

Defaults to use section names in the filenames.

```
528 \newbool{FileSectionNames}
529 \booltrue{FileSectionNames}
```

```
530 \end{warpall}
```

for HTML output: 531 \begin{warpHTML}

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

```
532 \newcounter{LWR@htmlfilename}
533 \setcounter{LWR@htmlfilename}{0}
```

\LWR@htmlsectionfilename {*htmlfilename or name*}

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

```
534 \newcommand*\LWR@htmlsectionfilename[1]{%
535 \LWR@traceinfo{\LWR@htmlsectionfilename A}%
}
```

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

```

536 \LWR@traceinfo{about to assign temp}%
537 \edef\LWR@tempone{\#1}%
538 \LWR@traceinfo{about to compare with ??}%
539 \ifthenelse{\equal{\LWR@tempone}{??}}{%
540 }{%
541 \LWR@traceinfo{found ??}%
542 }{%
543 \LWR@traceinfo{not found ??}%
544 }{%
545 \LWR@traceinfo{about to compare with zero or empty}%
546 \ifthenelse{%
547 \equal{\LWR@tempone}{0}}{%
548 \OR \equal{\LWR@tempone}{}}{%
549 \OR \equal{\LWR@tempone}{??}}{%
550 }{%
551 }{%
552 \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFileName.html}%
553 \HomeHTMLFileName.html}%
554 }%

```

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`:

```

555 {%
556 \LWR@traceinfo{LWR@htmlsectionfilename C \LWR@tempone}%
557 \ifthenelse{%
558 \equal{\HTMLfiveName}{}}{\AND \equal{\LWR@tempone}{Index}}{\OR \equal{\LWR@tempone}{index}}{%
559 }{%
560 }{%
561 \LWR@traceinfo{prefixing the index name with an underscore.}%
562 \_#1.html}%

```

Otherwise, create a filename with the chosen prefix:

```

563 {\HTMLfiveName#1.html}%
564 }{%
565 \LWR@traceinfo{LWR@htmlsectionfilename Z}%
566 }

```

`\LWR@htmlrefsectionfilename {<label>}`

Prints the filename for the given label

```

567 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
568 \LWR@traceinfo{LWR@htmlrefsectionfilename A: !#1!}%
569 \LWR@htmlsectionfilename{\LWR@htmlfileref{\#1}}%
570 \LWR@traceinfo{LWR@htmlrefsectionfilename B}%
571 }

```

572 \end{warpHTML}

29 Homepage link

for HTML output: 573 \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.

```
574 \newcommand*\{\LinkHome\}{%
575 \LWR@subhyperrefclass{%
576 \HomeHTMLFileName.html}%
577 {Home}\{linkhome\}%
578 }
```

\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.

```
579 \newcommand*\{\LWR@topnavigation\}{%
580 \LWR@htmlelementclassline{nav}\{topnavigation\}\{\LinkHome\}%
581 }
```

\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.

```
582 \newcommand*\{\LWR@botnavigation\}{%
583 \LWR@htmlelementclassline{nav}\{botnavigation\}\{\LinkHome\}%
584 }
```

585 \end{warpHTML}

30 \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: 586 \begin{warpHTML}

\PrintStack Prints the closeddepth stack.

```

587 \newcommand*\PrintStack|{
588 \LWR@startpars
589 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
590 \LWR@closedeptfour{} \LWR@closedeptfive{} \LWR@closedepthsix{}
591 \LWR@closedeptseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
592 \LWR@closedeptten{} \LWR@closedepttheleven{} \LWR@closedepthtwelve{}
593 }

594 \end{warpHTML}

```

31 Closing stack levels

for HTML output: 595 \begin{warpHTML}

Close one nested level:

```

596 \newcommand*\LWR@closeoneprevious}{%
597
598 \LWR@closeone{}
599
600 \popclose{}
601 }

```

\LWR@closeprevious {*depth*} Close everything up to the given depth:

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

Close any pending paragraph:

```
603 \LWR@stoppars
```

Close anything nested deeper than the desired depth:

```

604 \whiledo{\not\(\LWR@closedepthone<\#1\)}{\LWR@closeoneprevious}
605 }

```

```
606 \end{warpHTML}
```

32 Forcing a new PDF page

for HTML output: 607 \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.

```
608 \newcommand{\LWR@forcenewpage}{%
609 \LWR@stoppars\LWR@orignewpage\LWR@startpars%
610 }%
611 \end{warpHTML}
```

33 HTML tags, spans, divs, elements

for HTML output: 612 \begin{warpHTML}

33.1 Mapping L^AT_EX Sections to HTML Sections

```
613 \newcommand*{\LWR@tagpart}{h2}
614 \newcommand*{\LWR@tagpartend}{/h2}
615 \newcommand*{\LWR@tagchapter}{h3}
616 \newcommand*{\LWR@tagchapterend}{/h3}
617 \newcommand*{\LWR@tagsection}{h4}
618 \newcommand*{\LWR@tagsectionend}{/h4}
619 \newcommand*{\LWR@tagsubsection}{h5}
620 \newcommand*{\LWR@tagsubsectionend}{/h5}
621 \newcommand*{\LWR@tagsubsubsection}{h6}
622 \newcommand*{\LWR@tagsubsubsubsection}{/h6}
623 \newcommand*{\LWR@tagparagraph}{span class="paragraph"{}}
624 \newcommand*{\LWR@tagparagraphend}{/span}
625 \newcommand*{\LWR@tagsubparagraph}{span class="subparagraph"{}}
626 \newcommand*{\LWR@tagsubparagraphend}{/span}
627
628 \newcommand*{\LWR@tagregularparagraph}{p}
```

33.2 HTML tags

\LWR@htmlltagc {*tag*} Break ligatures and use upright apostrophes in HTML tags.

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

```
629
630 \newcommand*{\LWR@htmlltagc}[1]{%
631 {%
632 \protect\LWR@origttfamily%
```

```

633 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
634 }%
635 }

```

Env `LWR@nestspan` Disable `minipage`, `\parbox` 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 `/span` or an extra `<p>` may appear.

```

636 \newenvironment*{LWR@nestspan}
637 {%
638 \addtocounter{LWR@spandepth}{1}%
639 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{}{%
640 }%
641 {\addtocounter{LWR@spandepth}{-1}}}

```

`\LWR@htmlspan` `{<tag>}` `{<text>}`

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

```

642 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
643 \LWR@ensuredoingapar%
644 \LWR@htmntagc{#1}%
645 \begin{LWR@nestspan}%
646 #2%
647 \LWR@htmntagc{/#1}%
648 \end{LWR@nestspan}%
649 }

```

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

```

650 \NewDocumentCommand{\LWR@htmlspanclass}{m o +m}{%
651 \LWR@ensuredoingapar%
652 \LWR@subhtmlelementclass{span}{#1}[#2]%
653 \begin{LWR@nestspan}%
654 #3%
655 \LWR@htmntagc{/span}%
656 \end{LWR@nestspan}%
657 }

```

`\LWR@htmntag` `{<tag>}`

Print an HTML tag: `<tag>`

```

658 \newcommand*{\LWR@htmntagb}[1]{%
659 \LWR@htmtagc{#1}%
660 \endgroup%
661 }%
662 \newcommand*{\LWR@htmntag}{%
664 \begingroup\catcode`\_=12%
665 \LWR@htmtagb%
666 }

```

33.3 Block tags and comments

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

```

\LWR@htmlopencomment
\LWR@htmlclosecomment
667 \newcommand*{\LWR@htmlopencomment}{%
668 {\LWR@origttfamily\LWR@origtextless{}!{-}{-}}%
669 }%
670 \newcommand*{\LWR@htmlclosecomment}{%
672 {\LWR@origttfamily{-}{-}\LWR@origtextgreater{}{}}%
673 }

\LWR@htmlcomment {<comment>}
674 \newcommand{\LWR@htmlcomment}[1]{%
675 \LWR@htmlopencomment{}%
676 {%
677 \LWR@origttfamily% break ligatures
678 #1%
679 }%
680 \LWR@htmlclosecomment{}}

\LWR@htmlblockcomment {<comment>}
681 \newcommand{\LWR@htmlblockcommentb}[1]
682 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
683
684 \newcommand{\LWR@htmlblockcomment}
685 {%
686 \begingroup\catcode`\_=12%
687 \LWR@htmlblockcommentb%
688 }

```

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

```
689 \newcommand*\LWR@htmlblocktag[1]{%
690 \LWR@stoppars%
691 \LWR@htmlltag{\#1}%
692 \LWR@startpars%
693 }
```

33.4 Div class and element class

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

Factored and reused in several places.

```
694 \NewDocumentCommand{\LWR@subhtmlelementclass}{m m o}{%
695 \IfValueTF{\#3}{%
696 {%
697 \ifthenelse{\equal{\#3}{}}{%
698 {\LWR@htmlltag{\#1 class="#2"}% empty option
699 {\LWR@htmlltag{\#1 class="#2" style="#3"}% non-empty option
700 }% option
701 {\LWR@htmlltag{\#1 class="#2"}% no option
702 }}
```

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

```
703 \NewDocumentCommand{\LWR@htmlelementclass}{m m o}{%
704 \LWR@stoppars%
705 \LWR@subhtmlelementclass{\#1}{\#2}{\#3}%
706 \LWR@startpars%
707 }
```

```
\LWR@htmlelementclassend  {\langle element\rangle} {\langle class\rangle}
```

```
708 \newcommand*\LWR@htmlelementclassend[2]{%
709 \LWR@stoppars%
710 \LWR@htmlltag{/#1}%
711 \ifbool{HTMLDebugComments}{%
712 \LWR@htmllcomment{End of #1 '#2'}%
713 }{}%
714 \LWR@startpars%
715 }
```

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

```

716 \NewDocumentCommand{\LWR@htmldivclass}{m o}{%
717 \LWR@htmlelementclass{div}{#1}[#2]%
718 }

\LWR@htmldivclassend {⟨class⟩}

719 \newcommand*{\LWR@htmldivclassend}[1]{%
720 \LWR@htmlelementclassend{div}{#1}%
721 }

```

33.5 Single-line elements

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

```

\LWR@htmlelementclassline {⟨element⟩} {⟨class⟩} [⟨style⟩] {⟨text⟩}

722 \NewDocumentCommand{\LWR@htmlelementclassline}{m m o +m}{%
723 \LWR@stoppars
724 \LWR@subhtmlelementclass{#1}{#2}{#3}%
725 #4%
726 \LWR@htmlltag{/#1}
727 \LWR@startpars
728 }

```

33.6 HTML5 semantic elements

```

\LWR@htmlelement {⟨element⟩}

729 \newcommand*{\LWR@htmlelement}[1]{%
730 \LWR@htmlblocktag{#1}%
731 }

\LWR@htmlelementend {⟨element⟩}

732 \newcommand*{\LWR@htmlelementend}[1]{%
733 \LWR@stoppars
734 \LWR@htmlltag{/#1}
735 \LWR@startpars
736 }
737
738 \end{warpHTML}

```

33.7 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 66.

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

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

for HTML output: 739 \begin{warpHTML}
 740 \NewDocumentEnvironment{BlockClass}{m o}
 741 {
 742 \LWR@htmldivclass{#1}{#2}
 743 }
 744 {
 745 \LWR@htmldivclassend{#1}
 746 }
 747 \end{warpHTML}

for PRINT output: 748 \begin{warpprint}
 749 \NewDocumentEnvironment{BlockClass}{m o}{ }{ }
 750 \end{warpprint}

\BlockClassSingle *{<class>} {<text>}* A single-line <div>, without a paragraph tag for the line of text.

for HTML output: 751 \begin{warpHTML}
 752 \newcommand{\BlockClassSingle}[2]{%
 753 \LWR@htmlelementclassline{div}{#1}{#2}}%
 754 }
 755 \end{warpHTML}

for PRINT output: 756 \begin{warpprint}
 757 \newcommand{\BlockClassSingle}[2]{#2}
 758 \end{warpprint}

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

for HTML output: 759 \begin{warpHTML}
 760 \NewDocumentCommand{\InlineClass}{m o +m}{%
 761 \LWR@htmlspanclass{#1}{#2}{#3}}%
 762 }
 763 \end{warpHTML}

for PRINT output: 764 \begin{warpprint}
 765 \NewDocumentCommand{\InlineClass}{m o +m}{#3}

```
766 \end{warpprint}
```

33.8 Closing HTML tags

for HTML output: 767 \begin{warpHTML}

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

```
768 \newcommand*\{\LWR@printclosepart\}
769     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
770 \newcommand*\{\LWR@printclosechapter\}
771     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
772 \newcommand*\{\LWR@printclosesection\}
773     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
774 \newcommand*\{\LWR@printclosesubsection\}
775     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
776 \newcommand*\{\LWR@printclosesubsubsection\}
777     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
778 \newcommand*\{\LWR@printcloseparagraph\}
779     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
780 \newcommand*\{\LWR@printclosesubparagraph\}
781     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}
```

Lists require closing HTML tags:

```
782 \newcommand*\{\LWR@printcloselistitem\}
783     {\LWR@htmllistitem{/li}}
784 \newcommand*\{\LWR@printclosedescitem\}
785     {\LWR@htmldescitem{/dd}}
786 \newcommand*\{\LWR@printcloseitemize\}
787     {\LWR@htmllistitem{/ul}}
788 \newcommand*\{\LWR@printcloseenumerate\}
789     {\LWR@htmllistitem{/ol}}
790 \newcommand*\{\LWR@printclosedescription\}
791     {\LWR@htmldescription{/dl}}
```

```
792 \end{warpHTML}
```

34 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 35 has high-level commands which allow paragraph-tag generation to start/stop. Even when allowed (`\LWR@doingstartpars`), tags are not generated until a L^AT_EX 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: 793 `\begin{warpHTML}`

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

```
794 \newcounter{\LWR@spandepth}
795 \setcounter{\LWR@spandepth}{0}
```

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

```
796 \newbool{\LWR@doingstartpars}
797 \boolfalse{\LWR@doingstartpars}
```

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

```
798 \newbool{\LWR@doingapar}
799 \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:

```
800 \newcommand*{\LWR@ensuredoingapar}{%
801 \ifbool{\LWR@doingstartpars}{%
802 {\global\booltrue{\LWR@doingapar}}}{%
803 {}}%
804 }
```

`\LWR@openparagraph`

```
805 \newcommand*{\LWR@openparagraph}{%
806 {}%
```

See if paragraph handling is enabled:

```
807 \ifbool{\LWR@doingstartpars}{%
808 {}% handling pars}
```

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

```

809 \ifthenelse{%
810 \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
811 \cnttest{\value{LWR@spandepth}}{>}{0}%
812 }% 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 ``.

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

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

```

814 {% yes nest par tags
815 \LWR@htmllagc{\LWR@tagregularparagraph}%

```

Manually indent item list labels to avoid left margin intrustion:

`LATEX` default list environments use `\@itemdepth` and `\@enumdepth`, but `lwarp` uses the `enumitem` package, which uses `\@listdepth`.

See if are nested inside an item list:

```

816 \ifnumcomp{\@listdepth}{>}{0}%
817 {}%

```

If so, leave some horizontal room in the `LATEX` PDF output for list labels:

```

818 \LWR@orighspace{1in}%
819 }{}%

```

Now have started a paragraph.

```
820 \global\booltrue{\LWR@doingapar}%
```

At the endof each paragraph, generate closing tag and do regular `/par` stuff. (Attempting to use the `everyhook cr` hook for `\LWR@closeparagraph` does not work well.)

```

821 \let\par\LWR@closeparagraph%
822 }% end of yes nest par tags
823 }% end of handling pars
824 {}% not handling pars
825 }

```

```
\LWR@closeparagraph
```

```
826 \newcommand*{\LWR@closeparagraph}{%
827 {%
```

See if paragraph handling is enabled:

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

If currently in paragraph mode:

```
829 {%
  % handling pars
```

See if already started a `lateximage` or a ``:

```
830 \ifthenelse{%
831 \cnttest{\value{\LWR@lateximagedepth}}>}{0} \OR%
832 \cnttest{\value{\LWR@spandepth}}>}{0}%
833 }%
```

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

```
834 {%
  % no nested par tags
  \ifthenelse{%
    \cnttest{\value{\LWR@spandepth}}>}{0}%
    \AND%
    \cnttest{\value{\LWR@lateximagedepth}}=}{0}%
  }%
  \ifbool{\LWR@intabularmetadata}{}{\unskip\LWR@htmlltagc{br /}}%
}%
}%
}%
}%
844 {%
  % no nested par tags
```

If have not already started a `lateximage` or a ``:

```
845 {%
  % yes nest par tags
```

Print a closing tag:

```
846 \unskip%
847 \LWR@htmlltagc{/}\LWR@tagregularparagraph}%
```

No longer doing a paragraph:

```
848 \global\boolearn{\LWR@doingapar}{%
849 % Disable the special \env{minipage} \& \cs{hspace} interaction
850 % until a new minipage is found:
851 %   \begin{macrocode}
852 \global\boolearn{\LWR@minipagethispar}{%
```

```
853 }% end of yes nest par tags
854 }% end of handling pars
```

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

```
855 {%
  not handling pars
  \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}{%
    \ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}}%
858 {}%
859 }% not handling pars
```

Finish with regular paragraph processing

```
860 \LWR@origpar%
861 }
```

```
862 \end{warpHTML}
```

35 Paragraph start/stop handling

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

Section 34 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: 863 `\begin{warpHTML}`

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

```
864 \newcommand*\LWR@startpars{%
865 {}%
```

See if currently handling HTML paragraphs:

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

If already in paragraph mode, do nothing.

```
867 {}%
```

If not currently in paragraph mode:

```
868 {%
```

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

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

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

```
870 \let\par\LWR@closeparagraph  
871  
872 }% an intentionally blank line
```

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

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

No <par> tag yet to undo:

```
874 \global\boolfalse{\LWR@doingapar}%
875 }
```

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

```
876 \newcommand*{\LWR@stoppars}%
877 {%
```

See if currently handling HTML paragraphs:

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

if currently in an HTML paragraph:

```
879 {}%
```

Print a closing tag:

```
880 \unskip%
881 \LWR@htmllagc{/ \LWR@tagregularparagraph}%
```

No longer have an open HTML paragraph:

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

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

```
883 \global\boolfalse{\LWR@minipagethispar}%
884  
885 }% an intentionally blank line
```

If was not in an HTML paragraph:

886 {}%

See if currently allowing HTML paragraphs:

887 \ifbool{LWR@doingstartpars}{}%

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

888 {}%
 889 \ClearPreHook{par}{}%
 890 {}%

Else: do nothing

891 {}%

no longer in paragraph mode

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

no `<p>` tag to undo:

893 \global\boolfalse{LWR@doingapar}{}%
 894 {}

895 \end{warpHTML}

36 Page headers and footers

for HTML & PRINT: 896 \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.

897 \newcommand{\LWR@firstpagetop}{} % for the home page alone
 898 \newcommand{\LWR@pagetop}{} % for all other pages
 899 \newcommand{\LWR@pagebottom}{}
 900
 901 \newcommand{\LWR@setfirstpagetopb}[1]{}%
 902 \renewcommand{\LWR@firstpagetop}{#1}
 903 \catcode`_=8
 904 {}

```
\SetFirstPageTop  {\langle text and logos\rangle}

905 \newcommand{\SetFirstPageTop}{%
906 \catcode`\_=12
907 \LWR@setfirstpagetopb
908 }

909 \newcommand{\LWR@setpagetopb}[1]{%
910 \renewcommand{\LWR@pagetop}{#1}
911 \catcode`\_=8
912 }

\SetPageTop  {\langle text and logos\rangle}

913 \newcommand{\SetPageTop}{%
914 \catcode`\_=12
915 \LWR@setpagetopb
916 }

917 \newcommand{\LWR@setpagebottomb}[1]{%
918 \renewcommand{\LWR@pagebottom}{#1}
919 \catcode`\_=8
920 }

\SetPageBottom  {\langle text and logos\rangle}

921 \newcommand{\SetPageBottom}{%
922 \catcode`\_=12
923 \LWR@setpagebottomb
924 }

925 \end{warpall}
```

37 CSS

for HTML output: 926 \begin{warpHTML}

\LWR@currentcss The css filename to use. This may be changed mid-document using \NewCSS, allowing different CSS files to be used for different sections of the document.

```
927 \newcommand*{\LWR@currentcss}{lwarp.css}
```

```
\NewCSS  {\langle new-css-filename.css\rangle}
```

Assigns the CSS file to be used by the following HTML pages.

```

928 \newcommand*{\LWR@newcssb}[1]{%
929 \renewcommand*{\LWR@currentcss}{#1}
930 \catcode`\_=8
931 }
932
933 \newcommand*{\NewCSS}{
934 \catcode`\_=12
935 \LWR@newcssb
936 }
937 \end{warpHTML}

for PRINT output: 938 \begin{warpprint}
939 \newcommand*{\NewCSS}[1]{}
940 \end{warpprint}

```

38 HTML meta description and author

for HTML & PRINT: 941 \begin{warpall}

```

\HTMLauthor {\<authorname>}      The author to place into an HTML meta tag.

942 \newcommand{\theHTMLauthor}{\theauthor}
943
944 \newcommand{\HTMLauthor}[1]{\renewcommand{\theHTMLauthor}{#1}}

945 \end{warpall}

```

for HTML & PRINT: 946 \begin{warpall}

This is placed inside an HTML meta tag at the start of each file. This may be changed mid-document using \NewHTMLdescription, allowing different HTML descriptions to be used for different sections of the document.

 Do not use double quotes, and do not exceed 150 characters.

```

\NewHTMLdescription {\<New HTML meta description.>}      Assigns the HTML file's description meta
tag.

947 \newcommand{\LWR@currentHTMLdescription}{}
948
949 \newcommand{\NewHTMLdescription}[1]{%
950 \renewcommand{\LWR@currentHTMLdescription}{#1}

```

```
951 }  
952  
953 \end{warpall}
```

39 Footnotes

l warp uses native L^AT_EX footnote code, although with its own `\box` to avoid the L^AT_EX 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 handled differently.

39.1 Regular page footnotes

In HTML documents, footnotes are placed at the bottom of the web page using the L^AT_EX 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 39.4 for the implementation.

39.2 Minipage footnotes

See section 65.2 for minipage footnotes.

39.3 Titlepage thanks

See section 46.6 for titlepage footnotes.

39.4 Regular page footnote implementation

for HTML output: 954 `\begin{warpHTML}`

Patch L^AT_EX footnotes to use a new `\box` for l warp footnotes.

```
955 \newbox\LWR@footnotes
```

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

```

956 \long\def\@makefntext#1{\textsuperscript{\@thefnmark} #1}
957
958 \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.

```

959 \long\def\@footnotetext#1{%
960 \global\setbox\LWR@footnotes=\vbox{\unvbox\LWR@footnotes%
961 \protected@edef\@currentlabel{%
962 \csname p@footnote\endcsname\@thefnmark%
963 }% \@currentlabel
964 \color@begingroup%
965 \ifbool{\LWR@doingstartpars}{\LWR@htmlltagc{\LWR@tagregularparagraph}}{%
966 \@makefntext{#1}%
967 \ifbool{\LWR@doingstartpars}{\par}{\LWR@htmlltagc{/}\LWR@tagregularparagraph}}{%
968 \color@endgroup%
969 }% vbox
970 }%
971
972 \long\def\@mpfootnotetext#1{%
973 \global\setbox\@mpfootins\vbox{%
974 \unvbox\@mpfootins
975 \reset@font\footnotesize
976 \hsize\columnwidth
977 \parboxrestore
978 \protected@edef\@currentlabel
979 {\csname p@mpfootnote\endcsname\@thefnmark}%
980 \color@begingroup
981 \@makefntext{%
982 \ignorespaces#1%
983 }%

```

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

```

984 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}>0}{%
985 }%
986 {\LWR@htmlltagc{/}\LWR@tagregularparagraph}}{%
987 \color@endgroup}%
988 }

```

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

```

989 \newcommand*{\LWR@printpendingfootnotes}{%
990 \ifvoid\LWR@footnotes\else
991 \LWR@forcenewpage
992 \begin{BlockClass}{footnotes}
993 \LWR@origmedskip

```

```

994 \unvbox\LWR@footnotes
995 \setbox\LWR@footnotes=\vbox{}
996 \end{BlockClass}
997 \fi
998 }

```

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

```

999 \newcommand*\LWR@epubprintpendingfootnotes{%
1000 \ifthenelse{\boolean{FormatEPUB} \OR \boolean{FormatWordProcessor}}{%
1001 {\LWR@printpendingfootnotes}%
1002 {}%
1003 }

1004 \end{warpHTML}

```

40 Marginpars

for HTML output: 1005 \begin{warpHTML}

```

\marginpar [⟨left⟩] {⟨right⟩}

1006 \renewcommand{\marginpar}[2][]{%
1007 \LWR@htmlspanclass{marginpar}{#2}%
1008 }

1009 \end{warpHTML}

```

41 Splitting HTML files

- Files are split according to `FileDepth` and `CombineHigherDepths`.
- Filenames are sanitized by `\LWR@filenamenoblanks`.
- `\LWR@newhtmlfile` finishes an HTML page, adds a comment to tell where and how to split the file, then starts a new HTML page.

for HTML & PRINT: 1010 \begin{warpall}

`Ctr FileDepth {⟨section depth⟩}` determines how deeply to break into new HTML files, similar to

`tocdepth`. The default of -5 produces one large HTML file.

```
1011 \newcounter{FileDepth}
1012 \setcounter{FileDepth}{-5}
```

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

```
1013 \newbool{CombineHigherDepths}
1014 \booltrue{CombineHigherDepths}

1015 \end{warpall}
```

for HTML output: 1016 \begin{warpHTML}

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

```
1017 \newcommand*\{\LWR@thisfilename}\{}{}
```

`\LWR@thisnewfilename` The filename being sanitized.

```
1018 \newcommand*\{\LWR@thisnewfilename}\{}{}
```

`\LWR@filenamenoblocks` {*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.

```
1019 \newcommand*\{\LWR@filenamenoblocks}[1]{%
1020 \begingroup
```

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

```
1021 \renewcommand*\{\HTMLunicode}[1]{}
1022 \renewcommand*\{\HTMLentity}[1]{##1}
1023 \renewcommand*\{\LWR@htmlltagc}[1]{}
1024 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}
```

Ampersand becomes “and”, which is a short word and is then removed from the filename.

```
1025 \renewcommand*\{\&}{and}
1026 \renewcommand{\textit}[1]{##1}
1027 \renewcommand{\textsc}[1]{##1}
```

```

1028 \renewcommand{\textsl}[1]{##1}
1029 \renewcommand{\textbf}[1]{##1}
1030 \renewcommand{\texttt}[1]{##1}
1031 \renewcommand{\textsf}[1]{##1}
1032 \renewcommand{\textrm}[1]{##1}
1033 \renewcommand{\textsuperscript}[1]{##1}
1034 \renewcommand{\textsubscript}[1]{##1}

```

Replaces common symbols and short words with hyphens:

```

1035 \edef\LWR@thisfilename{#1}
1036 \fullexpandarg

```

Convert spaces into hyphens:

```
1037 \StrSubstitute{\LWR@thisfilename}{ }{-}[\LWR@thisfilename]
```

Convert punctuation into hyphens:

```

1038 \StrSubstitute{\LWR@thisfilename}{,}{-}[\LWR@thisfilename]
1039 \StrSubstitute{\LWR@thisfilename}{'}{-}[\LWR@thisfilename]
1040 \StrSubstitute{\LWR@thisfilename}{%}
1041 {\LWR@origampersand}{-}[\LWR@thisfilename]
1042 \StrSubstitute{\LWR@thisfilename}{+}{-}[\LWR@thisfilename]
1043 \StrSubstitute{\LWR@thisfilename}{,}{-}[\LWR@thisfilename]
1044 \StrSubstitute{\LWR@thisfilename}{/}{-}[\LWR@thisfilename]
1045 \StrSubstitute{\LWR@thisfilename}{:}{-}[\LWR@thisfilename]
1046 \StrSubstitute{\LWR@thisfilename}{;}{-}[\LWR@thisfilename]
1047 \StrSubstitute{\LWR@thisfilename}{=}{-}[\LWR@thisfilename]
1048 \StrSubstitute{\LWR@thisfilename}{?}{-}[\LWR@thisfilename]
1049 \StrSubstitute{\LWR@thisfilename}{@}{-}[\LWR@thisfilename]
1050 \StrSubstitute{\LWR@thisfilename}{"}{-}[\LWR@thisfilename]
1051 \StrSubstitute{\LWR@thisfilename}{%}
1052 {\textless}{-}[\LWR@thisfilename]
1053 \StrSubstitute{\LWR@thisfilename}{%}
1054 {\textgreater}{-}[\LWR@thisfilename]
1055 \StrSubstitute{\LWR@thisfilename}{\#}{-}[\LWR@thisfilename]
1056 \StrSubstitute{\LWR@thisfilename}{\%}{-}[\LWR@thisfilename]
1057 \StrSubstitute{\LWR@thisfilename}{\{}{-}[\LWR@thisfilename]
1058 \StrSubstitute{\LWR@thisfilename}{\}}{-}[\LWR@thisfilename]
1059 \StrSubstitute{\LWR@thisfilename}{\|}{-}[\LWR@thisfilename]
1060 \StrSubstitute{\LWR@thisfilename}{%}
1061 {\textbackslash}{-}[\LWR@thisfilename]
1062 \StrSubstitute{\LWR@thisfilename}{^}{-}[\LWR@thisfilename]
1063 \StrSubstitute{\LWR@thisfilename}{\~{}}{-}[\LWR@thisfilename]
1064 \% "~-{}" for babel
1065 \StrSubstitute{\LWR@thisfilename}{[]}{-}[\LWR@thisfilename]
1066 \StrSubstitute{\LWR@thisfilename}{\}]{-}[\LWR@thisfilename]
1067 \StrSubstitute{\LWR@thisfilename}{\'}{-}[\LWR@thisfilename]

```

Convert short words:

```

1068 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
1069 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
1070 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
1071 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
1072 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
1073 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
1074 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
1075 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
1076 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
1077 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
1078 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
1079 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
1080 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
1081 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
1082 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
1083 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
1084 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
1085 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]
```

Convert multiple hyphens:

```

1086 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
1087 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
1088 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
1089 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
1090 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
1091 %      emdash
1092 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
1093 %      endash
1094 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
1095 \endgroup
1096 }
```

`\LWR@newhtmlfile {<section name>}`

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

```
1097 \newcommand*{\LWR@newhtmlfile}[1]{
```

At the bottom of the ending file:

```

1098 \LWR@htmlelementclassend{section}{textbody}
1099
1100 \LWR@printpendingfootnotes
```

1101

No footer between files if EPUB:

```
1102 \ifbool{FormatEPUB}{}
1103 {}
1104 {
1105 \LWR@htmlelement{footer}
1106
1107 \LWR@pagebottom
1108
1109 \LWR@htmlelementend{footer}
1110 }
```

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

```
1111 \ifthenelse{\boolean{FormatEPUB} \OR \boolean{FormatWordProcessor}}{}
1112 {}
1113 {\ifnumcomp{\value{LWR@htmldocumentnumber}}{>}{0}{\LWR@botnavigation}{}}
```

End of this HTML file:

```
1114 \LWR@stopars
1115 \LWR@htmllasttag{/body}\LWR@orignewline
1116 \LWR@htmllasttag{/html}\LWR@orignewline
1117 \LWR@orignewpage
1118
1119 \addtocounter{LWR@htmldocumentnumber}{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.

```
1120 \ifbool{FileSectionNames}{%
1121 {\LWR@filenamenoblocks{\#1}}
1122 {\renewcommand*{\LWR@thisfilename}{\theLWR@htmldocumentnumber}}}
```

Include an HTML comment to instruct lwarfpmk 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.

```
1123 \LWR@htmloblockcomment{%
1124 |Start file|%
1125 \LWR@htmsectionfilename{\LWR@thisfilename}|%
1126 }
```

At the top of the starting file:

```
1127 \LWR@stoppars
1128
1129 \LWR@filestart{ -- #1}%
1130 % there is an EMdash in front of the #1
```

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

```
1131 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
1132 {}
1133 {\LWR@topnavigation}
1134
```

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

```
1135 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
1136 {}
1137 {
1138 \LWR@htmlelement{header}
1139
1140 \LWR@pagetop
1141
1142 \LWR@htmlelementend{header}
1143 }
1144
```

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

```
1145 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
1146 {}
1147 {\ifcvoid{\thetitle}{}{\LWR@printthetitle}}
1148
```

No sidetoc if formatting for an EPUB or word processor:

```
1149 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
1150 {}
1151 {\LWR@sidetoc}
1152
```

Start of the <textbody>:

```
1153 \LWR@htmlelementclass{section}{textbody}
1154
```

Keep paragraph tags disabled for now:

```
1155 \LWR@stoppars
1156
```

Track the page numbers:

```
1157 \setcounter{LWR@latestautopage}{\value{page}}
1158 }

1159 \end{warpHTML}
```

42 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 `pdflatex` is actually producing a symbol over or under a character, resulting in `pdftotext` picking up the accent symbol separately.

X_LA_TE_X and Lua_LA_TE_X directly support accented section and file names.

for HTML output: 1160 \begin{warpHTML}

42.1 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.

```
1161 \newbool{LWR@mainmatter}
1162 \DeclareDocumentCommand{\mainmatter}{}{%
1163 \booltrue{LWR@mainmatter}}%
1164 }
```

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

```
1165 \DeclareDocumentCommand{\frontmatter}{}{%
1166 \boolfalse{LWR@mainmatter}}%
1167 }
```

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

```
1168 \DeclareDocumentCommand{\backmatter}{}{%
```

```
1169 \boolfalse{LWR@mainmatter}
1170 }
```

42.2 Sectioning support macros

`\LWR@sectionnumber {<section type>}`

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

```
1171 \newcommand*{\LWR@sectionnumber}[1]{%
1172 \InlineClass{sectionnumber}{#1} %
1173 }
```

autosec A tag used by the TOC and index.

`\LWR@createautosec {<section type>}`

Create an autosection tag.

```
1174 \newcommand*{\LWR@createautosec}[1]{%
1175 \LWR@htmntag{#1 id="autosec-\thepage"{} }%
1176 }
```

`\LWR@pushoneclose {<depth>} {<printclose>}` 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.

```
1177 \NewDocumentCommand{\LWR@pushoneclose}{m m}{\pushclose{#2}{#1}}
```

`\LWR@startnewdepth {<depth>} {<printclose>}`

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.

```
1178 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
1179 \LWR@closeprevious{#1}%
```

Push a new section depth:

```
1180 \LWR@pushoneclose{\#1}{\#2}%
1181 }
```

Ctr `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.

```
1182 \newcounter{LWR@prevFileDepth}
1183 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

```
\LWR@section * [(TOC name)] {(name)} {(sectiontype)}
```

The common actions for the high-level sectioning commands.

```
1184 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
1185 \LWR@traceinfo{LWR@section}%
1186 \LWR@stopars%
```

Cancel special `minipage` horizontal space interaction:

```
1187 \global\boolfalse{LWR@minipagethispar}%
```

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

```
1188 \LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
1189 \IfBooleanTF{\#1}{% starred
```

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

```
1190 \LWR@orignewpage%
1191
1192 }% not starred
1193 \ifthenelse{%
1194 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}}%
1195 \AND%
1196 \(%
1197 \NOT\boolean{CombineHigherDepths}\OR%
1198 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}}%
1199 \)%
1200 }%
```

If so: start a new HTML file:

```
1201 {% new file
1202 \LWR@traceinfo{LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
1203 \IfNoValueTF{#2}{%
```

If no optional entry

```
1204 {\LWR@newhtmlfile{#3}}{%
```

If yes an optional entry

```
1205 {\LWR@newhtmlfile{#2}}{%
```

```
1206 }% new file
```

Else: No new HTML file:

```
1207 {%
  not new file
}
```

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

```
1208 \LWR@orignewpage{%
```

```
1209
```

```
1210 }% not new file
```

```
1211 }% not starred
```

Remember this section's name for \nameref:

```
1212 \LWR@traceinfo{\LWR@section: about to \LWR@setlatestname}{%
```

```
1213 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}{%
```

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

```
1214
```

```
1215 \ifbool{HTMLDebugComments}{%
```

```
1216 \LWR@htmlcomment{Opening #4 '#3' {}}
```

```
1217 }{}}
```

```
1218
```

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

```
1219 \ifthenelse{%
```

```
1220 \cnttest{\csuse{\LWR@depth#4}}{>=}{\LWR@depthparagraph}{%
```

```
1221 }%
```

```
1222 {\LWR@startpars}
```

```
1223 {}
```

Create the opening tag with an autosec:

```
1224 \LWR@createautosec{\csuse{\LWR@tag#4}}{%
```

If not starred, step counter and add to TOC:

```
1225 \IfBooleanTF{#1}%
1226 {}% starred
1227 {}% not starred
```

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

```
1228 \ifthenelse{%
1229 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}}%
1230 }%
1231 {}% if secnumdepth
```

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

```
1232 \LWR@traceinfo{LWR@section: about to test main matter}%
1233 \ifbool{LWR@mainmatter}%
1234 {}%
1235 \LWR@traceinfo{LWR@section: yes mainmatter}%
1236 \refstepcounter{#4}%
```

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

```
1237 \LWR@traceinfo{LWR@section: about to addcontentsline}%
1238 \addcontentsline{toc}{#4}%
1239 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
1240 \IfValueTF{#2}{#2}{#3}}%
1241 \LWR@traceinfo{LWR@section: finished addcontentsline}%
1242 }% end of if main matter
```

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

```
1243 {}% not main matter
1244 \LWR@traceinfo{LWR@section: no main matter}%
1245 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
1246 }% end of not main matter
1247 }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```
1248 {}%
1249 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
1250 }%
```

For part, print the section type:

```
1251 \ifbool{LWR@mainmatter}%
1252 {}%
```

```

1253 \ifthenelse{%
1254   (\cnttest{\csuse{LWR@depth#4}}{<=})%
1255   {\value{secnumdepth}}\) \AND
1256   (\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\)
1257 }%
1258 {\csuse{#4name}{}-{}}
1259 {}%

```

Print the section number:

```

1260 \LWR@traceinfo{LWR@section: about to print section number}%
1261 \ifthenelse{%
1262   \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}\%
1263 }%
1264 {\protect\LWR@sectionnumber{\csuse{the#4}}\}%
1265 {}%
1266 \LWR@traceinfo{LWR@section: finished print section number}%
1267 }{}%
1268 }% end of not starred

```

Print the section name:

```
1269 #3
```

close the heading tag, such as /H2

```
1270 \LWR@htmlltag{\csuse{LWR@tag#4end}}%
```

Generate a L^AT_EX label:

```
1271 \label{autopage-\thepage}%
```

Start paragraph handing unless is an inline paragraph or subparagraph:

```

1272 \ifthenelse{%
1273   \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}\}%
1274 {\LWR@startpars}%
1275 }%

```

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.

```

1276 \IfBooleanTF{#1}{}{%
  \notstarred
  \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
}
```

```
1278 }% not starred  
1279 \LWR@traceinfo{LWR@section: done} %  
1280 }
```

42.3 \section and friends

```
\part * [<TOC name>] {<name>}  
  
1281 \@ifundefined{chapter}  
1282 {}  
1283 {%  
1284 \DeclareDocumentCommand{\part}{s o m}{%  
1285 \LWR@epubprintpendingfootnotes%  
1286 \LWR@stoppars%  
1287  
1288 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart} %  
1289  
1290 \LWR@section{#1}{#2}{#3}{part} %  
1291 }  
1292 }  
  
\chapter * [<TOC name>] {<name>}  
  
1293 \ifundefined{chapter}  
1294 {}  
1295 {%  
1296 \DeclareDocumentCommand{\chapter}{s o m}{%  
1297 \LWR@traceinfo{chapter #3} %  
1298 \LWR@epubprintpendingfootnotes%  
1299 \LWR@stoppars%  
1300 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter} %  
1301  
1302 \LWR@traceinfo{chapter: about to LWR@section} %  
1303 \LWR@section{#1}{#2}{#3}{chapter} %  
1304 \LWR@traceinfo{chapter: done} %  
1305 }  
1306 }  
  
\section * [<TOC name>] {<name>}  
  
1307 \DeclareDocumentCommand{\section}{s o m}{%  
1308 \LWR@epubprintpendingfootnotes%  
1309 \LWR@stoppars%  
1310  
1311 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection} %  
1312
```

```
1313 \LWR@section{#1}{#2}{#3}{section}%
1314 }

\subsection * [(TOC name)] {(name)}

1315 \DeclareDocumentCommand{\subsection}{s o m}{%
1316 \LWR@epubprintpendingfootnotes%
1317 \LWR@stoppars%
1318
1319 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
1320
1321 \LWR@section{#1}{#2}{#3}{subsection}%
1322 }

\subsubsection * [(TOC name)] {(name)}

1323 \DeclareDocumentCommand{\subsubsection}{s o m}{%
1324 \LWR@epubprintpendingfootnotes%
1325 \LWR@stoppars%
1326
1327 \LWR@startnewdepth{\LWR@depthsubsubsection}{%
1328 {\LWR@printclosesubsubsection}%
1329
1330 \LWR@section{#1}{#2}{#3}{subsubsection}%
1331 }

\paragraph * [(TOC name)] {(name)}

1332 \DeclareDocumentCommand{\paragraph}{s o m}{%
1333 \LWR@epubprintpendingfootnotes%
1334 \LWR@stoppars%
1335
1336 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
1337
1338 \LWR@section{#1}{#2}{#3}{paragraph}%
1339 }

\ subparagraph * [(TOC name)] {(name)}

1340 \DeclareDocumentCommand{\ subparagraph}{s o m}{%
1341 \LWR@epubprintpendingfootnotes%
1342 \LWR@stoppars%
1343
1344 \LWR@startnewdepth{\LWR@depth subparagraph}{\LWR@printclosesubparagraph}%
1345
1346 \LWR@section{#1}{#2}{#3}{ subparagraph}%
1347 }
```

1348 \end{warpHTML}

43 Starting a new file

for HTML output: 1349 \begin{warpHTML}

\MetaLanguage Default language for the HTML lang tag.

1350 \newcommand*\{\MetaLanguage\}{en-US}

\LWR@filestart {\langle title_suffix\rangle}

Creates the opening HTML tags.

1351 \newcommand*\{\LWR@filestart\}[1]{

Locally temporarily disable direct-formatting commands:

```
1352 \begingroup
1353 \renewcommand{\textit}[1]{##1}% not used in filenames
1354 \renewcommand{\textsc}[1]{##1}
1355 \renewcommand{\textsl}[1]{##1}
1356 \renewcommand{\textbf}[1]{##1}
1357 \renewcommand{\texttt}[1]{##1}
1358 \renewcommand{\textsf}[1]{##1}
1359 \renewcommand{\textrm}[1]{##1}
1360 \renewcommand{\textsuperscript}[1]{##1}
1361 \renewcommand{\textsubscript}[1]{##1}
1362 \renewcommand*\{\HTMLUnicode\}[1]{}
1363 \renewcommand*\{\HTMLEntity\}[1]{}
1364 \RenewDocumentCommand{\LWR@htmlspanclass}{m o +m}{##3}
1365 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}
```

Create the page's HTML header:

1366 \LWR@htmlltag{!DOCTYPE html}\LWR@orignewline

The language is user-adjustable:

1367 \LWR@htmlltag{html lang="\MetaLanguage"\}\LWR@orignewline

Start of the meta data:

1368 \LWR@htmlltag{head}\LWR@orignewline

Charset is fixed at UTF-8:

```
1369 \LWR@htmltag{meta charset="UTF-8" /}\LWR@orignewline
```

Author:

```
1370 \ifcsempty{theHTMLauthor}{}{%
1371 \LWR@htmltag{meta name="author" content="\theHTMLauthor" /}\LWR@orignewline
1372 }
```

lwarf is the generator:

```
1373 \LWR@htmltag{meta name="generator" content="LaTeX lwarf package" /}%
1374     \LWR@orignewline
```

If there is a description, add it now:

```
1375 \ifdefempty{\LWR@currentHTMLdescription}{}{%
1376 \LWR@htmltag{%
1377 meta name="description" content="\LWR@currentHTMLdescription" /}%
1378     \LWR@orignewline
1379 }%
```

Mobile-friendly viewport:

```
1380 \LWR@htmltag{meta name="viewport" %
1381 content="width=device-width, initial-scale=1.0" /}%
1382     \LWR@orignewline
```

IE patch:

```
1383 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
1384 \LWR@htmltag{%
1385 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{}%
1386 \LWR@htmltag{/script}\LWR@orignewline
1387 \LWR@htmltag{![-]{-}}\LWR@orignewline
```

The page's title:

```
1388 \ifcsvoid{thetitle}{}{%
1389 \LWR@htmltag{title}\thetitle#\LWR@htmltag{/title}\LWR@orignewline%
1390 }%
```

The page's stylesheet:

```
1391 \LWR@htmltag{%
1392 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
1393 \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.

```
1394 \ifbool{mathjax}%
1395 {%
1396 \boolfalse{LWR@verbtags}%
1397 \VerbatimInput{lwarf_mathjax.txt}%
1398 \booltrue{LWR@verbtags}%
1399 \LWR@stoppars%
1400 }% end of mathjax
1401 {}%
```

End of the header:

```
1402 \LWR@htmlltag{/head}\LWR@orignewline
```

Start of the body:

```
1403 \LWR@htmlltag{body}\LWR@orignewline
1404 \endgroup
1405 }
```

```
1406 \end{warpHTML}
```

44 Starting HTML output

for HTML output: 1407 \begin{warpHTML}

\LWR@LwarfStart Executed at the beginning of the entire document.

```
1408 \newcommand*\LWR@LwarfStart{%
1409 {}%
```

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

```
1410 \ifbool{FormatWordProcessor}{%
1411 \setcounter{FileDepth}{-5}%
1412 \boolfalse{HTMLDebugComments}%
1413 }{}
```

Expand and detokenize \HomeHTMLFileName and \HTMLFileName:

```
1414 \edef\LWR@strresult{\HomeHTMLFileName}
```

```
1415 \edef\HomeHTMLFileName{\detokenize\expandafter{\LWR@strresult}}
1416 \edef\LWR@strresult{\HTMLfiveName}
1417 \edef\HTMLfiveName{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn:

```
1418 \LWR@origonecolumn%
```

Reduce chance of line overflow in verbatim environments:

```
1419 \LWR@origscriptsize%
```

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

```
1420 \LWR@origraggedright%
1421 \let\\LWR@endofline%
```

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

```
1422 \linespread{1.3}%
```

For `pdftotext` to reliably identify paragraph splits:

```
1423 \setlength{\parindent}{0pt}
1424 \setlength{\parskip}{2ex}
```

For the `lateximages` record file:

```
1425 \immediate\openout\LWR@file=lateximages.txt
```

Removes space after the caption in the HTML:

```
1426 \setlength{\belowcaptionskip}{-3ex}
```

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

```
1427 \renewcommand{\ps@plain}{}%
```

```
\centering Not used in the HTML environment:
\raggedleft
\raggedright 1428 \renewcommand*{\centering}{}%
1429 \renewcommand*{\raggedleft}{}%
1430 \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:

```

1431 \let\LWR@origtabular\tabular
1432 \let\LWR@origendtabular\endtabular
1433 \let\tabular\LWR@tabular
1434 \let\endtabular\endLWR@tabular

```

Float captions:

```
1435 \let\LWR@origcaption\caption
```

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

```

1436 \let\LWR@origltx@label\ltx@label
1437 \let\ltx@label\LWR@htmlmathlabel

```

Logos:

```

1438 \let\TeX\LWR@TeX
1439 \let\LaTeX\LWR@LaTeX
1440 \let\LuaTeX\LWR@LuaTeX
1441 \let\LuaLaTeX\LWR@LuaLaTeX
1442 \let\XeTeX\LWR@XeTeX
1443 \let\XeLaTeX\LWR@XeLaTeX
1444 \let\ConTeXt\LWR@ConTeXt

```

Graphics:

```

1445 \let\rotatebox\LWR@rotatebox
1446 \let\scalebox\LWR@scalebox
1447 \let\reflectbox\LWR@reflectbox

```

Not yet started any paragraph handling:

```

1448 \global\boolfalse{LWR@doingapar}
1449 \global\boolfalse{LWR@doingstartpars}

```

Start a new HTML file and a header:

```

1450 \LWR@filestart{}
1451 \LWR@htmlltag{header}\LWR@orignewline
1452 \LWR@startpars
1453 \LWR@firstpagetop
1454 \LWR@stoppars
1455 \LWR@htmlltag{/header}\LWR@orignewline
1456 \LWR@htmlltag{section class="textbody"{}}
1457 \LWR@origpagestyle{empty}

```

Document and page settings:

```
1458 \mainmatter
1459 \LWR@origpagenumbering{arabic}
```

Set default titlepage thanks footnote marks. See section 46.6.

```
1460 \if@titlepage
1461   \thanksmarkseries{arabic}
1462 \else
1463   \thanksmarkseries{fnsymbol}
1464 \fi
```

Initial default patch for fancyvrb:

```
1465 \fvset{frame=none}%
```

The ampersand is redefined active, and acts depending on whether it is inside a tabular.

```
1466 \catcode`&=\active
```

Allow HTML paragraphs to begin:

```
1467 \LWR@startpars
1468 }
```

```
1469 \end{warpHTML}
```

45 Ending HTML output

for HTML output: 1470 \begin{warpHTML}

\LWR@requesttoc {*boolean*} {*suffix*} Requests that a toc, lof, or lot be generated.

```
1471 \newcommand*{\LWR@requesttoc}[2]{%
1472 \ifbool{#1}{%
1473 {%
1474 \expandafter\newwrite\csuse{tf@#2}%
1475 \immediate\openout \csuse{tf@#2} \jobname.\#2\relax%
1476 }{}%
1477 }}
```

\LWR@LwarpEnd Final stop of all HTML output:

```
1478 \newcommand*{\LWR@LwarpEnd}
```

```
1479 {
1480 \LWR@stoppars
1481 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
1482 \LWR@htmlelementclassend{section}{textbody}
```

Print any pending footnotes:

```
1483 \LWR@printpendingfootnotes
```

Create the footer:

```
1484 \LWR@htmlelement{footer}
1485
1486 \LWR@pagebottom
1487
1488 \LWR@htmlelementend{footer}
```

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

Presumably has a table-of-contents.

```
1489 \ifthenelse{\boolean{FormatEPUB} \OR \boolean{FormatWordProcessor}}
1490 {}
1491 {
1492 \ifnumcomp{\value{LWR@htmlfilename}}{>}{0}{\LWR@botnavigation}{}
1493 }

1494 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
1495 \LWR@htmlelement{/body}\LWR@orignewline
1496 \LWR@htmlelement{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
1497 \LWR@orignewpage
```

For `\immediate\closeout\file` commands:

```
1498 \immediate\closeout\file
1499 }
```

1500 \end{warpHTML}

46 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`:

`\maketitlehookaa` **\maketitlehookaa:** Between “published” and the title.

`\maketitlehookaaa` **\maketitlehookaaa:** Between the title and the subtitle.

`\prepublished` **\prepublished:** Before the “published” field.

`\postpublished` **\postpublished:** After the “published” field.

`\presubtitle` **\presubtitle:** Before the subtitle.

`\postsubtitle` **\postsubtitle:** After the subtitle.

`\printthanks` **\printthanks:** has been added to force the printing of thanks inside a `titlingpage` environment when `\maketitle` 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}
...
```

46.1 Setting the title, etc.

The following provide setting commands for both HTML and print outputs.

<pre>\published \@title, \@subtitle, \@author, etc. store the values as originally assigned, in- \title including any \thanks, \and, or \affiliation. These are low-level macros intended \subtitle to be used by other macros only inside a <code>titlepage</code> or <code>titlingpage</code>, and are used \author by <code>\maketitle</code>. The author is printed inside a single-column table, which becomes \date multiple single-column tables if multiples authors are included. \printpublished \printtitle, \printsubtitle, etc. are user-level macros intended to be used \printtitle in <code>titlepage</code> and <code>titlingpage</code> environments in cases where <code>\maketitle</code> is not \printsubtitle desired. These commands preserve the \thanks, etc., and should not be used in \printauthor the main text. The author is printed inside a single-column table, which becomes \printdate multiple single-column tables if multiples authors are included. \thepublished \thetitle, \thesubtitle, \theauthor, etc. are user-level sanitized versions which \thetitle have removed the \thanks and \affiliation, and \and is changed for inline text \thesubtitle usage. The author is printed inline without \affiliation or \thanks, with \and \theauthor placing commas between multiple authors. Thus, these commands are to be used \thedate 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:</pre>	<pre>\SetPageBottom {{text}}</pre>
---	------------------------------------

```
\SetPageBottom{
\begin{center}\textcopyright~2016 \theauthor\end{center}
}
```

<pre>\author {{author}} While using <code>\maketitle</code>, the author is treated as a single-column table \and 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</pre>

```
\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 {<text>}
```

Adds the affiliation to the author for use in `\maketitle`. Nullified when later used for inline use of `\theauthor`.

for HTML output: 1501 `\begin{warpHTML}`
 1502 `\newrobustcmd{\affiliation}[1]{\\ \InlineClass{affiliation}{#1}}`
 1503 `\end{warpHTML}`

for PRINT output: 1504 `\begin{warpprint}`
 1505 `\newrobustcmd{\affiliation}[1]{\\ \textsc{\small#1}}`
 1506 `\end{warpprint}`

The following are based on the original titling code:

for HTML & PRINT: 1507 `\begin{warpall}`

```
\author {<text>}
```

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.

```
1508 \renewcommand{\author}[1]{%  

  1509 \gdef\@author{#1}  

  1510 \begingroup  

  1511 \renewcommand{\thanks}[1]{}  

  1512 \renewcommand{\and}{\unskip, }  

  1513 \renewcommand{\thanksmark}[1]{}  

  1514 \renewcommand{\thanksgap}[1]{}  

  1515 \renewcommand{\affiliation}[1]{}  

  1516 \protected@xdef\theauthor{#1}  

  1517 \endgroup}
```

```
\published {⟨text⟩}

1518 \newcommand{\published}[1]{%
1519 \gdef\@published{#1}%
1520 \begingroup%
1521 \renewcommand{\thanks}[1]{}%
1522 \renewcommand{\thanksmark}[1]{}%
1523 \renewcommand{\thanksgap}[1]{}%
1524 \protected@xdef\thepublished{#1}%
1525 \endgroup%
1526 }%
1527 \newcommand{\@published}{}%
1528 \newcommand{\thepublished}{}%

\subtitle {⟨text⟩}

1529 \newcommand{\subtitle}[1]{%
1530 \gdef\@subtitle{#1}%
1531 \begingroup%
1532 \renewcommand{\thanks}[1]{}%
1533 \renewcommand{\thanksmark}[1]{}%
1534 \renewcommand{\thanksgap}[1]{}%
1535 \protected@xdef\thesubtitle{#1}%
1536 \endgroup%
1537 }%
1538 \newcommand{\@subtitle}{}%
1539 \newcommand{\thesubtitle}{}%

1540 \end{warpall}
```

46.2 Changes to HTML titlepage and titlingpage

for HTML output: 1541 \begin{warpHTML}

Env **titlepage** Sets up a titlepage div with a L^AT_EX PDF minipage inside.

```
1542 \renewenvironment*{titlepage}%
1543 {%
1544 \LWR@forcenewpage%
1545 \BlockClass{titlepage}\LWR@subminipage%
1546 }%
1547 {\LWR@endsubminipage\endBlockClass}
```

Env **titlingpage**

```
1548 \renewenvironment*{titlingpage}
1549 {%
```

Start an HTML titlepage div:

```
1550 \begin{titlepage}
```

Prepare for a custom version of `\maketitle` inside the `titlingpage`:

```
1551 \LWR@maketitlesetup
1552 \let\maketitle\LWR@titlingmaketitle
1553 }
1554 {
```

At the end of the environment, end the HTML titlepage div:

```
1555 \end{titlepage}
```

Reset the footnote counter:

```
1556 \@bscontmark
1557 }
```

```
1558 \end{warpHTML}
```

for HTML & PRINT: 1559 `\begin{warpall}`

`\printthanks` Forces the `\thanks` to be printed.

This is necessary in a `titlingpage` environment when `\maketitle` was not used.

```
1560 \newcommand*{\printthanks}{\@thanks}
```

Env `titlingpage` At the end of the `titlingpage` for both print and HTML, reset footnote markers to zero.

```
1561 \AtEndEnvironment{titlingpage}{\@bscontmark}
```

```
1562 \end{warpall}
```

46.3 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: 1563 `\begin{warpHTML}`

Patch the pre/post title/author/date to add HTML tags, then initialize:

```
1564 \newcommand{\prepublished}[1]{%
1565 \def\@bsprepublished{\BlockClass{published}#1}%
1566 }
1567
1568 \newcommand{\postpublished}[1]{%
1569 \def\@bspostpublished{\#1\endBlockClass}%
1570 }
1571
1572 \renewcommand{\pretitle}[1]{%
1573 \def\@bspretitle{\#1\LWR@stoppars\LWR@htmltag{h1}}%
1574 }
1575
1576 \renewcommand{\posttitle}[1]{%
1577 \def\@bsposttitle{\LWR@htmltag{/h1}\LWR@startpars#1}%
1578 }
1579
1580 \newcommand{\presubtitle}[1]{%
1581 \def\@bspresubtitle{\BlockClass{subtitle}#1}%
1582 }
1583
1584 \newcommand{\postsubtitle}[1]{%
1585 \def\@bspostsubtitle{\#1\endBlockClass}%
1586 }
1587
1588 \renewcommand{\preauthor}[1]{%
1589 \def\@bspreauthor{\BlockClass{author}#1}%
1590 }
1591
1592 \renewcommand{\postauthor}[1]{%
1593 \def\@bspostauthor{\#1\endBlockClass}%
1594 }
1595
1596 \renewcommand{\predate}[1]{%
1597 \def\@bspredate{\#1\BlockClass{titledate}}%
1598 }
1599
1600 \renewcommand{\postdate}[1]{%
1601 \def\@bspostdate{\endBlockClass#1}%
1602 }
1603
1604 \prepublished{\begin{center}}
1605 \postpublished{\par\end{center}}
1606
1607 \pretitle{\begin{center}}
1608 \posttitle{\par\end{center}}
1609
1610 \presubtitle{\begin{center}}
1611 \postsubtitle{\par\end{center}}
```

```
1612
1613 \preauthor{\begin{center}%
1614 \begin{tabular}[t]{c}%
1615 }%
1616 \postauthor{\end{tabular}\par\end{center}}%
1617
1618 \predate{\begin{center}}%
1619 \postdate{\par\end{center}}
```

\printpublished

```
1620 \newcommand*\printpublished{%
1621 \ifthenelse{\equal{\thepublished}{}}%
1622 {}%
1623 {%
1624 \begin{BlockClass}{published}%
1625 @published%
1626 \end{BlockClass}%
1627 }%
1628 }
```

\printtitle

```
1629 \newcommand*\printtitle{%
1630 {%
1631 \LWR@stoppars%
1632 \LWR@htmlltag{h1}%
1633 @title%
1634 \LWR@htmlltag{/h1}%
1635 \LWR@startpars%
1636 }}
```

\LWR@printthetitle A private version which prints the title without footnotes, used to title each HTML page.

```
1637 \newcommand*\LWR@printthetitle{%
1638 {%
1639 \LWR@stoppars%
1640 \LWR@htmlltag{h1}%
1641 @title%
1642 \LWR@htmlltag{/h1}%
1643 \LWR@startpars%
1644 }}
```

\printsubtitle

```
1645 \newcommand*\printsubtitle{%
```

```

1646 \ifthenelse{\equal{\thesubtitle}{}}{%
1647 {}%
1648 {%
1649 \begin{BlockClass}{subtitle}%
1650 \@subtitle%
1651 \end{BlockClass}%
1652 }%
1653 }%
\printauthor
1654 \newcommand*{\printauthor}{%
1655 \begin{BlockClass}{author}%
1656 \begin{tabular}{c}\@author\end{tabular}%
1657 \end{BlockClass}%
1658 }%
\printdate
1659 \newcommand*{\printdate}{%
1660 \begin{BlockClass}{titledate}%
1661 \@date%
1662 \end{BlockClass}%
1663 }%
1664 \end{warpHTML}

```

46.4 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: 1665 `\begin{warpprint}`

```

\printpublished
1666 \newcommand*{\printpublished}{\Large\scshape\@published}%
\printtitle
1667 \newcommand*{\printtitle}{\Huge\@title}%
\printsubtitle
1668 \newcommand*{\printsubtitle}{\Large\itshape\@subtitle\bigskip}%

```

```
\printauthor
1669 \newcommand*{\printauthor}
1670   {{\large\begin{tabular}[t]{c}\@author\end{tabular}}}

\printdate
1671 \newcommand*{\printdate}{{\small\textrit{\@date}}}
```

46.5 \maketitle for print output

\maketitle From the `titling` package, patched to add the publisher and subtitle.

```
1672 \providecommand{\maketitle}{}
1673 \if@titlepage
1674   \renewcommand{\maketitle}{\begin{titlepage}%
1675     \let\footnotesize\small
1676     \let\footnoterule\relax
1677     \let \footnote \thanks
1678     \obsmarkseries
1679     \def\@makefnmark{\rlap{\textsuperscript{\%}}
1680       \normalfont\obsthanksheadpre \tamarck\obsthanksheadpost}}\%
1681     \long\def\@makefntext##1{\makethanksmark ##1}
1682   \null\vfil
1683   \vskip 60\p@
1684   \vspace*{\droptitle}
1685   \maketitlehooka
1686   \ifcsempty{@published}
1687 {}
1688 {{\obsprepublished\@published\obspostpublished}\maketitlehookaa}
1689   {\obspretitle\@title\obsposttitle}
1690   \ifcsempty{@subtitle}
1691 {}
1692 {{\maketitlehookaaa{\obspresubtitle\@subtitle\obspostsubtitle}}}
1693   \maketitlehookb
1694   {\obspreauthor\@author\obspostauthor}
1695   \maketitlehookc
1696   {\obspredate\@date\obspostdate}
1697   \maketitlehookd
1698   \par
1699   \thanks
1700   \vfil\null
1701   \end{titlepage}\%
1702   \obscntmark % \setcounter{footnote}{0}%
1703 %% \obsmtitleempty
1704 } % end titlepage defs
```

```
1705 \else
1706   \renewcommand{\maketitle}{\par
1707     \begingroup
1708       \@bsmarkseries
1709       \def\@makefnmark{\rlap{\@textsuperscript{%
1710         \normalfont\@bsthanksheadpre \tamar\k\@bsthanksheadpost}}}}%
1711       \long\def\@makefntext##1{\makethanksmark ##1}
1712       \if@twocolumn
1713         \ifnum \col@number=\@ne
1714           \@maketitle
1715         \else
1716           \twocolumn[\@maketitle]%
1717         \fi
1718       \else
1719         \newpage
1720         \global\@topnum\z@
1721         \@maketitle
1722       \fi
1723       \thispagestyle{plain}\@thanks
1724     \endgroup
1725     \@bscontmark % \setcounter{footnote}{0}%
1726 %% \@bsmtitleempty
1727 } % end non-titlepage
1728
1729 \def\@maketitle{%
1730   \newpage
1731   \null
1732   \vskip 2em%
1733   \vspace*{\droptitle}
1734   \maketitlehooka
1735   \ifcsempty{@published}
1736 {}
1737 {{\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
1738   {\@bspretitle \@title \@bsposttitle}
1739   \ifcsempty{@subtitle}
1740 {}
1741 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
1742   \maketitlehookb
1743   {\@bspreauthor \@author \@bspostauthor}
1744   \maketitlehookc
1745   {\@bspredate \@date \@bspostdate}
1746   \maketitlehookd
1747   \par
1748   \vskip 1.5em}
1749 \fi
1750
1751 \providecommand{\maketitlehookaa}{}%
1752
1753 \providecommand{\maketitlehookaaa}{}%
1754
```

```

1755 \newcommand{\prepublished}[1]{%
1756 \def\@bsprepublished{\#1}%
1757 }
1758
1759 \newcommand{\postpublished}[1]{%
1760 \def\@bspostpublished{\#1}%
1761 }
1762
1763 \newcommand{\presubtitle}[1]{%
1764 \def\@bspresubtitle{\#1}%
1765 }

```

\presubtitle Hook after printing the subtitle.

```

1766 \newcommand{\postsubtitle}[1]{%
1767 \def\@bspostsubtitle{\#1}%
1768 }

```

Initial settings:

```

1769 \if@titlepage
1770 \prepublished{%
1771 \vspace*{-\baselineskip}\vspace*{-\medskipamount}\vspace*{-2em}
1772 \begin{center}}%
1773 \postpublished{\par\end{center}\vskip 2em}
1774
1775 \presubtitle{\unskip\begin{center}\unskip}
1776 \postsubtitle{\par\end{center}\vskip 2em}
1777 \else
1778 \prepublished{\begin{center}}
1779 \postpublished{\par\end{center}\vskip 0.5em}
1780
1781 \presubtitle{\begin{center}\unskip}
1782 \postsubtitle{\par\end{center}\vskip 0.5em}
1783 \fi

```

```
1784 \end{warpprint}
```

46.6 \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 39 for other kinds of footnotes.

See `\thanksmarkseries{series}`, below, to set the style of the footnote marks.

for HTML output: 1785 `\begin{warpHTML}`

`\LWR@maketitlesetup` Patches `\thanks` macros to use L^AT_EX minipage footnotes.

1786 `\newcommand*\{\LWR@maketitlesetup\}{%`

Select which kind of footnote marks to use:

1787 `\@bsmarkseries`

1788 `\@mpbsmarkseries`

Redefine the footnote mark:

1789 `\def\@makefnmark{\thefootnote}`

`\thefootnote` ⇒ `\nameuse{arabic}{footnote}`, or
`\thefootnote` ⇒ `\nameuse{fnsymbol}{footnote}`

Redefine the footnote text:

1790 `\long\def\@makefntext##1{%`

Make the footnote mark and some extra horizontal space for the tags:

1791 `\makethanksmark \LWR@orighspace{1in}`

`\makethanksmark` ⇒ `\thanksfootmark` ⇒ `\tamark` ⇒
`\@thefnmark` ⇒ `\itshape a` (or similar)

Print the text:

1792 `##1%`

1793 `}%`

1794 `}`

`\@fnsymbol {<counter>}`

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used `\|` which was not being found by `pdftotext`.

1795 `\def\@fnsymbol#1{\ensuremath{\ifcase#1\or *\or \dagger\or \ddagger\or }`
1796 `\mathsection\or \mathparagraph\or \text{\HTMLunicode{2016}}\or`
1797 `**\or \dagger\dagger\or \ddagger\ddagger \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.

```
1798 \renewcommand*{\maketitle}{%
```

An HTML titlepage div is used for all classes.

```
1799 \begin{titlepage}
```

Set up special patches:

```
1800 \LWR@maketitlesetup
```

Typeset the title, etc:

```
1801 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
1802 \@thanks
```

Close the HTML titlepage div:

```
1803 \end{titlepage}
```

Reset the footnote counter:

```
1804 \@bscontmark
```

```
1805 }
```

\@maketitle Typesets the title, etc. for HTML:

```
1806 \DeclareDocumentCommand{\@maketitle}{}{%
1807 \maketitlehooka
1808 \ifcsempty{@published}%
1809 {}%
1810 {\{\@bsprepublished \@published \@bspostpublished\}\maketitlehookaa}%
1811 {\{\@bspretitle \@title \@bsposttitle\}%
1812 \ifcsempty{@subtitle}%
1813 {}%
1814 {\{\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}\}%
1815 \maketitlehookb
1816 {\{\@bspreauthor \@author \@bspostauthor\}%
1817 \maketitlehookc
1818 {\{\@bspredate \@date \@bspostdate\}%
1819 \maketitlehookd
1820 }%
```

```
1821 \providecommand{\maketitlehookaa}{}  
1822 \providecommand{\maketitlehookaaa}{}  
1823 \newcommand*\LWR@titlingmaketitle{%
```

\LWR@titlingmaketitle \maketitle for use inside an HTML titlingpage environment.

```
1824 \@maketitle
```

Typeset the title, etc:

```
1825 \@thanks
```

```
1826 }
```

\thanksmarkseries {\langle series\rangle}

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.

```
1827 \renewcommand{\thanksmarkseries}[1]{%  
1828 \def\@mpbsmarkseries{  
1829 \renewcommand*{\thempfootnote}{\@nameuse{\#1}{mpfootnote}}}%  
1830 \def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{\#1}{footnote}}}%  
1831 }
```

```
1832 \end{warpHTML}
```

47 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: 1833 \begin{warpHTML}

\abstractname User-redefinable title for the abstract.

Also over-written by the babel package.

```
1834 \providecommand*{\abstractname}{Abstract}
```

```

Env abstract

1835 \DeclareDocumentEnvironment{abstract}{}{%
1836 {%
1837 \LWR@forcenewpage
1838 \BlockClass{abstract}{%
1839 \BlockClassSingle{abstracttitle}{\abstractname}{%
1840 }{%
1841 {%
1842 \endBlockClass
1843 }{%
1844 \end{warpHTML}

```

48 Quote and verse

48.1 Citations and attributions

\attribution for use inside quote, quotation, verse:

ex: \attribution{author name} --- \citetitle{book name}

for HTML output: 1845 \begin{warpHTML}
1846 \newcommand{\attribution}[1]{%
1847 \InlineClass{attribution}{--\,#1}}% emdash
1848 \end{warpHTML}

for PRINT output: 1849 \begin{warpprint}
1850 \newcommand{\attribution}[1]{\textsc{---\,#1}}
1851 \end{warpprint}

\citetitle for use inside quote, quotation, verse:

for HTML output: 1852 \begin{warpHTML}
1853 \newcommand{\citetitle}[1]{%
1854 \InlineClass{citetitle}{--\,#1}}% emdash
1855 \end{warpHTML}

for PRINT output: 1856 \begin{warpprint}
1857 \newcommand{\citetitle}[1]{\textsl{---\,#1}}
1858 \end{warpprint}

48.2 Quotes, quotations

for HTML output: 1859 \begin{warpHTML}

```
Env  quote
1860 \renewenvironment*{quote}
1861 {
1862 \LWR@forcenewpage
1863 \LWR@htmlblocktag{blockquote}
1864 }
1865 {\LWR@htmlblocktag{/blockquote}}
1866
1867 \renewenvironment*{quotation}
1868 {
1869 \LWR@forcenewpage
1870 \LWR@htmlblocktag{blockquotation}
1871 }
1872 {\LWR@htmlblocktag{/blockquotation}}
1873 \end{warpHTML}
```

48.3 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. `l warp` 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}
```

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

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

for HTML & PRINT: 1874 `\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.

```
1875 \newlength{\HTMLvleftskip}
1876 \setlength{\HTMLvleftskip}{1em}
```

Len `\TMLleftmargini` Sets `\leftmargini` inside a `verse` environment in HTML.

```
1877 \newlength{\HTMLleftmargini}
1878 \setlength{\HTMLleftmargini}{4.5em}
```

```
1879 \end{warpall}
```

49 Verbatim

for HTML output: 1880 `\begin{warpHTML}`

Env `verbatim`

```
1881 \AfterEndPreamble{
1882 \AtBeginEnvironment{verbatim}{%
1883 \LWR@forcenewpage
1884 \LWR@atbeginverbatim{verbatim}\unskip\vspace*{-\baselineskip}%
1885 }
1886 \AfterEndEnvironment{verbatim}{\unskip\vspace*{-\baselineskip}\LWR@afterendverbatim}%
1887 }

1888 \end{warpHTML}
```

50 Fancyvrb

for HTML & PRINT: 1889 `\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.

```
1890 \newlength{\VerbatimHTMLWidth}
1891 \setlength{\VerbatimHTMLWidth}{4in}
1892 \end{warpall}
```

for HTML output: 1893 \begin{warpHTML}

Bool LWR@verbtags Used to temporarily turn off verbatim tags while doing VerbatimInput in the HTML head.

```
1894 \newbool{LWR@verbtags}
1895 \booltrue{LWR@verbtags}
```

For \VerbatimFootnotes:

```
1896 \renewcommand{\VerbatimFootnotes}{%
1897 \PackageError{l warp}%
1898 {Verbatim footnotes are not yet supported by l warp.}%
1899 {This may be improved some day.}%
1900 }
```

\LWR@atbeginverbatim {\langle class\rangle}

Encloses a verbatim environment with the given CSS class.

```
1901 \newcommand*{\LWR@atbeginverbatim}[1]
1902 {%
```

Avoid excessive space between lines:

```
1903 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
1904 \LWR@stoppars%
```

Create a new pre of the given class:

```
1905 \ifbool{LWR@verbtags}{\LWR@htmlltag{pre class="#1"{})){}}%
```

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.

```
1906 \LWR@origttfamily%
```

Do not produce HTML tags for `\hspace` inside a verse `par`. Restore plain L^AT_EX `\hspace` functionality:

```
1907 \let\hspace{\LWR@orighspace%
1908 }
```

`\LWR@afterendverbatim` Finishes enclosing a verbatim environment.

```
1909 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the `pre`:

```
1910 \unskip%
```

At the end of the environment, close the `pre`:

```
1911 \ifbool{\LWR@verbtags}{\noindent\LWR@htmlltag{/pre}%
1912
1913 }{}}
```

Resume regular paragraph handling:

```
1914 \LWR@startpars%
1915 }
```

`\LWR@Verbatimclass` Holds the class of the following verbatim.

```
1916 \newcommand*{\LWR@Verbatimclass}{fancyvrb}
```

Env `VerbatimClass` {*class*} [*Verbatim options*]

Creates a `Verbatim` enclosed in a `<div>` of the given class.

```
1917 \NewDocumentEnvironment{VerbatimClass}{m 0{}}
1918 {%
1919 \renewcommand*{\LWR@Verbatimclass}{#1}%
1920 \LWR@origVerbatim[#2]%
1921 }
1922 {\endVerbatim}
```

After the preamble is loaded, after any patches to `Verbatim`:

```
1923 \AfterEndPreamble{
```

Remember the original defintion of `Verbatim`:

```
1924 \let\LWR@origVerbatim\Verbatim
```

Env Verbatim Patched to place the environment in a `fancyverb` 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.

```

1925 \renewcommand*\{\\Verbatim\}{%
1926 \\LWR@forcenewpage
1927 \renewcommand*\{\\LWR@Verbatimclass\}{fancyverb}\%
1928 \\LWR@origVerbatim\%
1929 }

```

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 `fancyverb` package.

```

1930 \\newcommand*\{\\LWR@fvstartnone\}{%
1931 \\LWR@traceinfo{fvstartnone}\%
1932 \\ifbool{\\LWR@verbtags}{\\hbox to\\z@{\\LWR@htmltagc{div class=\"\\LWR@Verbatimclass\"}}}{\\}%
1933 \\hbox to\\z@{\\LWR@atbeginverbatim{verbatim}}\%
1934 }
1935
1936 \\newcommand*\{\\LWR@f vendnone\}{%
1937 \\LWR@traceinfo{f vendnone}\%
1938 \\hbox to\\z@{\\LWR@afterendverbatim}\%
1939 \\ifbool{\\LWR@verbtags}{\\hbox to\\z@{\\LWR@htmltagc{/div}}}{\\}%
1940 }
1941
1942 \\newcommand*\{\\LWR@fvstartsingle\}{%
1943 \\LWR@traceinfo{fvstartsingle}\%
1944 \\LWR@fvstartnone\%
1945 \\FV@BeginListFrame@Single\%
1946 }
1947
1948 \\newcommand*\{\\LWR@fvendsingle\}{%
1949 \\LWR@traceinfo{fvendsingle}\%
1950 \\FV@EndListFrame@Single\%
1951 \\LWR@f vendnone\%
1952 }
1953
1954 \\newcommand*\{\\LWR@fvstartline\}{%
1955 \\LWR@traceinfo{fvstartline}\%
1956 \\LWR@fvstartnone\%
1957 \\FV@BeginListFrame@Lines\%
1958 }
1959
1960 \\newcommand*\{\\LWR@fvendline\}{%
1961 \\LWR@traceinfo{fvendline}\%
1962 \\FV@EndListFrame@Lines\%
1963 \\LWR@f vendnone\%

```

1964 }

The following patches select the start/left/right/end behaviors depending on `frame`. Original code is from the `fancyvrb` package.

```

1965 \def\FV@Frame@none{%
1966 \let\FV@BeginListFrame\LWR@fvstartnone%
1967 \let\FV@LeftListFrame\relax%
1968 \let\FV@RightListFrame\relax%
1969 \let\FV@EndListFrame\LWR@fvendnone}
1970
1971 \def\FV@Frame@singl{%
1972 \let\FV@BeginListFrame\LWR@fvstartsingl%
1973 \let\FV@LeftListFrame\FV@LeftListFrame@Singl%
1974 \let\FV@RightListFrame\FV@RightListFrame@Singl%
1975 \let\FV@EndListFrame\LWR@fvendsingl}
1976
1977 \def\FV@Frame@lines{%
1978 \let\FV@BeginListFrame\LWR@fvstartline%
1979 \let\FV@LeftListFrame\relax%
1980 \let\FV@RightListFrame\relax%
1981 \let\FV@EndListFrame\LWR@fvendline}
1982
1983 \def\FV@Frame@topline{%
1984 \let\FV@BeginListFrame\LWR@fvstartline%
1985 \let\FV@LeftListFrame\relax%
1986 \let\FV@RightListFrame\relax%
1987 \let\FV@EndListFrame\LWR@fvendnone}
1988
1989 \def\FV@Frame@bottomline{%
1990 \let\FV@BeginListFrame\LWR@fvstartnone%
1991 \let\FV@LeftListFrame\relax%
1992 \let\FV@RightListFrame\relax%
1993 \let\FV@EndListFrame\LWR@fvendline}
1994
1995 \def\FV@Frame@leftline{%
1996 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
1997 \ifx\fancyVerbFillColor\relax%
1998 \let\FV@FrameFillLine\relax%
1999 \else%
2000 \tempdima\FV@FrameRule\relax%
2001 \multiply\tempdima-\tw@%
2002 \edef\FV@FrameFillLine{%
2003 {\noexpand\fancyVerbFillColor{\vrule\width\tempdima\sp}%
2004 \kern-\tempdima\sp}}%
2005 \fi%
2006 \let\FV@BeginListFrame\LWR@fvstartnone%
2007 \let\FV@LeftListFrame\FV@LeftListFrame@Singl%
2008 \let\FV@RightListFrame\relax%

```

```
2009 \let\FV@EndListFrame\LWR@fvendnone}
```

Adds the optional label to the top and bottom edges. Original code is from the `fancyvrb` package.

```
2010 \def\FV@SingleFrameLine#1{%
2011   \hbox to\z@{%
2012     \kern\leftmargin
2013     \ifnum#1=\z@\relax
2014       \let\FV@Label\FV@LabelBegin
2015     \else
2016       \let\FV@Label\FV@LabelEnd
2017     \fi
2018     \ifx\FV@Label\relax
2019       \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
2020     \else
2021       \ifnum#1=\z@
2022         \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
2023         \ifx\FV@LabelPositionTopLine\relax
2024 \else
2025           \LWR@htmltagc{div class="fancyvrblabel"}%
2026 \LWR@origextrm{\FV@LabelBegin}%
2027 \LWR@htmltagc{/div}
2028 \fi
2029     \else
2030       \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
2031       \ifx\FV@LabelPositionBottomLine\relax
2032 \else
2033           \LWR@htmltagc{div class="fancyvrblabel"}%
2034 \LWR@origextrm{\FV@LabelEnd}
2035 \LWR@htmltagc{/div}
2036 \fi
2037   \fi
2038
2039   \fi
2040   \hss
2041 }
2042 }
```

Processes each line, adding optional line numbers. Original code is from the `fancyvrb` package.

```
2043 \def\FV@ListProcessLine#1{%
2044   \hbox to \hsize{%
2045     \kern\leftmargin
2046     \hbox to \VerbatimHTMLWidth {%
2047       \ifcsvoid{FV@LeftListNumber}{}{\kern 2.5em}%
2048 \FV@LeftListNumber%
2049     \FV@LeftListFrame
2050   }}
```

```

2050      \FancyVerbFormatLine{#1}%
2051      \hss%
2052 %
2053      \FV@RightListFrame
2054 }%
2055      \hss% required to avoid underfull hboxes
2056 }
2057 }
```

Env **BVerbatim**

```

2058 \AtBeginEnvironment{BVerbatim}
2059 {
2060 \LWR@forcenewpage
2061 \LWR@tbeginverbatim{bverbatim}
2062
2063 }
2064
2065 \AfterEndEnvironment{BVerbatim}
2066 {
2067 \leavevmode\par\vspace{-\baselineskip}
2068 \LWR@afterendverbatim
2069 }
```

Env **LVerbatim** No changes required.

End of the modifications to make at the end of the preamble:

```
2070 } % \AfterEndPreamble
```

\UseVerbatim {\text{}}

No changes required.

```
2071 \end{warpHTML}
```

51 Theorems

```
\newtheorem {\text{}} [\text{counter}] -or- [\text{oldname}] {\text{}}
```

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: 2072 `\begin{warpHTML}`

```

\@begintheorem {⟨name⟩} {⟨number⟩}

2073 \renewcommand{\@begintheorem}[2]{%
2074 \LWR@forcenewpage
2075 \BlockClass{theoremcontents}
2076 \InlineClass{theoremlabel}{#1\ #2\ }%
2077 }

\@opargbegintheorem {⟨name⟩} {⟨number⟩} {⟨oparg⟩}

2078 \renewcommand{\@opargbegintheorem}[3]{%
2079 \LWR@forcenewpage
2080 \BlockClass{theoremcontents}
2081 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }%
2082 }

\@endtheorem

2083 \renewcommand*{\@endtheorem}{%
2084 \endBlockClass% theoremcontents
2085 }

2086 \end{warpHTML}

```

52 Lists

 **French** If using `babel` with French, use

```
\frenchbsetup{StandardLists=true}
```

to preserve the special HTML and `enumitem` list handling.

enumitem `enumitem` is pre-loaded during HTML output. Many of the spacing options are rendered irrelevant by `pdftotext` and HTML. Numbering, labels, and `\newlist` function correctly.

52.1 Itemize

for HTML output: 2087 `\begin{warpHTML}`

2088 \let\LWR@origitem\item

\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.

2089 \newcommand*\LWR@itemizeitem{%
 2090 \LWR@stoppars%
 2091 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem{}}
 2092 \LWR@htmlltag{li}%
 2093 \LWR@startpars%
 2094 \LWR@origitem%
 2095 }

To have a blank item, use \mbox{}. 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}
  ...
  
```

Env itemize [*enumitem options*]

2096 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
 2097
 2098 \newcommand*\LWR@itemizestart{%
 2099 \LWR@stoppars%
 2100 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}
 2101 \LWR@htmlltag{ul style="list-style-type:none"{}}
 2102 \LWR@startpars%
 2103 \let\item\LWR@itemizeitem%
 2104 }
 2105
 2106 \AtEndEnvironment{itemize}{\LWR@itemizeend}
 2107
 2108 \newcommand*\LWR@itemizeend{%
 2109 \LWR@stoppars%
 2110 \LWR@closeprevious{\LWR@depthlistitem}%
 2111 \LWR@closeoneprevious{}%
 2112 \LWR@startpars%
 2113 }

52.2 Enumerate

An HTML unordered list is used with customized L^AT_EX-generated labels.

```
Env  enumerate  [<enumitem options>]

2114 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
2115
2116 \newcommand*{\LWR@enumeratestart}{%
2117 \LWR@stoppars%
2118 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}
2119 \LWR@htmlltag{ul style="list-style-type:none"{}}
2120 \LWR@startpars%
2121 \let\item\LWR@itemizeitem%
2122 }
2123
2124
2125 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
2126
2127 \newcommand*{\LWR@enumerateend}{%
2128 \LWR@stoppars%
2129 \LWR@closeprevious{\LWR@depthlistitem}%
2130 \LWR@closeoneprevious{}%
2131 \LWR@startpars%
2132 }
```

52.3 Description

\LWR@descitem [<label>] Handles an \item inside a description.

```
2133 \newcommand*{\LWR@descitem}[1][]%
2134 {%
2135 \LWR@stoppars%
2136 \LWR@setlatestname{#1}%
2137 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem{}}
2138 \LWR@origitem[]%
```

Be sure the label doesn't print to the left of the rest of the file:

```
2139 \LWR@orighspace{1in}
2140 \LWR@htmlltag{dt}#1\LWR@htmlltag{/dt}%
2141 \LWR@orignewline%
2142 \LWR@htmlltag{dd}%
2143 \LWR@startpars%
2144 }
```

```
Env  description  [enumitem options] 
```

```

2145 \AtBeginEnvironment{description}{\LWR@descriptionstart}
2146
2147 \newcommand*{\LWR@descriptionstart}{%
2148 \LWR@stoppars%
2149 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}
2150 \LWR@htmlltag{d1}%
2151 \LWR@startpars%
2152 \let\item\LWR@descitem%
2153 }
2154
2155 \AtEndEnvironment{description}{\LWR@descriptionend}
2156
2157 \newcommand*{\LWR@descriptionend}{%
2158 \LWR@stoppars%
2159 \LWR@closeprevious{\LWR@depthlistitem}%
2160 \LWR@closeoneprevious{}%
2161 \LWR@startpars%
2162 } 
```

```
\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.

```

2163 \let\LWR@orignewlist\newlist
2164
2165 \renewcommand*{\newlist}[3]{%
2166 \LWR@orignewlist{#1}{#2}{#3}%
2167 \AtBeginEnvironment{#1}{\csuse{\LWR@#2start}}%
2168 \AtEndEnvironment{#1}{\csuse{\LWR@#2end}}%
2169 }

2170 \end{warpHTML} 
```

53 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

Limitations:

[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 l.
- tabularx ignores the width, but X columns do produce paragraph columns or multicolumns.

⚠ \multirow & \multicolumn

⚠ \multirow

\multirow with rules

rule at last row

⚠ paragraphs

\cmidrule width, trim

longtable headings

⚠ \warpprintonly

⚠ S columns

- Multirow and multicolumn cannot be used at the same time. (No rectangular holes wider than one column or taller than one row.)
- 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.
- If a multirow reaches to the bottom of a table, and \bottomrule does not go over to that edge, try adding a line of empty cells below the \bottomrule. This may be a browser bug.
- If a \midrule is desired after the last row, an additional row of blank cells must be used.
- Multiple paragraphs in one cell of a p, b, m column must have \newline between paragraphs.
- \cmidrule does not support width or trim options due to CSS limitations.
- For longtable, place headings and footings which do not apply to HTML inside \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.

- 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 l warp’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 \\}
```

53.1 Token lookahead

Used by `\LWR@futurenonospacelet` to look at the next token.

for HTML output: 2171 `\begin{warpHTML}`

```
\LWR@mynexttoken
```

```
2172 \newcommand{\LWR@mynexttoken}{\relax}
```

`\futurelet` copies the next token then executes a function to analyze

`\LWR@futurenonospacelet` does the same, but ignores intervening white space

Based on the booktabs style:

```
\LWR@futurenonospacelet
```

```
2173 \def{\LWR@futurenonospacelet}{\def{\LWR@cs{\#1}}%  
2174 {\afterassignment{\LWR@fnslone{\let{\nexttoken= }}  
2175 {\def{\LWR@fnslone{\expandafter{\futurelet{\LWR@cs{\LWR@fnsltwo}}%  
2176 {\def{\LWR@fnsltwo{\%  
2177 {\expandafter{\ifx{\LWR@cs{\@sp token}{\let{\next=\@BTfnslthree}}%  
2178 {\else{\let{\next=\nexttoken}\fi}\next}}  
2179 {\def{\@BTfnslthree{\afterassignment{\LWR@fnslone{\let{\next= }}
```

`\LWR@getmynexttoken` Looks ahead and copies the next token into `\LWR@mynexttoken`.

```
2180 \newcommand*{\LWR@getmynexttoken}{%  
2181 % nothing must follow this next line  
2182 {\LWR@futurenonospacelet{\LWR@mynexttoken{\LWR@tabledatacolumntag}}}  
2183 }
```

53.2 Booleans

Bool `LWR@startedrow` True if should print a row tag before this column.

```
2184 \newbool{\LWR@startedrow}  
2185 \boolfalse{\LWR@startedrow}
```

Bool `LWR@doinglehline` True if the next row will have an hline above it.

```
2186 \newbool{\LWR@doinglehline}  
2187 \boolfalse{\LWR@doinglehline}
```

Bool `LWR@doingtbrule` True if the next row will have a top/bottom rule above it.

```
2188 \newbool{LWR@doingtbrule}
2189 \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.

```
2190 \newbool{LWR@tableparcell}
```

Bool `LWR@skippingmrowcell` True if are doing an empty multi-row cell, and thus there is no data tag to close.

```
2191 \newbool{LWR@skippingmrowcell}
```

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.

```
2192 \newbool{LWR@intabularmetadata}
2193 \boolfalse{LWR@intabularmetadata}
```

53.3 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.

```
2194 \newcommand*{\LWR@closetabledatacell}{%
2195 \global\booltrue{LWR@intabularmetadata}%
2196 \ifbool{LWR@existingtabular}{}{%
2197 {}% not exiting tabular
2198 \ifbool{LWR@skippingmrowcell}{}{%
2199 {}% not skippingmrowcell}}
```

Insert any `<` then any `@` and `!` column contents:

```
2200 \unskip%
2201 \LWR@getexpparray{\LWR@col afterspec}{\theLWR@tablecolspos}%
2202 \LWR@getexpparray{\LWR@col atspec}{\theLWR@tablecolspos}%
2203 \LWR@getexpparray{\LWR@col bangspec}{\theLWR@tablecolspos}%
```

Close paragraphs:

```
2204 \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
2205 \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell:

```
2206 \unskip\LWR@htmlltag{/td}\LWR@orignewline%
2207 }% not skipping mrowcell
2208 }% not exiting tabular
2209 \global\boolfalse{LWR@skippingmrowcell}%
2210 }
```

`LWR@tabulardepth` tracks whether & is being used inside a `tabular`.

```
2211 \newcounter{LWR@tabulardepth}
2212 \setcounter{LWR@tabulardepth}{0}
2213
```

When not used inside a `tabular`, & performs its original function as recorded here (with catcode 4).

```
2214 \def\LWR@origampmacro{&}
```

See below for why the group is used.

```
2215 \begingroup
```

& 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.

The `\catcode` allows the & character to be redefined.

```
2216 \catcode`\&=\active
2217
2218 \gdef&{%
2219 \ifthenelse{\cnttest{\value{LWR@tabulardepth}}{>}{0}}{%
2220 {}}
```

If not skipping a multirow cell, close the current data cell.

```
2221 \unskip%
2222 \LWR@closetabledatacell%
```

Move to the next column.

```
2223 \addtocounter{LWR@tablecolspos}{1}%
```

Look at the next token to decide multi or single column data tag.

```
2224 \LWR@getmynexttoken%
2225 }%
```

If not inside a tabular, performs the original action:

```
2226 {\LWR@origampmacro}%
2227 }
2228 \endgroup
```

Outside the group, & is left its original catcode for now.

`tikz` package seems to require & be left alone until after `tikz` has been loaded.

`\LWR@lwarfStart` finally makes & active at the beginning of the HTML conversion.

53.4 Handling \\

Inside tabular, \\ is redefined to `\LWR@tabularendofline`

Throws away options \\[dim] or *

```
\LWR@tabularendofline
```

```
2229 \NewDocumentCommand{\LWR@tabularendofline}{s o}
2230 {%
2231 \LWR@closetabledatacell%
```

Finish the previous row:

```
2232 \LWR@htmlltag{/tr}\LWR@orignewline
2233 \global\booltrue{\LWR@intabularmetadata}
```

Not yet started a table row:

```
2234 \global\boolefalse{\LWR@startedrow}
```

Additional setup:

```
2235 \global\boolefalse{\LWR@doinghline}%
2236 \global\boolefalse{\LWR@doingtbrule}%
2237 \LWR@clearmidrules%
```

Start at first column:

```
2238 \setcounter{\LWR@tablecolspos}{1}
```

Look at the next token to decide between single column data tag or a special case:

```
2239 \LWR@getmynexttoken%
2240 }
```

53.5 Variables

```
2241 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
2242 \newcommand*{\LWR@pposition}{}%
2243 \newcommand*{\LWR@pleft}{}%
2244 \newcommand*{\LWR@pright}{}%
```

`\LWR@tablecolspec` Holds the parsed column specification, of total width `\LWR@tabletotalcols`.

Will contain a string such as `llrrccpc`, exactly one letter per column, without `@`, `>`, `<`, or the vertical pipe.

```
2245 \newcommand*{\LWR@tablecolspec}{}%
```

`\LWR@strresult` Holds the result of `Str` functions.

```
2246 \newcommand*{\LWR@strresult}{}%
```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

```
2247 \newcommand*{\LWR@origcolspec}{}%
```

Ctr `\LWR@tablecolswidth` Holds the width of the table specification.

(This is not the total # columns.)

```
2248 \newcounter{\LWR@tablecolswidth}
```

Ctr `\LWR@tablecolspos` Where are currently looking into the table column specification.

```
2249 \newcounter{\LWR@tablecolspos}
```

Ctr `\LWR@tabletotalcols` Holds the final number of table columns.

```
2250 \newcounter{\LWR@tabletotalcols}
```

Ctr `\LWR@tabletotalcolsnext` Holds the next column while parsing. Is one more than `\LWR@tabletotalcols`.

```
2251 \newcounter{\LWR@tabletotalcolsnext}
```

- LWR@colatspec A data array of specifications for @ columns. The leftmost's index is `leftheadge`, the others are counter values. See section 26.
- LWR@colbangspec A data array of specifications for ! columns. The leftmost's index is `leftheadge`, the others are counter values. See section 26.
- LWR@colbeforespec A data array of specifications for > columns.
- LWR@colafterspec A data array of specifications for < columns.

53.6 Parsing @, >, <, ! columns

\LWR@parseatcolumn Handles @{text} columns.

```
2252 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
2253 \LWR@traceinfo{at column}%
2254 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into \LWR@strresult, expanding once:

```
2255 \LWR@traceinfo{about to read the next token:}%
2256 \expandarg%
2257 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]
2258 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```
2259 \LWR@traceinfo{have now read the next token}%
2260 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}%
2261 {%
2262   \LWR@traceinfo{at the left edge}%
2263   \LWR@setexpparray{LWR@colatspec}{leftheadge}{\LWR@strresult}%
2264 }%
2265 {%
2266   \LWR@traceinfo{not at the left edge}%
2267   \LWR@setexpparray{LWR@colatspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
2268   \LWR@traceinfo{at \theLWR@tabletotalcols: \%}
2269   \LWR@getexpparray{LWR@colatspec}{\theLWR@tabletotalcols}!}%
2270 \let\LWR@strresult\relax%
2271 \booltrue{LWR@validtablecol}%
2272 }%
2273 }
```

\LWR@parsebangcolumn

2274 \newcommand*\{\LWR@parsebangcolumn}{%

Move to the next token after the '!':

2275 \LWR@traceinfo{bang column}%

2276 \addtocounter{LWR@tablecolspos}{1}%

Read the next token into \LWR@strresult, expanding once:

2277 \LWR@traceinfo{about to read the next token:}%

2278 \expandarg%

2279 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]

2280 \fullexpandarg%

Store the result into a data array, expanding once out of \LWR@strresult:

2281 \LWR@traceinfo{have now read the next token}%

2282 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}%

2283 {%
 left edge of the table:

2284 \LWR@traceinfo{at the left edge}%

2285 \LWR@setexpparray{LWR@colbangspec}{leftedge}{\LWR@strresult}%

2286 }%

2287 {%
 not at the left edge:

2288 \LWR@traceinfo{not at the left edge}%

2289 \LWR@setexpparray{LWR@colbangspec}{\theLWR@tabletotalcols}{\LWR@strresult}%

2290 \LWR@traceinfo{bang \theLWR@tabletotalcols: \LWR@colbangspec(\theLWR@tabletotalcols)!}%

2291 }%

2292 \let\LWR@strresult\relax%

2293 \booltrue{LWR@validtablecol}%

2294 }

\LWR@parsebeforecolumn Handles >{text} columns.

2295 \newcommand*\{\LWR@parsebeforecolumn}{%

Move to the next token after the '>':

2296 \addtocounter{LWR@tablecolspos}{1}%

Read the next token, expanding once into \LWR@strresult:

2297 \expandarg%

2298 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%

2299 \fullexpandarg%

Store the result into a data array, expanding once out of \LWR@strresult:

```

2300 \LWR@setexparray{\LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\LWR@strresult}%
2301 \let\LWR@strresult\relax%
2302 \booltrue{\LWR@validtablecol}%
2303 }

```

\LWR@parseaftercolumn Handles <{text} columns.

```
2304 \newcommand*{\LWR@parseaftercolumn}{%
```

Move to the next token after the '<':

```
2305 \addtocounter{\LWR@tablecolspos}{1}%


```

Read the next token, expanding once into \LWR@strresult:

```

2306 % \StrChar{\#1}{\theLWR@tablecolspos}[\LWR@strresult]
2307 \expandarg%
2308 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
2309 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@strresult:

```

2310 \LWR@setexparray{\LWR@colaferspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
2311 \let\LWR@strresult\relax%
2312 \booltrue{\LWR@validtablecol}%
2313 }

```

\LWR@parseskipcolumn Handles columns to skip, such as the vertical bar.

```

2314 \newcommand*{\LWR@parseskipcolumn}{%
2315 \booltrue{\LWR@validtablecol}%
2316 }

```

53.7 Parsing ‘l’, ‘c’, or ‘r’ columns

\LWR@parsenormalcolumn {<thiscolumn>}

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```

2317 \newcommand*{\LWR@parsenormalcolumn}[1]{%
2318 \appto{\LWR@tablecolspec{\#1}}%
2319 \addtocounter{\LWR@tabletotalcols}{1}%
2320 \addtocounter{\LWR@tabletotalcolsnext}{1}%
2321 \LWR@setexparray{\LWR@colatspec}{\theLWR@tabletotalcolsnext}{\relax}%

```

```

2322 \LWR@setexparray{\LWR@colbangspec}{\the\LWR@tabletotalcolsnext}{\relax}%
2323 \LWR@setexparray{\LWR@colbeforespec}{\the\LWR@tabletotalcolsnext}{\relax}%
2324 \LWR@setexparray{\LWR@colafterspec}{\the\LWR@tabletotalcolsnext}{\relax}%
2325 \booltrue{\LWR@validtablecol}%
2326 }

```

53.8 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn {<thiscolumn>}` The width will be ignored.

```
2327 \newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type.

```
2328 \LWR@parsenormalcolumn{#1}%
```

skips the following width

```

2329 \addtocounter{\LWR@tablecolspos}{1}%
2330 }

```

53.9 Parsing ‘D’ columns

From the `dcolumn` package.

`\LWR@parseDcolumn {<thiscolumn>}` The three parameters will be ignored.

```
2331 \newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
2332 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```

2333 \addtocounter{\LWR@tablecolspos}{3}%
2334 }

```

53.10 Parsing the column specifications

 HTML/CSS cannot exactly match the L^AT_EX concept of a baseline for a table row. Table 6 shows the L^AT_EX results for various vertical-alignment choices, with the

Table 6: Tabular baseline

	l	p	m	b	r
				bot	
			mid	bot	
l	par	mid	bot	bot	r
	par	mid			
	par				

baseline of the first column drawn across all the columns for comparison. See the p column specification in table 7 for details.

Table 7 describes how each kind of column is converted to HTML.

Bool LWR@validtablecol True if found a valid table column type.

```
2335 \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.

```
2336 \newcommand*{\LWR@parsetablecols}[1]{%
2337 \LWR@traceinfo{\LWR@parsetablecols started}%
}
```

Remember the original supplied column spec:

```
2338 \renewcommand*{\LWR@origcolspec}{\#1}%

```

Clear the parsed resulting column spec:

```
2339 \renewcommand*{\LWR@tablecolspec}{}%

```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
2340 \setcounter{LWR@tabletotalcols}{0}%
2341 \setcounter{LWR@tabletotalcolsnext}{1}%
2342 \LWR@setexpparray{\LWR@colatspec}{leftedge}{\relax}%
2343 \LWR@setexpparray{\LWR@colatspec}{1}{\relax}%
2344 \LWR@setexpparray{\LWR@colatspec}{2}{\relax}%
2345 \LWR@setexpparray{\LWR@colatspec}{3}{\relax}%

```

Table 7: 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 6, L^AT_EX 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 L^AT_EX 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.
-

```

2346 \LWR@setexpparray{\LWR@colbangspec}{leftedge}{\relax}%
2347 \LWR@setexpparray{\LWR@colbangspec}{1}{\relax}%
2348 \LWR@setexpparray{\LWR@colbangspec}{2}{\relax}%
2349 \LWR@setexpparray{\LWR@colbangspec}{3}{\relax}%
2350 \LWR@setexpparray{\LWR@colbeforespec}{1}{\relax}%
2351 \LWR@setexpparray{\LWR@colbeforespec}{2}{\relax}%
2352 \LWR@setexpparray{\LWR@colbeforespec}{3}{\relax}%
2353 \LWR@setexpparray{\LWR@colafterspec}{1}{\relax}%
2354 \LWR@setexpparray{\LWR@colafterspec}{2}{\relax}%
2355 \LWR@setexpparray{\LWR@colafterspec}{3}{\relax}%

```

Starting at the first column specification:

```
2356 \setcounter{\LWR@tablecolspos}{1}%
```

Place the colspecs string length into `\LWR@strresult`, and remember the number of characters in the column specification:

```

2357 \LWR@traceinfo{about to StrLen}%
2358 \noexpandarg%
2359 \StrLen{\#1}[\LWR@strresult]%
2360 \fullexpandarg%
2361 \LWR@traceinfo{finished StrLen}%
2362 \setcounter{\LWR@tablecolswidth}{\LWR@strresult}%

```

Scan through the column specifications:

```
2363 \whiledo{\not\value{\LWR@tablecolspos}>\value{\LWR@tablecolswidth}}{%
```

Place the next single-character column type into `\LWR@strresult`:

```

2364 \noexpandarg%
2365 \StrChar{\#1}{\the\LWR@tablecolspos}[\LWR@strresult]%
2366 \fullexpandarg%

```

Not yet found a valid column type

```
2367 \boolfalse{\LWR@validtablecol}%
```

Note that the parameter for a `p{spec}` column is a token list which will NOT match l,c,r,p.

```

2368 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{%
2369 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{%
2370 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{%
2371 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{%
2372 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{%
2373 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{%
2374 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{%

```

```

2375 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{}%
2376 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
2377 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
2378 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
2379 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
2380 \IfStrEq{\LWR@strresult}{|}{\LWR@parseskipcolumn}{}%
2381 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
2382 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
2383 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%

```

From the `dcolumn` package:

```
2384 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{}%
```

From the `tabularx` package. X column has no parameter, but will be given paragraph tags.

```
2385 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%
```

Many people define centered versions “P”, “M”, and “B”:
`\newcolumntype{P}[1]{>\centering\arraybackslash}p{#1}}`

```

2386 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
2387 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
2388 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%

```

If this column was an invalid column type, convert it to a p column:

```

2389 \ifbool{\LWR@validtablecol}{ }{%
2390 \LWR@parsenormalcolumn{1}}{%
2391 }{%
2392 \addtocounter{\LWR@tablecolspos}{1}}{%
2393 }{%
2394 }%

```

53.11 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.

```

2395 \newcommand*{\LWR@maybenewtablerow}{%
2396 {}%
2397 \ifbool{\LWR@startedrow}{ }{%
2398 {}% started the row
2399 {}% not started the row

```

Remember that now have started the row:

```
2400 \global\booltrue{LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```
2401 \global\booltrue{LWR@intabularmetadata}%
2402 \ifbool{LWR@doinghline}%
2403 {\LWR@htmlltag{tr class="hline"}\LWR@orignewline}%
2404 { % not doing hline
2405 \ifbool{LWR@doingtbrule}%
2406 {\LWR@htmlltag{tr class="tbrule"}\LWR@orignewline}%
2407 {\LWR@htmlltag{tr}\LWR@orignewline}%
2408 } % end of not doing hline
2409 } % end of not started the row
2410 }
```

53.12 Data opening tag

\LWR@tabledatasinglecolumntag Print a table data opening tag with style for alignment

```
2411 \newcommand*{\LWR@tabledatasinglecolumntag}%
2412 {%
2413 \LWR@maybenewtablerow%
```

If have found the end of tabular command, do not create the next data cell:

```
2414 \ifbool{LWR@exittingtabular}{}%
2415 { % not exiting tabular
```

Fetch the current column's alignment character into \LWR@strresult:

```
2416 \StrChar{\LWR@tablecols}{\theLWR@tablecolspos}[\LWR@strresult]%
```

print the start of a new table data cell:

```
2417 \LWR@htmlltag{td class="td%"
```

append this column's spec:

```
2418 \LWR@strresult%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag:

```
2419 \ifthenelse{\equal{\LWR@getexparray{LWR@midrules}}{\theLWR@tablecolspos}}{Y}{rule}{}%
2420 " {}%}
```

If this is a p, m, b, or X column, allow paragraphs:

```

2421 \ifthenelse{%
2422 \equal{\LWR@strresult}{p}\OR%
2423 \equal{\LWR@strresult}{m}\OR%
2424 \equal{\LWR@strresult}{b}\OR%
2425 \equal{\LWR@strresult}{P}\OR%
2426 \equal{\LWR@strresult}{M}\OR%
2427 \equal{\LWR@strresult}{B}\OR%
2428 \equal{\LWR@strresult}{X}%
2429 }%
2430 {%
2431 \allowpars%
2432 \global\booltrue{\LWR@tableparcell}%
2433 }%
2434 {}%
2435 }%

```

Print the @ and ! contents before first column, and then the > contents:

```

2435 \ifthenelse{\cnttest{\value{\LWR@tablecolspos}}=1}%
2436 {%
2437 \LWR@getexpparray{\LWR@colatspec}{leftedge}%
2438 \LWR@getexpparray{\LWR@colbangspec}{leftedge}%
2439 }%
2440 {}%
2441 \LWR@getexpparray{\LWR@colbeforespec}{\the\LWR@tablecolspos}%
2442 \global\boolfalse{\LWR@intabularmetadata}%
2443 }%
2444 }%

```

53.13 Midrules

LWR@midrules **LWR@midrules** is a data array (section 26) of columns containing Y if a midrule should be created for each column.

Ctr **LWR@midrulecounter** Indexes across the **LWR@midrules** data array.

```
2445 \newcounter{LWR@midrulecounter}
```

\LWR@clearmidrules Start new midrules. Called at beginning of tabular and also at \\.

Clears all **LWR@midrules** markers for this line.

```

2446 \newcommand*{\LWR@clearmidrules}%
2447 {%
2448 \setcounter{LWR@midrulecounter}{1}%

```

```

2449 \whiledo{%
2450 \cnttest{\value{LWR@midrulecounter}}{<=}{\value{LWR@tablecolswidth}}}%
2451 }%
2452 {%
2453 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{\relax}%
2454 \addtocounter{LWR@midrulecounter}{1}%
2455 }%
2456 }

```

\LWR@subcmidrule [*width*] {*trim*} {*leftcolumn*} {*rightcolumn*}

Marks LWR@midrules data array elements to be “Y” from left to right columns.

```

2457 \newcommand*{\LWR@subcmidrule}[4]{%
2458 \setcounter{LWR@midrulecounter}{#3}%
2459 \whiledo{\cnttest{\value{LWR@midrulecounter}}{<=}{#4}}{%
2460 {%
2461 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{Y}%
2462 \addtocounter{LWR@midrulecounter}{1}%
2463 }% end of the whiledo
2464 }

```

\LWR@docmidrule [*width*] {*trim*} {*leftcolumn-rightcolumn*}

Marks LWR@midrules array elements to be “Y” from left to right columns.

```

2465 \NewDocumentCommand{\LWR@docmidrule}{o d() >{\SplitArgument{1}{-}}m}{%
2466 {\LWR@subcmidrule{#1}{#2}{#3}}

```

53.14 Multicolumns

53.14.1 Parsing multicolumns

```

2467 \newcounter{LWR@tablemulticolswidth}
2468 \newcounter{LWR@tablemulticolspos}

```

\LWR@printmccoltype {*colspec*} Print any valid column type found. Does not print @, >, or < columns or their associated tokens.

This is printed as part of the table data tag’s class.

```

2469 \newcommand*{\LWR@printmccoltype}[1]{%
2470 \LWR@traceinfo{lwr@printmccoltype -#1-}%

```

Get one token of the column spec:

```

2471 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%

```

Add to the HTML tag depending on which column type is found:

```

2472 \IfStrEq{\LWR@strresult}{1}{\}%
2473 \IfStrEq{\LWR@strresult}{c}{\}%
2474 \IfStrEq{\LWR@strresult}{r}{\}%
2475 \IfStrEq{\LWR@strresult}{p}{\}%
2476 \IfStrEq{\LWR@strresult}{m}{\}%
2477 \IfStrEq{\LWR@strresult}{b}{\}%
2478 \IfStrEq{\LWR@strresult}{P}{\}%
2479 \IfStrEq{\LWR@strresult}{M}{\}%
2480 \IfStrEq{\LWR@strresult}{B}{\}%
2481 \IfStrEq{\LWR@strresult}{S}{\}%
2482 \IfStrEq{\LWR@strresult}{X}{\}%
2483 \LWR@traceinfo{lwr@printmccoltype done}%
2484 }
```

\LWR@multicolpartext Print the data with paragraph tags:

```

2485 \newcommand*{\LWR@multicolpartext}{%
2486 \LWR@startpars%
2487 \LWR@multicoltext%
2488 \LWR@stoppars%
2489 }
```

\LWR@multicolother {*colspec*} For @, >, <, print the next token without paragraph tags:

```

2490 \newcommand*{\LWR@multicolother}[1]{%
2491 \addtocounter{\LWR@tablemulticolspos}{1}%
2492 \StrChar{#1}{\the\LWR@tablemulticolspos}[\LWR@strresult]%
2493 \LWR@strresult}
```

A valid column data type was found:

```

2494 \booltrue{\LWR@validtablecol}%
2495 }
```

\LWR@multicolskip Nothing to print for this column type.

```
2496 \newcommand*{\LWR@multicolskip}{%
```

A valid column data type was found:

```

2497 \booltrue{\LWR@validtablecol}%
2498 }
```

\LWR@printmccoldata {*colspec*} Print the data for any valid column type found.

```
2499 \newcommand*{\LWR@printmccoldata}[1]{%
2500 \LWR@traceinfo{lwr@printmccoldata -#1}%
```

Not yet found a valid column type:

```
2501 \boolfalse{\LWR@validtablecol}%
```

Get one token of the column spec:

```
2502 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
2503 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
2504 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
2505 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
2506 \IfStrEq{\LWR@strresult}{D}{}%
2507 \addtocounter{\LWR@tablemulticolspos}{3}%
2508 \LWR@multicoltext%
2509 }{}%
2510 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolpartext}{}%
2511 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolpartext}{}%
2512 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolpartext}{}%
2513 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolpartext}{}%
2514 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolpartext}{}%
2515 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolpartext}{}%
2516 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolpartext}{}%
2517 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolpartext}{}%
2518 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
2519 \IfStrEq{\LWR@strresult}{@}{\detokenize{@}}{\LWR@multicolother{#1}}{}%
2520 \IfStrEq{\LWR@strresult}{!}{\detokenize{!}}{\LWR@multicolother{#1}}{}%
2521 \IfStrEq{\LWR@strresult}{>}{\detokenize{>}}{\LWR@multicolother{#1}}{}%
2522 \IfStrEq{\LWR@strresult}{<}{\detokenize{<}}{\LWR@multicolother{#1}}{}%
```

If an invalid column type:

```
2523 \ifbool{\LWR@validtablecol}{}{\LWR@multicoltext}%
```

Tracing:

```
2524 \LWR@traceinfo{lwr@printmccoldata done}%
2525 }
```

```
\parsemulticolumnalignment {\langle 1: colspec \rangle} {\langle 2: printresults \rangle}
```

Scan the multicolumn specification and execute the printfunction for each entry.

Note that the spec for a `p{spec}` column, or `@, >, <`, is a token list which will NOT match `l, c, r, or p`.

```
2526 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
2527 \setcounter{LWR@tablemulticolspos}{1}%
2528 \StrLen{#1}[\LWR@strresult]%
2529 \setcounter{LWR@tablemulticolswidth}{\LWR@strresult}%
```

Scan across the tokens in the column spec:

```
2530 \whiledo{%
2531 \not\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolswidth}%
2532 }%
2533 {%
```

Execute the assigned print function for each token in the column spec:

```
2534 #2{#1}%
```

Move to the next token in the column spec:

```
2535 \addtocounter{LWR@tablemulticolspos}{1}%
2536 }%
2537 }
```

53.14.2 High-level multicolumn interface

```
\LWR@domulticolumn {\langle 1: numcols \rangle} {\langle 2: colspec \rangle} {\langle 3: text \rangle}
```

```
2538 \newcommand{\LWR@multicoltext}{}%
2539 %
2540 \NewDocumentCommand{\LWR@domulticolumn}{m m +m}{%
2541 \LWR@traceinfo{lwr@domulticolumn -#1- -#2-}%
}
```

Remember the text to be inserted, and remember that a valid column type was found:

```
2542 \renewcommand{\LWR@multicoltext}{%
2543 #3%
2544 \booltrue{LWR@validtablecol}%
2545 }%
```

Row processing:

```
2546 \LWR@maybenewtablerow%
```

Begin the opening table data tag:

```
2547 \LWR@htmntag{td colspan="#1"
2548   class="td%"}
```

Print the column type:

```
2549 \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”.

```
2550 \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}}{\the\LWR@tablecolspos}}{Y}{rule}{}%"
```

Close the class tag’s opening quote:

```
2551 "%
2552 }% end of the opening table data tag
2553 \global\boolfalse{\LWR@intabularmetadata}%
2554 \LWR@parsemulticolumnalignment{#2}{\LWR@printmccoldata}%
2555 }
```

53.14.3 Longtable captions

Bool `\LWR@starredlongtable` Per the caption pacakge, step the counter if `longtable*`.

```
2556 \newbool{\LWR@starredlongtable}
2557 \boolfalse{\LWR@starredlongtable}
```

Per the `caption` package. User-redefinable float type.

```
2558 \providecommand*\LTcaptype{table}
```

```
\LWR@longtabledatacaptiontag * [<toc entry>] {<caption>}
```

```
2559 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
2560 {}%
```

Remember the latest name for `\nameref`:

```
2561 \IfValueTF{#2}{% optional given?
2562 \ifthenelse{\equal{#2}{}}% optional empty?
2563 {\LWR@setlatestname{#3}}% empty
2564 {\LWR@setlatestname{#2}}% given and non-empty
2565 }% optional given
2566 {\LWR@setlatestname{#3}}% no optional
```

create a multicolumn across all the columns

```
2567 \LWR@domulticolumn{\theLWR@tabletotalcols}{P}{%
2568 % \IfBooleanTF{#1}{ star?
2569 % {\IfValueTF{#2}{\LWR@origcaption*[#2]{#3}}{\LWR@origcaption*[#3]}}
2570 % {\IfValueTF{#2}{\LWR@origcaption[#2]{#3}}{\LWR@origcaption[#3]}}
2571 \IfBooleanTF{#1}{ star?
```

Star version, show a caption but do not make a LOT entry:

```
2572 {%
2573   yes star
2574   \LWR@htmlblocktag{figcaption}%
2574 #3%
2575   \LWR@htmlblocktag{/figcaption}%
2576 }%
2577 {%
2577   No star:
```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```
2578 \ifbool{LWR@starredlongtable}{%
2579 {%
2580 \ifthenelse{\equal{#2}{}}{%
2581 {}%
2582 {%
2583 \refstepcounter{LTcaptive}%
2584 \protected@edef\@currentlabel{%
2585 \csuse{p@\LTcaptive}\csuse{the\LTcaptive}}%
2586 {}%
2587 }}{}}
```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```
2588 \LWR@htmlblocktag{figcaption}%
2589 \csuse{fnum@\LTcaptive}\CaptionSeparator#3%
2590 \LWR@htmlblocktag{/figcaption}%
```

See if an optional caption was given:

```
2591 \ifthenelse{\equal{#2}{}}{%
2591   TOC entry empty}
```

if the optional caption was given, but empty, do not form a TOC entry

```
2592 {}%
```

If the optional caption was given, but might only be []:

```
2593 {%
2593   TOC entry not empty}
```

```
2594 \IfNoValueTF{#2}{% No TOC entry?}
```

The optional caption is []:

```
2595 {%
2596   \addcontentsline{%
2597     \csuse{ext@\LTcaptype}}{%
2598     \LTcaptype}}%
2599 {%
2600   \protect\newline}%
2601   {\csuse{p@\LTcaptype}\csuse{the\LTcaptype}}{%
2602     \ignorespaces #3\protect\relax}}%
2603 }%
2604 }% end of No TOC entry
```

The optional caption has text enclosed:

```
2605 {%
2606   \addcontentsline{%
2607     \csuse{ext@\LTcaptype}}{%
2608     \LTcaptype}}%
2609 {%
2610   \protect\newline}%
2611   {\csuse{p@\LTcaptype}\csuse{the\LTcaptype}}{%
2612     \ignorespaces #2\protect\relax}}%
2613 }%
2614 }% end of yes TOC entry
2615 }% end of TOC entry not empty
2616 }% end of no star
2617 }% end of \LWR@domulticolumn
2618
2619 \addtocounter{\LWR@tablecolspos}{\the\LWR@tabletotalcols}
2620 \addtocounter{\LWR@tablecolspos}{-1}
2621
2622 }
```

53.14.4 \tabledatamulticolumntag

```
\LWR@tabledatamulticolumntag {<numcols>} {<alignment>} {<text>}
2623 \NewDocumentCommand{\LWR@tabledatamulticolumntag}{m m +m}{%
2624 {%
2625   \LWR@domulticolumn{#1}{#2}{#3}{%
2626     \addtocounter{\LWR@tablecolspos}{#1}{%
2627     \addtocounter{\LWR@tablecolspos}{-1}{%
2628   }}
```

53.15 Multirow

Pkg **multirow**

```
\LWR@tabledatamultirowtag {\<numrows>} [\<bigstruts>] {\<width>} [\<fixup>] {\<text>}
2629 \NewDocumentCommand{\LWR@tabledatamultirowtag}{m o m o m}%
2630 {%
2631 \LWR@maybenewtablerow%
```

Print the start of a new table data cell:

```
2632 \LWR@htmlltag{td rowspan="#1" class="td%"}
```

Append this column's spec:

```
2633 \StrChar{\LWR@tablecolspe}{\theLWR@tablecolspos}%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag:

```
2634 \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}}{\theLWR@tablecolspos}}{Y}{rule}{}%
2635 "()}
```

While printing the text, redefine \\ to generate a new line

```
2636 \begingroup \let\\=\LWR@endofline #5 \endgroup
2637 \LWR@stoppars%
2638 \global\boolearn{\LWR@intabularmetadata}%
2639 }%
```

53.16 Utility macros inside a table

```
2640 \newcommand*{\LWR@donothing}{}
2641 \newcommand*{\LWR@domidrule}{\booltrue{\LWR@doinghline}}
2642 \newcommand*{\LWR@dotbrule}{\booltrue{\LWR@doingtbrule}}
```

53.17 Checking for a new table cell

\LWR@tabledatacolumntag Open a new HTML table cell unless the next token is for a macro which does not create data, such as `\hline`, `\toprule`, etc:

```
2643 \newbool{\LWR@exittingtabular}
2644 \newcommand*{\LWR@tabledatacolumntag}{%
2645 }%
```

\show\LWR@mynexttoken to see what tokens to look for

If not any of the below, start a new table cell:

```
2646 \let\mynext\LWR@tabledatasinglecolumntag%
```

If exiting the tabular:

```
2647 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\end}}{%
2648 {\booltrue{\LWR@exittingtabular}}{}}
```

`longtable` can have a caption in a cell

```
2649 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\caption}}{%
2650 {\let\mynext\LWR@donothing}{}}
```

Look for other things which would not start a table cell:

```
2651 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multicolumn}}{%
2652 {\let\mynext\LWR@donothing}{}}%
2653 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multirow}}{%
2654 {\let\mynext\LWR@donothing}{}}%
```

if come to an `\mrowcell`, this is a cell to be skipped over

```
2655 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\mrowcell}}{%
2656 {\let\mynext\LWR@donothing}{}}%
2657 %
2658 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\hline}}{%
2659 {\let\mynext\LWR@donothing}{}}%
2660 %
2661 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\toprule}}{%
2662 {%
2663 \let\mynext\LWR@donothing}{}}%
2664 %
2665 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\midrule}}{%
2666 {\let\mynext\LWR@donothing}{}}%
2667 %
2668 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cmidrule}}{%
2669 {\let\mynext\LWR@donothing}{}}%
2670 %
2671 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cline}}{%
2672 {\let\mynext\LWR@donothing}{}}%
2673 %
2674 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\bottomrule}}{%
2675 {\let\mynext\LWR@donothing}{}}%
2676 %
2677 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpprintonly}}
```

```
2678 {\let\mynext\LWR@donothing}{}%
2679 %
2680 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpHTMLonly}}%
2681 {\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:

```
2682 \mynext%
2683 }

2684 \end{warpHTML}
```

53.18 \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: 2685 \begin{warpall}
2686 \newcommand*{\mrowcell}{}%
2687 \end{warpall}

53.19 New \tabular definition

for HTML output: 2688 \begin{warpHTML}

Env LWR@tabular [*vertposition*] {[*colspecs*}]

The new tabular environment will be \let in \LWR@LwarpStart, since siunitx might redefine tabular in the user's document.

```
2689 \newenvironment*{\LWR@tabular}[2] []
2690 {%
2691 \LWR@traceinfo{tabular started}%
2692 \begingroup%
2693 \addtocounter{\LWR@tabulardepth}{1}%

```

Not yet started a table row:

```
2694 \global\boolfalse{\LWR@startedrow}%

```

Not yet doing an hline:

```
2695 \global\boolfalse{LWR@doinghline}%
```

Not yet doing a top/bottom rule:

```
2696 \global\boolfalse{LWR@doingtbrule}%
```

Have not yet found the end of tabular command:

```
2697 \boolfalse{LWR@exittingtabular}%
```

Create the **table** tag:

```
2698 \global\booltrue{LWR@intabularmetadata}%
```

```
2699 \LWR@forcenewpage
```

```
2700 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
2701 \LWR@parsetablecols{#2}%
```

Table col spec is: `\LWR@tablecols` which is a string of `llccrr`, etc.

Do not place the table inside a paragraph:

```
2702 \LWR@stopars%
```

Track column # for setting text-align:

```
2703 \setcounter{LWR@tablecolspos}{1}%
```

Start looking for midrules:

```
2704 \LWR@clearmidrules%
```

`\` becomes a macro to end the table row:

```
2705 \let\\LWR@tabularendofline%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with `\LWR@getmynexttoken` to see if the next token might create a new data cell:

```
2706 \renewcommand*{\hline}{\LWR@domidrule\LWR@getmynexttoken}%
```

```
2707 \newcommand*{\midrule}{\LWR@domidrule\LWR@getmynexttoken}%
```

```
2708 \NewDocumentCommand{\cmidrule}{o d() m}{}%
```

```
2709 {\LWR@docmidrule[##1] (##2){##3}\LWR@getmynexttoken}%
```

```

2710 \RenewDocumentCommand{\cline}{m}%
2711 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%
2712 \newcommand*{\toprule}{\LWR@dotbrule\LWR@getmynexttoken}%
2713 \newcommand*{\bottomrule}{\LWR@dotbrule\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`.

```

2714 \let\multicolumn\LWR@tabledatamulticolumntag%
2715 \let\multirow\LWR@tabledatamultirowtag%
2716 \renewcommand*{\mrowcell}{\global\booltrue{\LWR@skippingmrowcell}}%
2717 \let\caption\LWR@longtabledatacaptiontag%

```

Reset for new processing:

```

2718 \global\boolearn{\LWR@tableparcell}%
2719 \global\boolearn{\LWR@skippingmrowcell}%

```

Look ahead for a possible table data cell:

```

2720 \LWR@getmynexttoken%
2721 }%

```

Ending the environment:

```

2722 {%
2723 \LWR@closetabledatacell%
2724 \LWR@htmlblocktag{/tr}%
2725 \LWR@htmlblocktag{/table}%
2726 \global\boolearn{\LWR@intabularmetadata}%

2727 \addtocounter{\LWR@tabulardepth}{-1}%
2728 \endgroup%
2729 }

2730 \end{warpHTML}

```

53.20 Array

Pkg array

array is also automatically loaded by siunitx.

54 Cross-references

Sectioning commands have been emulated from scratch, so the cross-referencing commands are custom-written for them. Emulating both avoids several layers of patches.

The `zref` package is used to remember section name, file, and lateximage depth and number for each label.

Table 8 shows the data structures related to cross-referencing.

for HTML output: 2731 \begin{warpHTML}

54.1 Setup

\@currentlabelname To remember the most recently defined section name, description, or caption, for `\nameref`.

2732 \newcommand*\{@currentlabelname}{}%

\LWR@stripperiod {\text} [.]

Removes a trailing period.

2733 \def\LWR@stripperiod#1.\ltx@empty#2@nil{#1}%

\LWR@setlastname {\object name}

Removes `\label`, strips any final period, and remembers the result.

2734 \newcommand*\@LWR@setlastname}[1]{%

Remove `\label` and other commands from the name, the strip any final period.
See `zref-titleref` and `gettitlestring`.

```
2735 \GetTitleStringExpand{#1}%
2736 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
2737 \edef\@currentlabelname{%
2738 \expandafter\lwr@stripperiod\@currentlabelname%
2739 \ltx@empty.\ltx@empty@nil%
2740 }%
2741 }
```

Table 8: 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:	
\newlabel{<label>}{{\currentlabel}{\thepage}}	
\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 l warp:	(HTML only)
\label: Adds HTML tags (section 54.3), plus \splabel data (section 54.2):	
zLWR@name: The section name for this label.	
zLWR@htmlfilenumber: The filenumber or name for this label.	
zLWR@lateximagedepth: The lateximagedepth for this label.	
zLWR@lateximagenumber: The lateximagenumber for this label.	
\nameref: Emulated from hyperref for l warp. See section 54.4.	
\ref and \nameref: Adds HTML tags. See section 54.4.	
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 l warp,) later patched by cleverref.	
Added by cleverref:	(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 39.4 for use with footnotes.	
\label: Writes to the .aux file:	
\newlabel{<label>@\cref}{{\cref@currentlabel}{\thepage}}	
\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 l warp:	(HTML only)
\cref, etc.: Modified for l warp. See section 63.	
\label inside math: See section 58.4.1.	
Footnotes: See \noteentry in section 39.4.	

54.2 Zref setup

See:

<http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference>

Create a new property list called special:

```
2742 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
2743 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
2744 \zref@newprop{zLWR@htmlfilename}{%
2745 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilename}%
2746 }%
```

Additional properties for lateximages:

```
2747 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
2748 \zref@newprop{zLWR@lateximageindex}{\arabic{LWR@lateximageindex}}
```

`zLWR@htmlfilename` property holds the file number or name

Add a `LWR@htmlfilename` property, and `lateximage` properties to `special`:

```
2749 \zref@addprop{special}{zLWR@name}
2750 \zref@addprop{special}{zLWR@htmlfilename}
2751 \zref@addprop{special}{zLWR@lateximagedepth}
2752 \zref@addprop{special}{zLWR@lateximageindex}
```

Returns the selected field:

```
2753 \newcommand*{\LWR@sref}[2]{%
2754 \zref@extractdefault{#1}{#2}{??}}
```

`\LWR@nameref {<label>}` Returns the section name for this label:

```
2755 \newcommand*{\LWR@nameref}[1]{%
2756 \LWR@sref{#1}{zLWR@name}%
2757 }
```

`\LWR@htmlfileref {<label>}` Returns the file number for this label:

```

2758 \newcommand*{\LWR@htmlfileref}[1]{%
2759 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
2760 \LWR@sref{#1}{zLWR@htmlfilename}%
2761 }

```

\LWR@lateximagedepthref {*label*} Returns the `lateximagedepth` for this label:

```

2762 \newcommand*{\LWR@lateximagedepthref}[1]{%
2763 \LWR@sref{#1}{zLWR@lateximagedepth}%
2764 }

```

\LWR@lateximagenumberref {*label*} Returns the `lateximagenumber` for this label:

```

2765 \newcommand*{\LWR@lateximagenumberref}[1]{%
2766 \LWR@sref{#1}{zLWR@lateximagenumber}%
2767 }

```

\LWR@splabel {*label*} Sanitize the name and then creates the label:

```

2768 \newcommand*{\LWR@splabel}[1]{%
2769 \LWR@setlatestname{@currentlabelname}%
2770 \zref@labelbylist{#1}{special}}

```

54.3 Labels

\LWR@subsublabel {*label*} Creates an HTML id tag.

```

2771 \newcommand*{\LWR@subsublabel}[1]{%

```

Create an HTML id tag unless are inside a `lateximage`, since it would appear in the image:

```

2772 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}{>}{0}}{%
2773 {}%
2774 {% not lateximage

```

If not doing a `lateximage`, create an HTML ID tag: (To be factored...)

```

2775 \ifbool{\LWR@doingstartpars}{%
2776 {}% pars allowed
2777 \ifbool{\LWR@doingapar}{%
2778 {}% par started
2779 \LWR@htmntag{a id="#1"}\LWR@htmntag{/a}%
2780 }% par started
2781 {}% par not started

```

```

2782 \LWR@stoppars%
2783 \LWR@htmlltag{a id="#1"{} }\LWR@htmlltag{/a}%
2784 \LWR@startpars%
2785 }% par not started
2786 }% pars allowed
2787 { % pars not allowed
2788 \LWR@htmlltag{a id="#1"{} }\LWR@htmlltag{/a}%
2789 }% pars not allowed
2790 }% not lateximage
2791 }

```

`\LWR@newlabel {⟨label⟩} [⟨type⟩]`

`\label` during HTML output when not in math mode, removing extra spaces around the label, as done by regular L^AT_EX `\label`.

`cleveref` later encases this to add its own cross-referencing.

The optional `⟨type⟩` is per the `ntheorem` package, and is ignored.

```

2792 \NewDocumentCommand{\LWR@newlabel}{m o}{%
2793 \LWR@traceinfo{\LWR@newlabel: starting}%
2794 \LWR@traceinfo{\LWR@newlabel: !#1!}%
2795 % \@bsphack%

```

Create a traditional LaTeX label, as modified by `cleveref`:

`\LWR@origlabel{#1}%`

Create a special label which holds the section number, `LWR@htmlfilename`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```

2797 \LWR@traceinfo{\LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
2798 \LWR@traceinfo{\LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
2799 \LWR@traceinfo{\LWR@newlabel: LWR@htmlfilename is \theLWR@htmlfilename}%
2800 \LWR@splabel{#1}%
2801 \LWR@subsublabel{#1}%
2802 % \@esphack%
2803 \LWR@traceinfo{\LWR@newlabel: done}%
2804 }

```

54.4 References

`\LWR@startref {⟨label⟩}` (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, # character, and a label.

```

2805 \newcommand*{\LWR@startref}[1]
2806 {%
2807 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
2808 \LWR@traceinfo{\LWR@startref A: !#1!}%

```

Create the filename part of the link:

```

2809 \LWR@htmlltag{a href="%"
2810 \LWR@traceinfo{\LWR@startref B}%
2811 \LWR@htmllrefsectionfilename{#1}%
2812 \LWR@traceinfo{\LWR@startref C}%
2813 \#%

```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```

2814 \LWR@traceinfo{\LWR@startref D: !#1!}%
2815 \ifthenelse{\equal{\LWR@lidref}{??}}{%

```

“??” if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:

```

2816 {%
2817 \LWR@traceinfo{\LWR@startref DO: ??}%
2818 ??}%

```

If `LWR@lateximagedepth` is known. Use a `lateximage` if the depth is greater than zero, or a regular link otherwise:

```

2819 {%
2820 \LWR@traceinfo{\LWR@startref D1: \LWR@lidref}%
2821 \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}{%
2822 {%
2823 \LWR@traceinfo{\LWR@startref D2: \LWR@lidref}%
2824 lateximage\LWR@lateximagenumberref{#1}%
2825 }%
2826 {%
2827 \LWR@traceinfo{\LWR@startref D3}%
2828 #1%
2829 }%
2830 }%
2831 \LWR@traceinfo{\LWR@startref E}%

```

Closing quote:

```

2832 "}{}}%
2833 \LWR@traceinfo{\LWR@startref F}%
2834 }

```

\LWR@subnewref {*label*} {*label or sub@label*}

Factored for the `subfig` package. Uses the original label for the hyper-reference, but prints its own text, such as “1(b)”.

```
2835 \NewDocumentCommand{\LWR@subnewref}{m m}{%
2836 \LWR@traceinfo{\LWR@subnewref #1 #2}%
2837 \LWR@startref{#1}%
2838 \LWR@origref{#2}%
2839 \LWR@htmlltag{/a}%
2840 }
```

\ref * {*label*} \ref is \let to \LWR@newref

\LWR@newref * {*label*} Create an internal document reference link, or without a link if starred per `hyperref`.

```
2841 \NewDocumentCommand{\LWR@newref}{s m}{%
2842 \LWR@traceinfo{\LWR@newref #2}%
2843 \IfBooleanTF{#1}%
2844 {\LWR@origref{#2}}%
2845 {\LWR@subnewref{#2}{#2}}%
2846 }
```

\pagerefPageFor Text for starred page references.

```
2847 \newcommand*{\pagerefPageFor}{see }
```

\pageref * {*label*} Create an internal document reference, or just the unlinked number if starred, per `hyperref`.

```
2848 \NewDocumentCommand{\LWR@newpageref}{s m}{%
2849 \IfBooleanTF{#1}%
2850 {(\pagerefPageFor\LWR@origref{#2})}%
2851 {(\cpageref{#2})}%
2852 }
```

\nameref {*label*}

```
2853 \newcommand*{\nameref}[1]{%
2854 \LWR@traceinfo{\nameref A}%
2855 \LWR@startref{#1}%
2856 \LWR@traceinfo{\nameref B}%
2857 \LWR@nameref{#1}%
2858 \LWR@traceinfo{\nameref C}%
2859 \LWR@htmlltag{/a}%
2860 }
```

```
2860 \LWR@traceinfo{nameref D}%
2861 }
```

\Nameref {*label*} In print, adds the page number. In HTML, does not.

```
2862 \let\nameref\nameref
```

54.5 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.

```
2863 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF:
2864 % \EmulatesPackage{hyperref}[2015/08/01]%. Disabled. Do not do this.
```

Create a link with a text name:

```
\LWR@subhyperref {URL} {text}
```

```
2865 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
2866 \LWR@htmltag{a href="#" target="\_\_blank"\LWR@newline}\#2\LWR@htmltag{/a}%
2867 \LWR@ensuredoingapar%
2868 }
```

```
\LWR@subhyperrefclass {URL} {text} {htmlclass}
```

```
2869 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
2870 \LWR@htmltag{a href="#"%
2871 class="#3"\LWR@newline}\#2\LWR@htmltag{/a}%
2872 \LWR@ensuredoingapar%
2873 }
```

```
\href [options] {URL} {text}
```

Create a link with accompanying text:

```

2874 \NewDocumentCommand{\LWR@hrefb}{0{} m +m}{%
2875 \LWR@subhyperref{#2}{#3}%
2876 \endgroup%
2877 \LWR@ensuredoingapar%
2878 }%
2879
2880 \newcommand{\href}{%
2881 \LWR@ensuredoingapar%
2882 \begingroup%
2883 \catcode`\_=12%
2884 \LWR@hrefb%
2885 }

```

\nolinkurl {⟨URL⟩}

Print the name of the link without creating the link:

```

2886 \newcommand*{\LWR@nolinkurlb}[1]{#1\endgroup\LWR@ensuredoingapar}%
2887
2888 \newcommand{\nolinkurl}{%
2889 \LWR@ensuredoingapar%
2890 \begingroup\catcode`\_=12%
2891 \LWR@nolinkurlb%
2892 }

```

\url {⟨URL⟩}

Create a link whose text name is the address of the link:

```

2893 \newcommand*{\LWR@urlb}[1]{%
2894 \href{#1}{#1}%
2895 \endgroup%
2896 \LWR@ensuredoingapar%
2897 }%
2898
2899 \newcommand{\url}{%
2900 \LWR@ensuredoingapar%
2901 \begingroup\catcode`\_=12%
2902 \LWR@urlb%
2903 }

```

\LWR@subinlineimage [⟨alttag⟩] {⟨class⟩} {⟨filename⟩} {⟨extension⟩} {⟨style⟩}

```

2904 \newcommand*{\LWR@subinlineimage}[5][]{%
2905 \ifthenelse{\equal{#1}{}}{%
2906 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"{}}}%

```

```
2907 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{};}%  
2908 }
```

```
2909 \end{warpHTML}
```

Table 9: Float data structures

For each `<type>` of float (figure, table, etc.) there exists the following:

counter <type>: A counter called `<type>`, such as `figure`, `table`.

`\<type>name`: Name. `\figurename` prints “Figure”, etc.

`\ext@<type>`: File extension. `\ext@figure` prints “lof”, etc.

`\fps@<type>`: Placement.

`\the<type>`: Number. `\thetable` prints the number of the table, etc.

`\p@<type>`: Parent’s number. Prints the number of the [within] figure, etc.

`\fnum@<type>`: Prints the figure number for the caption.

`\<type>name \the<type>`, “Figure 123”.

`\<type>`: Starts the float environment. `\figure` or `\begin{figure}`

`\end<type>`: Ends the float environment. `\endfigure` or `\end{figure}`

`\tf@<ext>`: The L^AT_EX file identifier for the output file.

`LWR@have<type>`: A boolean rememebering 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.

55 Floats

Floats are supported, although partially through emulation.

Table 9 shows the data structure associated with each `<type>` of float.

`\@makecaption` is redefined to print the float number and caption text, separated by `\CaptionSeparator`, which works with the `babel` package to adjust the caption separator according to the language. French, for example, uses an en-dash instead of a colon: “Figure 123 – Caption text”.

55.1 Float captions

for HTML output: 2910 \begin{warpHTML}

```
\LWR@floatbegin {\langle type\rangle} [(placement)]
```

Begins a `\newfloat` environment.

```
2911 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
2912 \ifthenelse{\boolean{FormatWordProcessor}}{\AND\boolean{HTMLMarkFloats}}{%
2913
2914 === #1 begin
2915
2916 }{}%
2917 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
2918 \addtocounter{\LWR@thisfloat}{1}%
2919 \booltrue{\LWR@freezethisfloat}%

2920 \begingroup
```

Settings while inside the environment:

```
2921 \LWR@origraggedright
```

Open an HTML figure tag:

```
2922 \LWR@htmntag{figure id="autofloat-\arabic{\LWR@thisfloat}" class="#1"}%
2923 \renewcommand*{\@capttype}{#1}
2924 \caption@settype{#1}
2925 \LWR@startpars
2926 }
```

`\@float` Support packages which create floats directly.

```
\@dblfloat
2927 \let\@float\LWR@floatbegin
2928 \let\@dblfloat\LWR@floatbegin
```

`\LWR@floatend` Ends a `\newfloat` environment.

```
2929 \newcommand*{\LWR@floatend}{%
2930 \LWR@stoppars%
2931 \LWR@htmlelementend{figure}}%
```

```

2932 \endgroup%
2933 \boolfalse{LWR@freezethisfloat}%
2934 \LWR@startpars%
2935 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
2936
2937 === end
2938
2939 }{}%
2940 }

```

\end@float Support packages which create floats directly.

```

\end@dblfloat
2941 \let\end@float\LWR@floatend
2942 \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.

```
2943 \newcounter{LWR@thisfloat}
```

Bool LWR@freezethisfloat Prevents multiple increments of \LWR@thisfloat inside a float.

```

2944 \newbool{LWR@freezethisfloat}
2945 \boolfalse{LWR@freezethisfloat}

```

\LWR@maybeincthisfloat

```

2946 \newcommand*{\LWR@maybeincthisfloat}{%
2947 \ifbool{LWR@freezethisfloat}{}{\addtocounter{LWR@thisfloat}{1}}%
2948 }

```

\@captype Remembers which float type is in use.

```
2949 \newcommand*{\@captype}{}%
```

55.1.1 Caption inside a float environment

\CaptionSeparator How to separate the float number and the caption text.

```
2950 \AtBeginDocument{\providecommand*{\CaptionSeparator}{:~}}
```

\@makecaption {\langle name and num \rangle} {\langle text \rangle}

Prints the float type and number, the caption separator, and the caption text.

```
2951 \AtBeginDocument{\renewcommand{\@makecaption}[2]{#1\CaptionSeparator#2}}
```

55.1.2 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.

Ctr `LWR@nextautofloat` Tracks autofloat for floats. Tracks autopage for floats.

Ctr `LWR@nextautopage` These are updated per float as the `.lof` file is read.

```
2952 \newcounter{LWR@nextautofloat}
2953 \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.

```
2954 \newcommand*\{\LWRsetnextfloat}[2]{%
2955 \setcounter{LWR@nextautopage}{#1}%
2956 \setcounter{LWR@nextautofloat}{#2}%
2957 }
```

Ctr `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.

```
2958 \newcounter{LWR@latestautopage}
2959 \setcounter{LWR@latestautopage}{1}
```

```
2960 \let\LWR@origcaption@begin\caption@begin
2961 \let\LWR@origcaption@end\caption@end
2962 \let\LWR@orig@@par\@@par
```

`\LWR@caption@begin` Low-level patches to create HTML tags for captions.

```
2963 \newcommand{\LWR@caption@begin}{%
2964 %
2965 \LWR@traceinfo{LWR@caption@begin}%
}
```

Keep par and minipage changes local:

```
2966 \begingroup%
```

The caption code was not allowing the closing par tag:

```
2967 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a minipage or \parbox inside the caption:

```
2968 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{\LWR@minipage}{\LWR@minipage}%  
2969 \RenewDocumentCommand{\parbox}{0{t} o 0{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
2970 \LWR@htmlblocktag{figcaption}%  
2971 \LWR@origcaption@begin%  
2972 }
```

\LWR@caption@end Low-level patches to create HTML tags for captions.

```
2973 \newcommand{\LWR@caption@end}{%  
2974 \LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
2976 \vspace*{\baselineskip}%
```

Closing tag:

```
2977 \LWR@htmlblocktag{/figcaption}%  
2978 \endgroup%  
2979 % \leavevmode% avoid bad space factor (0) error  
2980 \LWR@traceinfo{\LWR@caption@end: done}%  
2981 }
```

\caption@begin Low-level patches to create HTML tags for captions.

```
\caption@end  
2982 \AtBeginDocument{  
2983 \let\caption@begin\LWR@caption@begin  
2984 \let\caption@end\LWR@caption@end  
2985 }
```

\captionlistentry Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```

2986 \let\LWR@origcaptionlistentry\captionlistentry
2987
2988 \renewcommand*{\captionlistentry}{%
2989 \LWR@maybeinthisfloat%
2990 \LWR@ensuredoingapar%
2991 \LWR@htmlltag{a id="autofloat-\arabic{\LWR@thisfloat}"}{\}\LWR@htmlltag{/a}}%
2992 \LWR@origcaptionlistentry%
2993 }
2994
2995 \def\LWR@LTcaptionlistentry{%
2996 \LWR@ensuredoingapar%
2997 \LWR@htmlltag{a id="autofloat-\arabic{\LWR@thisfloat}"}{\}\LWR@htmlltag{/a}}%
2998 \bgroup
2999 \ifstar{\egroup\LWR@LT@captionlistentry}{% gobble *
3000 {\egroup\LWR@LT@captionlistentry}}%
3001 \def\LWR@LT@captionlistentry#1{%
3002 \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.

```

3003 \let\LWR@origaddcontentsline\addcontentsline
3004
3005 \renewcommand*{\addcontentsline}[3]{%
3006 \ifthenelse{\equal{#1}{toc}}{}{%
3007 \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{}%
3008 \addtocontents{\@nameuse{ext@#2}}{%
3009 \protect\LWRsetnextfloat%
3010 {\arabic{\LWR@latestautopage}}%
3011 {\arabic{\LWR@thisfloat}}%
3012 }% addtocontents
3013 }% not toc
3014 \LWR@origaddcontentsline{#1}{#2}{#3}%
3015 }

```

\captionof Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

3016 \AtBeginDocument{
3017 \let\LWR@origcaptionof\captionof
3018
3019 \renewcommand*{\captionof}{%
3020 \LWR@maybeinthisfloat%
3021 \LWR@stoppars
3022 \LWR@htmlltag{a id="autofloat-\arabic{\LWR@thisfloat}"}{\}\LWR@htmlltag{/a}}%
3023 \LWR@origcaptionof%
3024 }

```

```
3025 }

3026 \end{warpHTML}
```

56 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 L^AT_EX infrastructure is used for TOC, along with some patches to generate HTML output.

for HTML output: 3027 `\begin{warpHTML}`

56.1 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 L^AT_EX, 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.

```
3028 \newcommand*{\LWR@myshorttoc}[1]{
3029 \LWR@ensuredoingapar
```

Only if the file exists:

```
3030 \IfFileExists{\jobname.#1}{
```

 Make @ a regular letter. Many of the commands in the file will have @ characters in them, so @ must be made a regular letter.

 For pdflatex, also change to latin1 encoding. When reading back a file with accented characters, the encoding change seems to be required, rather than leaving it utf8.

```
3031 \begingroup
3032 % \ifxetexorluatex%
3033 % \else
3034 % \inputencoding{latin1}% currently disabled
3035 % \fi
3036 \makeatletter
```

Read in the TOC file:

```
3037 \@input{\jobname.\#1}
3038 % \makeatother
3039 \endgroup
3040 }%
3041 {}%
3042 }
```

\LWR@subtableofcontents {⟨toc/lof/lot⟩} {⟨sectionstarname⟩}

Places a TOC/LOF/LOT at the current position.

```
3043 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```
3044 \@ifundefined{chapter}
3045 {\LWR@closeprevious{\LWR@depthsection}}
3046 {\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
3047 \LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
3048 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}
```

Create a new HTML `nav` containing the TOC/LOF/LOT:

```
3049 \LWR@htmlelementclass{nav}{#1}
```

Create the actual list:

```
3050 \LWR@myshorttoc{#1}
```

Close the `nav`:

```
3051 \LWR@htmlelementclassend{nav}{#1}
3052 }
```

Patch \starttoc to encapsulate the TOC inside HTML tags:

```
3053 \let\LWR@orig@starttoc\starttoc
3054
3055 \renewcommand{\starttoc}[1]{
3056 \LWR@htmlelementclass{nav}{#1}
3057 \LWR@orig@starttoc{#1}
3058 \LWR@htmlelementclassend{nav}{#1}
3059 }
```

Patch \tableofcontents, etc. to print footnotes first. newfloat uses \listoffigures for all future float types.

```
3060 \let\LWR@origtableofcontents\tableofcontents
3061 \let\LWR@origlistoffigures\listoffigures
3062 \let\LWR@origlistoftables\listoftables
3063
3064 \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:

```
3065 \ifbool{FormatWordProcessor}{}{
```

Copy the .toc file to .sidetoc for printing the sidetoc. The original .toc file is renewed when \tableofcontents is finished.

```
3066 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
3067 \LWR@printpendingfootnotes
3068 \LWR@origtableofcontents
3069 }
3070 }
3071 \renewcommand*\listoffigures{}{
3072 \ifbool{FormatWordProcessor}{}{
3073 \LWR@printpendingfootnotes
3074 \LWR@origlistoffigures
3075 }
3076 }
3077
3078 \renewcommand*\listoftables{}{
3079 \ifbool{FormatWordProcessor}{}{
3080 \LWR@printpendingfootnotes
3081 \LWR@origlistoftables
3082 }
3083 }
```

56.2 High-level TOC commands

`\listof {⟨type⟩} {⟨title⟩}`

Emulate the `\listof` command from the `float` package (section 95). Used to create lists of custom float types. Also used to redefine the standard L^AT_EX `\listoffigures` and `\listoftables` commands.

```
3084 \NewDocumentCommand{\listof}{m +m}{%
3085   \LWR@subtableofcontents{@nameuse{ext@#1}}{#2}%
3086   \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname%
3087   \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname%
3088   \jobname.\csuse{ext@#1}\relax%
3089 }
```

56.3 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.

```
3090 \end{warpHTML}
```

for HTML & PRINT: 3091 `\begin{warpall}`

Ctr SideTOCDepth Controls how deep the side-TOC gets. Use a standard L^AT_EX section level similar to `tocdepth`.

```
3092 \newcounter{SideTOCDepth}%
3093 \setcounter{SideTOCDepth}{1}
```

`\sidetocname` Holds the default name for the sidetoc.

```
3094 \newcommand{\sidetocname}{Contents}
```

3095 \end{warpall}

for HTML output: 3096 \begin{warpHTML}

\LWR@sidetoc Creates the actual side-TOC.

3097 \newcommand*\LWR@sidetoc{
 3098 \LWR@forcenewpage
 3099 \LWR@stoppars
 3100

The entire sidetoc is placed into a nav of class `sidetoc`.

3101 \LWR@htmlelementclass{nav}{sidetoc}
 3102
 3103 \setcounter{tocdepth}{\value{SideTOCDepth}}
 3104

The title is placed into a div of class `sidetoctitle`, and may contain paragraphs.

3105 \begin{BlockClass}{sidetoctitle}
 3106 \sidetocname
 3107 \end{BlockClass}

The table of contents is placed into a div of class `sidetoccocontents`.

3108 \begin{BlockClass}{sidetoccocontents}
 3109 \LinkHome
 3110
 3111 \LWR@myshorttoc{sidetoc}
 3112 \end{BlockClass}
 3113 \LWR@htmlelementclassend{nav}{sidetoc}
 3114 }

56.4 Low-level TOC line formatting

\numberline {⟨number⟩}

(Called from each line in the .aux, .lof files.)

Record this section number for further use:

3115 \renewcommand*\numberline[1]{%
 3116 \LWR@sectionnumber{\#1}%
 3117 }

```
\hypertoc {<1: depth>} {<2: type>} {<3: name>} {<4: page>}
```

Called by \l@**section**, etc. to create a hyperlink to a section.

The autopage label is always created just after the section.

#1 is depth

#2 is **section**, **subsection**, etc.

#3 the text of the caption

#4 page number

```
3118 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
3119 \ifthenelse{\cnttest{#1}{<=}{\value{tocdepth}}}{%
3120 \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```
3121 \LWR@subhyperrefclass{%
3122 \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
3123 \LWR@stoppars%
3124 }%
3125 {}%
3126 }
```

Ctr **lofdepth** TOC depth for figures.

```
3127 \newcounter{lofdepth}
3128 \setcounter{lofdepth}{1}
```

Ctr **lotdepth** TOC depth for tables.

```
3129 \newcounter{lotdepth}
3130 \setcounter{lotdepth}{1}
```

```
\hypertocfloat {<1: depth>} {<2: type>} {<3: ext of parent>} {<4: caption>} {<5: page>}
```

#1 is depth

#2 is **figure**, **table**, etc.

#3 is **lof**, **lot**, of the parent.

#4 the text of the caption

#5 page number

```
3131 \newcommand{\hypertocfloat}[5]{%
3132 \LWR@startpars%
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
3133 \@ifundefined{c@#3depth}{%
3134 \newcounter{#3depth}%
3135 \setcounter{#3depth}{1}%
3136 }{}%
```

Respond to `lofdepth`, etc.:

```
3137 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
3138 \ifthenelse{\cnttest{#1}{<=}{\arabic{#3depth}}}{%
3139 \LWR@startpars%
```

Create an HTML link to `filename#autofloat-(float number)`, with text of the caption, of the given HTML class.

```
3140 \LWR@subhyperrefclass{%
3141 \LWR@htmrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
3142 \#autofloat-\arabic{LWR@nextautofloat}}%
3143 {#4}{toc#2}%
3144 \LWR@stoppars%
3145 }{}%
3146 }
```

Automatically called by `\contentsline`:

```
3147 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
3148 \DeclareDocumentCommand{\l@chapter}{m m}
3149 {\hypertoc{0}{chapter}{#1}{#2}}
3150 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
3151 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
3152 \renewcommand{\l@subsubsection}[2]
3153 {\hypertoc{3}{subsubsection}{#1}{#2}}
3154 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
3155 \renewcommand{\l@ subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
3156 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
3157 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}
```

3158 \end{warpHTML}

57 Index and glossary

See:

[http://tex.stackexchange.com/questions/187038/
how-to-mention-section-number-in-index-created-by-imakeidx](http://tex.stackexchange.com/questions/187038/how-to-mention-section-number-in-index-created-by-imakeidx)

Index links are tracked by the counter `LWR@autoindex`. This counter is used to create a label for each index entry, and a reference to this label for each entry in the index listing. This method allows each index entry to link directly to its exact position in the document.

for HTML output: 3159 `\begin{warpHTML}`

```
3160 \newcounter{LWR@autoindex}
3161 \setcounter{LWR@autoindex}{0}
3162
3163 \newcounter{LWR@autoglossary}
3164 \setcounter{LWR@autoglossary}{0}
```

`\LWR@indexsection` Controls whether the index will be in a section or a chapter, depending on the `documentclass`.

```
3165 \@ifundefined{chapter}
3166 {\newcommand*{\LWR@indexsection}{\section{\indexname}}}
3167 {\newcommand*{\LWR@indexsection}{\chapter{\indexname}}}
```

`\printindex`

```
3168 \let\LWR@origprintindex\printindex
3169
3170 \renewcommand*{\printindex}
3171 {
3172 \LWR@indexsection
3173 \LWR@startpars
3174 \LWR@origprintindex
3175 }
```

`Env theindex`

```
3176 \renewenvironment*{\theindex}{%
3177 \let\item\LWR@indexitem%
3178 \let\subitem\LWR@indexsubitem%
3179 \let\subsubitem\LWR@indexsubsubitem%
3180 }{}}
```

`\LWR@indexitem`

```

3181 \newcommand{\LWR@indexitem}{%
3182   \InLineClass{indexitem}{}%
3183 }%
3184 }%

\LWR@indexitem

3185 \newcommand{\LWR@indexsubitem}{%
3186   \InLineClass{indexsubitem}{}%
3187 }%
3188 }%

\LWR@indexitem

3189 \newcommand{\LWR@indexsubsubitem}{%
3190   \InLineClass{indexsubsubitem}{}%
3191 }%
3192 }%

\@wrindex {term}      Redefined to write the LWR@latestautopage counter instead of page

3193 \def\@wrindex#1{%
3194   \addtocounter{LWR@autoindex}{1}%
3195   \LWR@newlabel{LWRindex-\theLWR@autoindex}%
3196   \protected@write\@indexfile{%
3197     {\string\indexentry{#1}{\theLWR@autoindex}}%
3198   }%
3199 }%
3200 @esphack}%

\@wrglossary {term}      Redefined to write the LWR@latestautopage counter instead of page

3200 \def\@wrglossary#1{%
3201   \addtocounter{LWR@autoglossary}{1}%
3202   \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
3203   \protected@write\@glossaryfile{%
3204     {\string\glossaryentry{#1}{\theLWR@autoglossary}}%
3205   }%
3206 }%
3207 @esphack}%

\hyperindexref {autosecnumber}

\hyperindexref{web address} is inserted into *.ind by the xindy style file
lwarf_html.xdy

3207 \newcommand*\{\hyperindexref}[1]{
```

```

3208 \nameref{LWRindex-#1}%
3209 }

3210 \end{warpHTML}

```

for PRINT output: A null command for print mode, in case `hyperref` was not used:

```

3211 \begin{warpprint}
3212 \newcommand{\hyperindexref}[1]{#1}
3213 \end{warpprint}

```

58 Math

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 L^AT_EX 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 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 tag 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

⁹See section 165 regarding fonts and fractions.

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 L^AT_EX math expression to manually insert tags, but this has not yet been done.

footnotes in math

lateximage

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

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

siunitx

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

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

- MathJax does not automatically support custom L^AT_EX macros, but they may be set up by the user.

⚠ siunitx inside an equation

L^AT_EX macros

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: 3214 `\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.

```
3215 \let\LWR@origtextdollar\\$
3216
3217 \renewcommand*{\$}{%
3218 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}>}{0}{%
3219 {\LWR@origtextdollar}%
3220 {\LWR@htmlltagc{span}\LWR@origtextdollar\LWR@htmlltagc{/span}}%
3221 }}
```

Ctr `LWR@externalfilecnt` Counter for the external files which are generated and then referenced from the HTML:

```
3222 \newcounter{LWR@externalfilecnt}
```

58.1 Inline and display math

```
3223 \let\LWR@origdollar=\\$%
3224 \let\LWR@secondorigdollar=\\$% balance for editor syntax highlighting
3225 \let\LWR@origopenparen\\(
3226 \let\LWR@origcloseparen\\)
3227 \let\LWR@origopenbracket\\[
3228 \let\LWR@origclosebracket\\]
```

\\\$ Redefine the dollar sign to place math inside a `lateximage`, or use MathJax:
\\\$\\\$

```

3229 \begingroup
3230 \catcode`\$=\active%
3231 \protected\gdef${\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

`\LWR@doubledollar` Redefine the double dollar sign to place math inside a `lateximage`, or use MathJax:

```

3232 \gdef\LWR@doubledollar##1$${
3233 \ifbool{mathjax}{%

```

For MathJax, print the math between `\[` and `\]`:

```

3234 {\textbackslash[LWR@HTMLsanitize{\#1}\textbackslash]}

```

For SVG, print the math inside a `lateximage`, with an ALT tag of the L^AT_EX code:

```

3235 {%
3236   \begin{math}
3237     \begin{lateximage}%
3238       \textbackslash[LWR@HTMLsanitize{\#1}\textbackslash]{}%
3239     \LWR@origdollar\textbackslash[LWR@origdollar#1\textbackslash[LWR@origdollar\textbackslash[LWR@origdollar%
3240   \end{math}
3241 }
3242 }%
3243 }%

```

`\LWR@singledollar` Redefine the single dollar sign to place math inside a `lateximage`, or use MathJax:

```

3244 \gdef\LWR@singledollar#1${%
3245 \ifbool{mathjax}{%

```

For MathJax, print the math between `\(` and `\)`:

```

3246 {\textbackslash[LWR@HTMLsanitize{\#1}\textbackslash]}

```

For SVG, print the math inside a `lateximage`, with an ALT tag of the L^AT_EX code:

```

3247 {%
3248   \begin{math}
3249     \begin{lateximage}%
3250       \textbackslash[LWR@HTMLsanitize{\#1}\textbackslash]{}%
3251     \LWR@origdollar#1\textbackslash[LWR@origdollar%
3252   \end{math}
3253 }%

```

`\(`` Redefine to the above dollar macros.

`\``

```

3254 \gdef`{\#1`}{\$#1\$}
3255 \gdef`{\#1`}{\$\$#1\$\$}

```

```
3256
3257 \endgroup
```

Remove the old `math` and `displaymath` environments:

```
3258 \let\math\relax
3259 \let\endmath\relax
3260 \let\displaymath\relax
3261 \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`.

```
3262 \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`.

```
3263 \NewEnviron{displaymath}{\expandafter[\BODY]\@ignoretrue}
```

When the document begins, the dollar sign must be made active to trigger the new math macros:

```
3264 \AtBeginDocument{\catcode`\$=\active}
```

58.2 MathJax support

Ctr `LWR@nextequation` Used to add one to compute the next equation number.

```
3265 \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 L^AT_EX.

```
3266 \newcommand*\LWR@syncmathjax{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
3267 \ifcsdef{thechapter}{
3268 \BlockClassSingle{hidden}{
3269 \textbackslash(
```

```

3270 \textbackslash{}seteqsection \{\thechapter\}
3271 \textbackslash{})
3272 }
3273 }
3274 {}% not using chapters
3275

```

MathJax doesn't allow setting the equation number to 1:

```

3276 \ifthenelse{\cnttest{\value{equation}}>0}{}
3277 {

```

Tell MathJax that the next set of equations begins with the current L^AT_EX equation number, plus one.

```

3278 \setcounter{LWR@nextequation}{\value{equation}}
3279 \addtocounter{LWR@nextequation}{1}

```

Place the MathJax command inside “\(`” and “\)`” characters, to be printed to HTML, not interpreted by L^AT_EX.

```

3280 \BlockClassSingle{hidden}{%
3281 \textbackslash{}(
3282 \textbackslash{}seteqnumber {\arabic{LWR@nextequation}}%
3283 \textbackslash{})}
3284 }
3285 }% not eq > 1
3286 }

```

\LWR@restoremathlatexformatting While producing math, use regular L^AT_EX formatting instead of HTML tags.

```

3287 \newcommand*{\LWR@restoremathlatexformatting}{%
3288 \let\hspace{\LWR@origspace}%
3289 \let\rule{\LWR@origrule}%
3290 \let\,,\LWR@origcomma% disable HTML short unbreakable space
3291 \let\textit{\LWR@origtextit}%
3292 \let\textbf{\LWR@origtextbf}%
3293 \let\texttt{\LWR@origtexttt}%
3294 \let\textsc{\LWR@origtextsc}%
3295 \let\textsf{\LWR@origtextsf}%
3296 \let\textrm{\LWR@origtextrm}%
3297 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
3298 \let\textsuperscript{\LWR@origtextsuperscript}%
3299 \let\textsubscript{\LWR@origtextsubscript}%
3300 \let~\LWR@origtilde%
3301 \let\enskip\LWR@origenskip%
3302 \let\quad\LWR@origquad%
3303 \let\qquad\LWR@origqquad%
3304 }

```

```
\LWR@hidelatexequation {⟨environment⟩} {⟨contents⟩}
```

Creates the L^AT_EX version of the equation inside an HTML comment.

```
3305 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```
3306 \LWR@stoppars
3307 \LWR@htmlopencomment
3308
```

Start the LaTe_X math environment inside the HTML comment:

```
3309 \begingroup
3310 \csuse{\LWR@orig#1}
```

While in the math environment, restore various commands to their L^AT_EX meanings.

```
3311 \LWR@restoremathlatextformatting
```

See \LWR@htmlmathlabel in section 58.4.1.

Print the contents of the equation:

```
3312 #2
```

End the L^AT_EX math environment inside the HTML comment:

```
3313 \csuse{\LWR@origend#1}
3314 \endgroup
3315
```

Close the HTML comment and resume HTML paragraph handling:

```
3316 \LWR@htmclosecomment
3317 \LWR@startpars
3318 }
```

```
\LWR@addmathjax {⟨environment⟩} {⟨contents⟩}
```

Given the name of a math environment and its contents, create a MathJax instance. The contents are printed to HTML output, not interpreted by L^AT_EX.

```
3319 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
3320 }
```

Enclose the MathJax environment inside printed “\((” and “\))” characters.

```
3321 \textbackslash{}begin\{\#1\}
```

Print the contents, sanitizing for HTML special characters.

```
3322 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\#2}}
```

Close the MathJax environment:

```
3323 \textbackslash{}end\{\#1\}
```

```
3324
```

```
3325 }
```

58.3 Equation environment

Remember existing `equation` environment:

```
3326 \let\origequation\equation
3327 \let\origendequation\endequation
```

Remove existing `equation` environment:

```
3328 \let\equation\relax
3329 \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.

```
3330 \NewEnviron{equation}{%
3331
3332 \ifbool{mathjax}{
```

MathJax output:

```
3333 {
```

Print commands to synchronize MathJax's equation number and format to the current L^AT_EX chapter/section and equation number:

```
3334 \LWR@syncmathjax
```

Print the L^AT_EX math inside an HTML comment:

```
3335 \LWR@hidelatexequation{equation}{\BODY}
3336 }
```

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`.

```
3337 {%
  not mathjax
```

Begin the `lateximage` with an ALT tag containing the math source:

```
3338 \begin{lateximage}[(\theequation) \textbackslash\begin\{equation\}\}%
3339 \LWR@HTMLsanitizeexpand{\detokenize\expandafter\{\BODY\}\}%
3340 \textbackslash\end\{equation\}\}]% alt tag
```

Create the actual L^AT_EX-formatted equation inside the `lateximage` using the contents of the environment.

```
3341 \LWR@origequation
3342 \BODY% contents collected by NewEnviron
3343 \LWR@origendequation
3344 \end{lateximage}%
3345 }
3346
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
3347 }[\ifbool{mathjax}{\LWR@addmathjax{equation}{\BODY}}{}]
```

58.4 AMS Math environments

58.4.1 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 it is used in an `amsmath` environment which is not a `multiline` environment and not an `equation`.

```
3348 \newbool{LWR@amsmultiline}
3349 \boolfalse{LWR@amsmultiline}
```

```
\LWR@htmlmathlabel {⟨label⟩}
```

`lwarp` points `\ltx@label` here. This is used by `\label` when inside a L^AT_EX AMS math environment's math display environment.

`\LWR@origltx@label` points to the L^AT_EX original, modified by `lwarp`, then by `amsmath`, then by `cleveref`.

```
3350 \newcommand*{\LWR@htmlmathlabel}[1]{%
3351 \LWR@traceinfo{\LWR@htmlmathlabel #1}%
3352 \ifbool{mathjax}{%
```

The combined L^AT_EX & HTML label is printed in a `\text` field:

```
3353 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
3354 \ifbool{LWR@amsmultiline}{}{\hspace*{\totwidth@}}
```

Temporarily end the HTML comment, insert the L^AT_EX & HTML label, then resume the HTML comment. `\Qfirstofone` is required to remove extra braces introduced by the `amsmath` package.)

```
3355 \LWR@htmclosecomment%
3356 \LWR@origltx@label{#1}%
3357 \LWR@htmopencomment%
3358 }% text
3359 }% mathjax
3360 {%
3361 \LWR@origltx@label{#1}%
3362 }%
3363 }
```

`\LWR@beginhideamsmath` Starts hiding L^AT_EX math inside an HTML comment.

```
3364 \newcommand*{\LWR@beginhideamsmath}{%
3365 \LWR@stoppars
3366
3367 \LWR@htmopencomment
3368
3369 \begingroup
3370 \LWR@restoremathlatexformatting
3371 }
```

`\LWR@endhideamsmath` Ends hiding L^AT_EX math inside an HTML comment.

```

3372 \newcommand*{\LWR@endhideamsmath}{
3373 \endgroup
3374
3375 \LWR@htmlclosecomment
3376
3377 \LWR@startpars
3378 }

```

58.4.2 Environment patches

The following `amsmath` environments already collect their contents in `\@envbody` for further processing.

For SVG math: Each environment is encapsulated inside a `lateximage` environment, along with a special `LWR@AMSMATHBODY` 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  multiline

3379 \BeforeBeginEnvironment{multiline}{
3380 \ifbool{mathjax}
3381 {
3382 \LWR@syncmathjax
3383 \booltrue{\LWR@amsmultiline}
3384 \LWR@beginhideamsmath
3385 }
3386 {
3387 \latexitimage[LWR@AMSMATHBODY]
3388 }
3389 }
3390
3391 \AfterEndEnvironment{multiline}{
3392
3393 \ifbool{mathjax}
3394 {
3395 \LWR@endhideamsmath
3396 \boolfalse{\LWR@amsmultiline}
3397 \LWR@addmathjax{multiline}{\the\@envbody}
3398 }
3399 {\endlatexitimage}
3400
3401 }

```

```
Env  multiline*
```

```
3402 \BeforeBeginEnvironment{multiline*}{  
3403 \ifbool{mathjax}{  
3404 {  
3405 \LWR@syncmathjax  
3406 \booltrue{\LWR@amsmultiline}  
3407 \LWR@beginhideamsmath  
3408 }  
3409 {  
3410 \latexitimage[LWRAMSMATHBODY]  
3411 }  
3412 }  
3413  
3414 \AfterEndEnvironment{multiline*}{  
3415  
3416 \ifbool{mathjax}{  
3417 {  
3418 \LWR@endhideamsmath  
3419 \boolfalse{\LWR@amsmultiline}  
3420 \LWR@addmathjax{multiline*}{\the\@envbody}  
3421 }  
3422 {\endlatexitimage}  
3423  
3424 }  
3425
```

```
Env  gather
```

```
3426 \BeforeBeginEnvironment{gather}{  
3427 \ifbool{mathjax}{  
3428 {  
3429 \LWR@syncmathjax  
3430 \boolfalse{\LWR@amsmultiline}  
3431 \LWR@beginhideamsmath  
3432 }  
3433 {  
3434 \latexitimage[LWRAMSMATHBODY]  
3435 }  
3436 }  
3437  
3438 \AfterEndEnvironment{gather}{  
3439  
3440 \ifbool{mathjax}{  
3441 {  
3442 \LWR@endhideamsmath  
3443 \LWR@addmathjax{gather}{\the\@envbody}  
3444 }  
3445 {\endlatexitimage}
```

```
3446
3447 }

Env  gather*
3448 \BeforeBeginEnvironment{gather*}{

3449 \ifbool{mathjax}
3450 {
3451 \LWR@syncmathjax
3452 \boolfalse{\LWR@amsmultline}
3453 \LWR@beginhideamsmath
3454 }
3455 {
3456 \lateximage[LWRAMSMATHBODY]
3457 }
3458 }
3459
3460 \AfterEndEnvironment{gather*}{

3461
3462 \ifbool{mathjax}
3463 {
3464 \LWR@endhideamsmath
3465 \LWR@addmathjax{gather*}{\the\@envbody}
3466 }
3467 {\endlatexitimage}
3468
3469 }
```

```
Env  align
```

```
3470 \BeforeBeginEnvironment{align}{

3471 \ifbool{mathjax}
3472 {
3473 \LWR@syncmathjax
3474 \boolfalse{\LWR@amsmultline}
3475 \LWR@beginhideamsmath
3476 }
3477 {
3478 \latexitimage[LWRAMSMATHBODY]
3479 }
3480
3481
3482 \AfterEndEnvironment{align}{

3483
3484 \ifbool{mathjax}
3485 {
3486 \LWR@endhideamsmath
3487 \LWR@addmathjax{align}{\the\@envbody}
3488 }
```

```
3489 {\endlateximage}
3490
3491 }

Env  align*
3492 \BeforeBeginEnvironment{align*}{
3493 \ifbool{mathjax}
3494 {
3495 \LWR@syncmathjax
3496 \boolfalse{\LWR@amsmultline}
3497 \LWR@beginhideamsmath
3498 }
3499 {
3500 \latexitimage[LWRAMSMATHBODY]
3501 }
3502 }
3503
3504 \AfterEndEnvironment{align*}{
3505
3506 \ifbool{mathjax}
3507 {
3508 \LWR@endhideamsmath
3509 \LWR@addmathjax{align*}{\the\@envbody}
3510 }
3511 {\endlateximage}
3512
3513 }

Env  flalign
3514 \BeforeBeginEnvironment{flalign}{
3515 \ifbool{mathjax}
3516 {
3517 \LWR@syncmathjax
3518 \boolfalse{\LWR@amsmultline}
3519 \LWR@beginhideamsmath
3520 }
3521 {
3522 \latexitimage[LWRAMSMATHBODY]
3523 }
3524 }
3525
3526 \AfterEndEnvironment{flalign}{
3527
3528 \ifbool{mathjax}
3529 {
3530 \LWR@endhideamsmath
3531 \LWR@addmathjax{flalign}{\the\@envbody}
```

```

3532 }
3533 {\end{lateximage}
3534
3535 }

Env flalign*
3536 \BeforeBeginEnvironment{flalign*}{

3537 \ifbool{mathjax}
3538 {
3539 \LWR@syncmathjax
3540 \boolfalse{\LWR@amsmultline}
3541 \LWR@beginhideamsmath
3542 }
3543 {
3544 \latexitimage[LWRAMSMATHBODY]
3545 }
3546 }
3547
3548 \AfterEndEnvironment{flalign*}{

3549 \ifbool{mathjax}
3550 {
3551 \LWR@endhideamsmath
3552 \LWR@addmathjax{flalign*}{\the\@envbody}
3553
3554 }
3555 {\end{lateximage}
3556
3557 }

3558 \end{warpHTML}

```

59 Lateximages

A `\latexitimage` 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 `lwarfpmk` to extract the `\latexitimage` 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 L^AT_EX 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: 3559 `\begin{warpHTML}`

Ctr `LWR@lateximagenumber` Sequence the images.

```
3560 \newcounter{LWR@lateximagenumber}
3561 \setcounter{LWR@lateximagenumber}{0}
```

Ctr `LWR@lateximagedepth` Do not create `\latexitimage` inside of `\latexitimage`.

```
3562 \newcounter{LWR@lateximagedepth}
3563 \setcounter{LWR@lateximagedepth}{0}
```

Declare the `\LWR@file` for writing to generate file `lateximages.txt`:

```
3564 \ifcsdef{LWR@file}{}{\newwrite{\LWR@file}}
```

A few utility macros to write special characters:

```
3565 \edef\LWR@hashmark{\string#} % for use in \write
3566 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctr `LWR@LIpage` Used to reference the PDF page number of a `\latexitimage` to be written into `lateximages.txt`.

```
3567 \newcounter{LWR@LIpage}
```

```
3568 \end{warpHTML}
```

for HTML & PRINT: 3569 `\begin{warpall}`

`\LateximageFontSizeName` Declares how large to write text in the `\latexitimage`. The `.svg` file text size should blend well with the surrounding HTML text size.

Do not include the leading backslash in the name.

```
3570 \newcommand*{\LateximageFontSizeName}{large}
```

```
3571 \end{warpall}
```

for HTML output: 3572 `\begin{warpHTML}`

\LWR@HTMLsanitize {*text*}

Math expressions are converted to `lateximages`, and some math environments may contain “&”, “<”, or “>”, which should not be allowed inside an HTML ALT tag, so must convert them to HTML entities.

Two versions follow, depending on expansion needs. There may be a better way...

```

3573 \newcommand{\LWR@HTMLsanitize}[1]{%
3574 \protect\StrSubstitute{\detokenize{#1}}{%
3575 {\detokenize{&}}}{%
3576 {\detokenize{&#038;}}[\LWR@strresult]}{%
3577 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3578 {\detokenize{<}}}{%
3579 {\detokenize{&lt;}}}{%
3580 [\LWR@strresult]}{%
3581 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3582 {\detokenize{>}}}{%
3583 {\detokenize{&gt;}}}{%
3584 [\LWR@strresult]}{%
3585 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3586 {\detokenize{##}}}{%
3587 {\#\#}}}{%
3588 [\LWR@strresult]}{%
3589 \LWR@strresult}%
3590 }

```

\LWR@HTMLsanitizeexpand {*text*}

This version expands the argument before sanitizing it.

```

3591 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
3592 \protect\StrSubstitute{\detokenize\expandafter{#1}}{%
3593 {\detokenize{&}}}{%
3594 {\detokenize{&#038;}}}{%
3595 [\LWR@strresult]}{%
3596 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3597 {\detokenize{<}}}{%
3598 {\detokenize{&lt;}}}{%
3599 [\LWR@strresult]}{%
3600 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3601 {\detokenize{>}}}{%
3602 {\detokenize{&gt;}}}{%
3603 [\LWR@strresult]}{%
3604 \LWR@strresult}%
3605 }

```

Env `lateximage [`*alntag*`]`

```
3606 \NewDocumentEnvironment{lateximage}{O{image}}{%
3607 \LWR@traceinfo{lateximage: starting on page \arabic{page}}%
3608 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}>0}{%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
3609 {%
3610 \addtocounter{\LWR@lateximagedepth}{1}%
3611 }%
```

Otherwise, this is the outer-most lateximage:

```
3612 { % start of outer-most lateximage
```

Starting a new lateximage:

```
3613 \addtocounter{\LWR@lateximagenumber}{1}%
3614 \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{\LWR@lateximagenumber}}%
```

While inside a lateximage, do not use mathjax:

```
3615 \boolfalse{mathjax}
```

Be sure that are doing a paragraph:

```
3616 \LWR@ensuredoingapar%
```

Next file:

```
3617 \addtocounter{\LWR@externalfilecnt}{1}%
3618 \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{\LWR@externalfilecnt}}%
```

Figure out what the next page number will be:

```
3619 \setcounter{LWR@LIPage}{\LWR@lateximage\the\LWR@lateximagenumber}%
3620 \LWR@traceinfo{lateximage: LWR@LIPage is \arabic{\LWR@LIPage}}%
```

Create an HTML span which will hold the comment which contains the pdftotext translation of the image's page, and also will hold the link to the .svg file:

```
3621 \LWR@htmlltag{span id="lateximage\arabic{\LWR@lateximagenumber}" %
3622 class="lateximagesource"} \LWR@newline
```

Write instructions to the `lateximages.txt` file:

```
3623 \immediate\write\LWR@file{| \the\LWR@LIPage | \the\LWR@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.

```
3624 \LWR@htmlopencomment%
3625 \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
3626 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
3627 \LWR@origminipage{6in}%
3628 \csuse{LWR@orig\LateximageFontSizeName}%

```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for `\hspace`, etc. inside a lateximage.

```
3629 \let\hspace\LWR@orighspace%
3630 \let\rule\LWR@origrule%
3631 \let\,\LWR@origcomma% disable HTML short unbreakable space
3632 \let{textit}\LWR@origtextit%
3633 \let{textbf}\LWR@origtextbf%
3634 \let{texttt}\LWR@origtexttt%
3635 \let{textsc}\LWR@origtextsc%
3636 \let{textsf}\LWR@origtextsf%
3637 \let{textrm}\LWR@origtextrm%
3638 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
3639 \let{textsuperscript}\LWR@origtextsuperscript%
3640 \let{textsubscript}\LWR@origtextsubscript%
3641 \let~\LWR@origtilde%
3642 \let\enskip\LWR@origenskip%
3643 \let\quad\LWR@origquad%
3644 \let\qquad\LWR@origqquad%
3645 \let\tabular\LWR@origtabular%
3646 \let\endtabular\LWR@origendtabular%
3647 \let\newline\LWR@orignewline%
3648 \LWR@origlabel{LWR@lateximage\arabic{LWR@lateximagenumber}}%
3649 }% end of outer-most lateximage
3650 }% end of \begin{lateximage}
3651 { % start of \end{lateximage}
3652 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}>}{1}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
3653 {%
3654 \addtocounter{LWR@lateximagedepth}{-1}%
3655 }%
```

if this is the outer-most lateximage:

```
3656 { % end of outer-most lateximage
```

Finish the lateximage minipage and start a new PDF page:

```
3657 \LWR@origendminipage%
3658 \LWR@orignewline%
3659 \LWR@origscriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by pdftotext:

```
3660 \LWR@htmclosecomment{} \LWR@orignewline%
3661 \LWR@traceinfo{[lateximage: The page after the image is \arabic{page}]}%
```

Create a link to the lateximage, allowing its natural height:

If the alt tag is given as “LWRAMSMATHBODY”, then use the text collected by the amsmath multiline, gather, or align environments.

```
3662 \ifthenelse{\equal{#1}{LWRAMSMATHBODY}}%
3663 { %
3664 \LWR@subinlineimage[%
3665 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}%
3666 ]%
3667 {[lateximage]}%
3668 {[lateximages\OSPathSymbol{}lateximage-\the\@externalfilecnt]}%
3669 {[svg]}%
3670 {}%
3671 }%
3672 { %
3673 \LWR@subinlineimage[#1]{[lateximage]}%
3674 {[lateximages\OSPathSymbol{}lateximage-\the\@externalfilecnt}{[svg]}{}%
3675 }%
3676 % \LWR@orignewline% Removed to prevent extra space.
```

Be sure that are doing a paragraph:

```
3677 \LWR@ensuredoingapar%
```

Close the HTML span which has the pdftotext comment and also the link to the .svg image:

```
3678 \LWR@htmntag{/span}%
3679 \ifbool{HTMLDebugComments}{%
3680 \LWR@htmlcomment{End of lateximage}}%
3681 }{}%
3682 % \LWR@orignewline% Removed to prevent extra space.
```

Undo one lateximage level:

```
3683 \addtocounter{LWR@lateximagedepth}{-1}%
3684 }% end of outer-most lateximage
3685 \LWR@traceinfo{lateximage: done}
3686 }%
3687 \end{warpHTML}
```

for PRINT output: 3688 \begin{warpprint}
3689 \newenvironment{lateximage}[1] [] {\minipage{\linewidth}}{\endminipage}
3690 \end{warpprint}

60 center, flushleft, flushright

for HTML output: 3691 \begin{warpHTML}

Env **center** Replace center functionality with CSS tags:

```
3692 \renewenvironment*{center}
3693 {
3694 \LWR@forcenewpage
3695 \BlockClass{center}
3696 }
3697 {\endBlockClass}
```

Env **flushright**

```
3698 \renewenvironment*{flushright}
3699 {
3700 \LWR@forcenewpage
3701 \BlockClass{flushright}
3702 }
3703 {\endBlockClass}
```

Env **flushleft**

```
3704 \renewenvironment*{flushleft}
3705 {
3706 \LWR@forcenewpage
3707 \BlockClass{flushleft}
3708 }
3709 {\endBlockClass}

3710 \end{warpHTML}
```

61 Siunitx

Pkg **siunitx**

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

for HTML output: 3711 `\begin{warpHTML}`

Options for siunitx:

```
3712 \PassOptionsToPackage{
3713 detect-mode=true,
3714 per-mode=symbol,% fraction is not seen by pdftotext
3715 text-celsius = {\HTMLentity{deg}C},
3716 text-degree = {\HTMLentity{deg}},
3717 }{siunitx}

3718 \end{warpHTML}
```

62 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.

⚠ \graphicxpath `\graphicxpath` 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.

for HTML output: 3719 `\begin{warpHTML}`

62.1 `\graphicspath`

```
\graphicspath {<path>}

3720 \newcommand*{\thisgraphicspath}{{}}
3721 \renewcommand*{\graphicspath}[1]{\renewcommand*{\thisgraphicspath}{#1}}

\DeclareGraphicsExtensions {<list>}

\DeclareGraphicsRule {} {} {} {}

3722 \renewcommand*{\DeclareGraphicsExtensions}[1]{}
3723 \renewcommand*{\DeclareGraphicsRule}[4]{}
```

62.2 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.

3724 `\renewcommand*{\unitspace}{}{}`

Used to store the user's selected dimensions and HTML class.

The class defaults to "inlineimage" unless changed by a `class=xyz` option.

```
3725 \newlength{\LWR@igwidth}
3726 \newlength{\LWR@igheight}
```

```

3727 \newcommand*\{\LWR@igwidthstyle\}{}%
3728 \newcommand*\{\LWR@igheightstyle\}{}%
3729 \newcommand*\{\LWR@igorigin\}{}%
3730 \newcommand*\{\LWR@igangle\}{}%
3731 \newcommand*\{\LWR@igxscale\}{1}%
3732 \newcommand*\{\LWR@igyscale\}{1}%
3733 \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:

```

3734 \define@key{igraph}{width}{%
3735 \setlength{\LWR@igwidth}{#1}%
3736 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}{%
3737 {%

```

Default to use the converted fixed length given:

```

3738 \uselengthunit{PT}%
3739 \renewcommand*\{\LWR@igwidthstyle\}{width:\rndprintlength{\LWR@igwidth}}%

```

If ex or em dimensions were given, use those instead:

```

3740 \IfEndWith{#1}{ex}{%
3741 {\renewcommand*\{\LWR@igwidthstyle\}{width:#1}}% yes ex
3742 {}% not ex
3743 \IfEndWith{#1}{em}{%
3744 {\renewcommand*\{\LWR@igwidthstyle\}{width:#1}}% yes em
3745 {}% not em
3746 \IfEndWith{#1}{\%}{%
3747 {\renewcommand*\{\LWR@igwidthstyle\}{width:#1}}% yes percent
3748 {}% not percent
3749 \IfEndWith{#1}{px}{%
3750 {\renewcommand*\{\LWR@igwidthstyle\}{width:#1}}% yes px
3751 {}% not px
3752 }{}% end of length > 0pt
3753 }

```

If an optional height was given, set an HTML style:

```

3754 \define@key{igraph}{height}{%
3755 \setlength{\LWR@igheight}{#1}%
3756 \ifthenelse{\lengthtest{\LWR@igheight > 0pt}}{%
3757 {%

```

Default to use the converted fixed length given:

```

3758 \uselengthunit{PT}%
3759 \renewcommand*{\LWR@igheightstyle}{%
3760 height:\rndprintlength{\LWR@igheight} %
3761 }%

```

If ex or em dimensions were given, use those instead:

```

3762 \IfEndWith{#1}{ex}%
3763 {\renewcommand*{\LWR@igheightstyle}{height:#1}%
3764 {}%
3765 \IfEndWith{#1}{em}%
3766 {\renewcommand*{\LWR@igheightstyle}{height:#1}%
3767 {}%
3768 \IfEndWith{#1}{\%}%
3769 {\renewcommand*{\LWR@igheightstyle}{height:#1}%
3770 {}%
3771 \IfEndWith{#1}{px}%
3772 {\renewcommand*{\LWR@igheightstyle}{height:#1}%
3773 {}%
3774 }{}%
3775 }%

```

Handle origin key:

```

3776 \define@key{igraph}{origin}{%
3777 \renewcommand*{\LWR@igorigin}{#1}%
3778 }%

```

Handle angle key:

```

3779 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}

```

Handle class key:

```

3780 \define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}%
3781 }%

```

It appears that `graphicx` does not have separate keys for `xscale` and `yscale`. `scale` adjusts both at the same time.

```

3782 \define@key{igraph}{scale}{%
3783 \renewcommand*{\LWR@igxscale}{#1}%
3784 \renewcommand*{\LWR@igyscale}{#1}%

```

Numerous ignored keys:

```

3785 \define@key{igraph}{bb}{}
3786 \define@key{igraph}{bbllx}{}
3787 \define@key{igraph}{bblly}{}

```

```

3788 \define@key{igraph}{bburx}{}
3789 \define@key{igraph}{bbury}{}
3790 \define@key{igraph}{natwidth}{}
3791 \define@key{igraph}{natheight}{}
3792 \define@key{igraph}{hiresbb}{}
3793 \define@key{igraph}{viewport}{}
3794 \define@key{igraph}{trim}{}
3795 \define@key{igraph}{totalheight}{}
3796 \define@key{igraph}{keepaspectratio}{}
3797 \define@key{igraph}{clip}{}
3798 \define@key{igraph}{draft}{}
3799 \define@key{igraph}{type}{}
3800 \define@key{igraph}{ext}{}
3801 \define@key{igraph}{read}{}
3802 \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.

```

3803 \newcommand*{\LWR@rotstyle}[2]{%
3804   #1transform:rotate(-#2deg);%
3805 }

```

\LWR@scalestyle {*prefix*} {*xscale*} {*yscale*}

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.

```

3806 \newcommand*{\LWR@scalestyle}[3]{%
3807   #1transform:scale(#2,#3);%
3808 }

```

62.3 \includegraphics

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```

3809 \newbool{LWR@infloatrow}
3810 \boolfalse{LWR@infloatrow}

3811 \newcommand*{\LWR@imageextension}{}%
3812 \newcommand*{\LWR@expgraphicsfilename}{}%

```

```
\LWR@includegraphicsb * [<2: options>] [<3: options>] {<4: filename>}
  graphics syntax is \includegraphics * [<llx, lly>] [<urx, ury>] {<file>}
  graphicx syntax is \includegraphics [<key values>] {<file>}
  If #3 is empty, only one optional argument was given, thus graphicx syntax.
```

```
3813 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
3814 {%
```

Start the image tag on a new line, allow PDF output word wrap:

```
3815 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute \linewidth, \textwidth, \textheight arguments with a 6x9 inch size until the next \endgroup.

```
3816 \ifthenelse{\cnttest{\value{\LWR@minipagedepth}}{=}{0}}{%
3817 \ifbool{\LWR@infloatrow}{%
3818 {}}
3819 {%
3820 \setlength{\linewidth}{6in}%
3821 \setlength{\textwidth}{6in}%
3822 \setlength{\textheight}{9in}%
3823 }%
3824 }{}}
```

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.

```
3825 \edef{\LWR@expgraphicsfilename}{#4}
3826 \renewcommand*{\LWR@imageextension}{}%
3827 \IfFileExists{\detokenize\expandafter\thisgraphicspath\expgraphicsfilename.jpg}{%
3828 \renewcommand*{\LWR@imageextension}{.jpg}{}%
3829 \IfFileExists{\detokenize\expandafter\thisgraphicspath\expgraphicsfilename.JPG}{%
3830 \renewcommand*{\LWR@imageextension}{.JPG}{}%
3831 \IfFileExists{\detokenize\expandafter\thisgraphicspath\expgraphicsfilename.png}{%
3832 \renewcommand*{\LWR@imageextension}{.png}{}%
3833 \IfFileExists{\detokenize\expandafter\thisgraphicspath\expgraphicsfilename.PNG}{%
3834 \renewcommand*{\LWR@imageextension}{.PNG}{}%
3835 \IfFileExists{\detokenize\expandafter\thisgraphicspath\expgraphicsfilename.gif}{%
3836 \renewcommand*{\LWR@imageextension}{.gif}{}%
3837 \IfFileExists{\detokenize\expandafter\thisgraphicspath\expgraphicsfilename.GIF}{%
3838 \renewcommand*{\LWR@imageextension}{.GIF}{}%
3839 \IfFileExists{\detokenize\expandafter\thisgraphicspath\expgraphicsfilename.svg}{%
```

```

3840 {\renewcommand*{\LWR@imageextension}{.svg}}{}%
3841 \IfFileExists{\detokenize\expandafter>thisgraphicspath\LWR@expgraphicsfilename.SVG}%
3842 {\renewcommand*{\LWR@imageextension}{.SVG}}{}%
3843 \IfFileExists{\detokenize\expandafter>thisgraphicspath\LWR@expgraphicsfilename.svgz}%
3844 {\renewcommand*{\LWR@imageextension}{.svgz}}{}%
3845 \IfFileExists{\detokenize\expandafter>thisgraphicspath\LWR@expgraphicsfilename.SVGZ}%
3846 {\renewcommand*{\LWR@imageextension}{.SVGZ}}{}%

```

Convert a PDF extension to SVG, leave the result in `\LWR@strresult`:

Must also `\detokenize .pdf` and `.svg` comparison strings.

```

3847 \StrSubstitute{\detokenize\expandafter{\LWR@expgraphicsfilename}}{%
3848 {\detokenize{.pdf}}{\detokenize{.svg}}[\LWR@strresult]}%
3849 %
3850 \StrSubstitute{\LWR@strresult}{%
3851 {\detokenize{.PDF}}{\detokenize{.SVG}}[\LWR@strresult]}%

```

For correct em sizing during the width and height conversions:

```
3852 \large%
```

Reset some defaults, possibly will be changed below if options were given:

```

3853 \setlength{\LWR@igwidth}{Opt}%
3854 \setlength{\LWR@igheight}{Opt}%
3855 \renewcommand*{\LWR@igwidthstyle}{()}%
3856 \renewcommand*{\LWR@igheightstyle}{()}%
3857 \renewcommand*{\LWR@igorigin}{()}%
3858 \renewcommand*{\LWR@igangle}{()}%
3859 \renewcommand*{\LWR@igxscale}{1}%
3860 \renewcommand*{\LWR@igyscale}{1}%
3861 \renewcommand*{\LWR@igclass}{inlineimage}%

```

If #3 is empty, only one optional argument was given, thus `graphicx` syntax:

```

3862 \IfValueTF{#3}{}{%
3863 \IfValueTF{#2}{%
3864 {\setkeys{igraph}{#2}}%
3865 {\setkeys{igraph}{}}%
3866 }%

```

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.

```
3867 \href{\thisgraphicspath\LWR@strresult\LWR@imageextension}{}%
```

```

3868 {%
3869   % start of href
3870   \LWR@htmltag{%
3871     % start of image tags
3872     img src="\thisgraphicspath\LWR@strresult\LWR@imageextension" \LWR@orignewline
3873     \LWR@origtilde{} alt="\LWR@strresult" \LWR@orignewline

```

Only include a style tag if a width, height, angle, or scale was given:

```

3872 \ifthenelse{%
3873   \NOT\equal{\LWR@igwidthstyle}{} \OR
3874   \NOT\equal{\LWR@igheightstyle}{} \OR
3875   \NOT\equal{\LWR@igorigin}{} \OR
3876   \NOT\equal{\LWR@igangle}{} \OR
3877   \NOT\equal{\LWR@igxscale}{1} \OR
3878   \NOT\equal{\LWR@igyscale}{1}
3879 }%
3880 {\LWR@origtilde{} style="%"
3881 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{} }{%
3882   \LWR@igwidthstyle; }{%
3883 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{} }{%
3884   \LWR@igheightstyle; }{%
3885 \ifthenelse{\NOT\equal{\LWR@igorigin}{} }{%
3886   \LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline }{%
3887 \ifthenelse{\NOT\equal{\LWR@igangle}{} }{%
3888   %
3889   \LWR@rotstyle{-ms-}{\LWR@igangle}%
3890   \LWR@rotstyle{-webkit-}{\LWR@igangle}%
3891   \LWR@rotstyle{}{\LWR@igangle}%
3892 }{%
3893 \ifthenelse{\NOT\equal{\LWR@igxscale}{1} \OR%
3894 \NOT\equal{\LWR@igyscale}{1} }{%
3895   \LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}%
3896   \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
3897   \LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyscale} }{%
3898 " \LWR@orignewline }{%

```

Set the class:

```

3899 \LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%
3900 }% end of image tags
3901 }% end of href
3902 \endgroup

```

Return to small-sized output:

```

3903 \LWR@origscriptsize
3904 }

```

\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.

```
3905 \renewcommand*{\includegraphics}{%
3906 {%
```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```
3907 \LWR@ensuredoingapar%
3908 \begingroup%
3909 \LWR@includegraphicsb%
3910 }%
3911 \end{warpHTML}
```

for PRINT output: For print output, accept and then discard the new `class` key:

```
3912 \begin{warpprint}
3913 \define@key{Gin}{class}{}%
3914 \end{warpprint}
```

62.4 \rotatebox, \scalebox, \reflectbox

for HTML output: 3915 \begin{warpHTML}

\LWR@rotboxorigin Holds the origin key letters.

```
3916 \newcommand*{\LWR@rotboxorigin}{}%
```

```
\LWR@originname {\langle letter\rangle}
```

Given one L^AT_EX origin key value, translate into an HTML origin word:

```
3917 \newcommand*{\LWR@originname}[1]{%
3918 \ifthenelse{\equal{\#1}{t}}{top}{}
3919 \ifthenelse{\equal{\#1}{b}}{bottom}{}
3920 \ifthenelse{\equal{\#1}{c}}{center}{}
3921 \ifthenelse{\equal{\#1}{l}}{left}{}
3922 \ifthenelse{\equal{\#1}{r}}{right}{}
3923 }
```

```
\LWR@originnames {⟨letters⟩}
```

Given one- or two-letter L^AT_EX origin key values, translate into HTML origin words:

```
3924 \newcommand*{\LWR@originnames}[1]{%
3925 \StrChar{#1}{1}[\LWR@strresult]%
3926 \LWR@originname{\LWR@strresult}%
3927 \StrChar{#1}{2}[\LWR@strresult]%
3928 \LWR@originname{\LWR@strresult}%
3929 }
```

Handle the origin key for \rotatebox:

```
3930 \define@key{krotbox}{origin}{%
3931 \renewcommand*{\LWR@rotboxorigin}{#1}%
3932 }
```

These keys are ignored:

```
3933 \define@key{krotbox}{x}{}
3934 \define@key{krotbox}{y}{}
3935 \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.

```
3936 \NewDocumentCommand{\LWR@rotatebox}{O{} m +m}{%
```

Reset the origin to “none-given”:

```
3937 \renewcommand*{\LWR@rotboxorigin}{}
```

Process the optional keys, which may set \LWR@rotateboxorigin:

```
3938 \setkeys{krotbox}{#1} %
```

Select `inline-block` so that HTML will transform this span:

```
3939 \LWR@htmlltagc{span style="display: inline-block; %
```

If an origin was given, translate and print the origin information:

```
3940 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}{%
3941 {transform-origin: \LWR@originnames{\LWR@rotboxorigin}; \LWR@origtilde}{}%
```

Print the rotation information:

```
3942 \LWR@rotstyle{-ms-}{#2} %
3943 \LWR@rotstyle{-webkit-}{#2} %
3944 \LWR@rotstyle{}{#2} %
3945 "{}"\LWR@orignewline%
```

Print the text to be rotated:

```
3946 \begin{LWR@nestspan}%
3947 #3%
```

Close the span:

```
3948 \LWR@htmntagc{/span}%
3949 \end{LWR@nestspan}%
3950 }
```

`\scalebox {<h-scale>} [<v-scale>] {<text>}`

Will `\let\scalebox\LWR@scalebox` at `\LWR@LwarpStart`, in case `\scalebox` was over-written by a later package load.

```
3951 \NewDocumentCommand{\LWR@scalebox}{m o m}{}%
```

Select `inline-block` so that HTML will transform this span:

```
3952 \LWR@htmntagc{span style="display: inline-block; %
```

Print the scaling information:

```
3953 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3954 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3955 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3956 "{}%"
```

Print the text to be scaled:

```
3957 \begin{LWR@nestspan}%
3958 #3%
```

Close the span:

```
3959 \LWR@htmntagc{/span}%
3960 \end{LWR@nestspan}%
3961 }
```

`\reflectbox {<text>}`

Will \let\reflectbox\LWR@reflectbox at \LWR@LwarpStart, in case \reflectbox was over-written by a later package load.

```
3962 \newcommand{\LWR@reflectbox}[1]{\LWR@scalebox{-1}[1]{#1}}
```

```
3963 \end{warpHTML}
```

62.5 Null functions

These functions are not supported by l warp's HTML conversion.

for HTML output: 3964 \begin{warpHTML}

```
\resizebox {<h-length>} {<v-length>} {<text>}
```

Simply prints its text argument.

```
3965 \renewcommand{\resizebox}[3]{#3}
```

```
3966 \end{warpHTML}
```

63 Cleverref

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 l warp 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 l warp's additional loading of cleveref will have no effect.

Table 8 on 208 shows the data structure of the label/reference system as revised by l warp and cleveref.

A few patches allow cleveref to work as-is:

for HTML output: 3967 \begin{warpHTML}

\AtEndPreamble forces cleveref to be loaded last:

The following patches are applied after cleveref has loaded, and after \AtBeginDocument:

```

3968 \AfterEndPreamble{

\@setref  {\langle kindofref\rangle} {\langle label\rangle}

3969 \renewcommand*\@setref[2]{#1{\ref{#2}}{}{}}

\@setrefrange {\langle text\rangle} {\langle label\rangle} {\langle label\rangle}

3970 \renewcommand*\@setrefrange[3]{%
3971 #1{\ref{#2}}{\ref{#3}}{}{}{}}

```

\cpagerefFor Redefinable word between “page(s)” and the page numbers.

```

3972 \newcommand*\cpagerefFor[for]

\@setcpageref {\langle typeofref\rangle} {\langle label\rangle}, where typeofref is “page” or “pages”

3973 \renewcommand*\@setcpageref[2]{%
3974 #1{\cpagerefFor\ \cref{#2}}{}{}%
3975 }

3976 \renewcommand*\@setcpagerefrange[3]{%
3977 #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}{}{}}
3978 }% AfterEndPreamble

```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

```

3979 \let\LWR@origlabel\label
3980 \let\label\LWR@newlabel
3981 \let\LWR@origref\ref
3982 \let\ref\LWR@newref% \end{ syntax highlighting
3983 \let\LWR@origpageref\pageref
3984 \let\pageref\LWR@newpageref
3985
3986
3987
3988 \end{warpHTML}

```

64 Picture

Env picture The `picture` environment is enclosed inside a `\lateimage`.

for HTML output: 3989 `\begin{warpHTML}`

```

Env  picture

3990 \BeforeBeginEnvironment{picture}{%
3991 \latexitimage%
3992 \let\makebox\LWR@origmakebox%
3993 }%
3994
3995 \AfterEndEnvironment{picture}{\endlatexitimage}

3996 \end{warpHTML}

```

65 Boxes and Minipages

A CSS flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

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

⚠ 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`.

side-by-side Side-by-side minipages may be separated by `\quad`, `\quad`, `\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 are ignored. 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` tag, 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: 3997 `\begin{warpHTML}`

65.1 Counters and lengths

Ctr `\LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
3998 \newcounter{\LWR@minipagedepth}
3999 \setcounter{\LWR@minipagedepth}{0}
```

Len `\LWR@minipagewidth` Used to convert the width into printable units.

```
4000 \newlength{\LWR@minipagewidth}
```

Len `\LWR@minipageheight` Used to convert the height into printable units.

```
4001 \newlength{\LWR@minipageheight}
```

Remember the original definitions:

```
4002 \let\LWR@origminipage\minipage
4003 \let\LWR@origendminipage\endminipage
```

65.2 Footnote handling

Also see section 39 for other forms of footnotes.

65.3 Minipage handling

`\LWR@endminipage` Used to close a minipage.

Copied the L^AT_EX definition and modified to create a `mpfootnotes` div class:

```
4004 \def\LWR@endminipage{%
4005   \par
4006   \unskip
4007   \ifvoid\@mpfootins\else
4008     \vskip\skip\@mpfootins
4009     \normalcolor
4010   \LWR@htmldivclass{mpfootnotes}
4011   \LWR@origmedskip
4012     \unvbox\@mpfootins
4013   \LWR@htmldivclassend{mpfootnotes}
4014   \fi
4015   \minipagefalse
4016   \color@endgroup
4017   \egroup
4018   \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.

```
4019 \newcommand*\{\LWR@subminipage\}{%
4020 \LWR@stoppars
4021 \LWR@origminipage{6in}
```

\raggedright cancels hyphenation, which will be done by HTML instead.

```
4022 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
4023 \LWR@startpars%
4024 }
```

\LWR@endsubminipage Closes the subminipage.

```
4025 \newcommand*\{\LWR@endsubminipage\}{%
4026 \LWR@stoppars%
4027 \LWR@endminipage% The following empty line is required:
4028
4029 }
```

Bool LWR@minipagefullwidth Should the next minipage have no HTML width?

```
4030 \newbool{\LWR@minipagefullwidth}
4031 \boolfalse{\LWR@minipagefullwidth}
```

\minipagefullwidth Requests that the next minipage have no width tag in HTML:

for HTML output: 4032 \newcommand*\{\minipagefullwidth\}{\booltrue{\LWR@minipagefullwidth}}
 4033 \end{warpHTML}

for PRINT output: 4034 \begin{warpprint}
 4035 \newcommand*\{\minipagefullwidth\}{}
 4036 \end{warpprint}

for HTML output: 4037 \begin{warpHTML}

Bool LWR@minipagethispar Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
4038 \newbool{\LWR@minipagethispar}
4039 \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.

```
4040 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}
4041 {%
```

Pre-compute the given width and height:

Reset the text area if are starting the outer-most minipage:

```
4042 \LWR@traceinfo{starting minipage of width #4}%
4043 \uselengthunit{in}%
4044 \setlength{\LWR@minipagewidth}{#4}%
4045 \ifthenelse{\cnttest{\value{\LWR@minipagedepth}}=0}{%
4046 \addtolength{\LWR@minipagewidth}{3em}%
4047 \setlength{\linewidth}{6in}%
4048 \setlength{\textwidth}{6in}%
4049 \setlength{\textheight}{9in}%
4050 }{%
4051 \LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}%
4052 \addtocounter{\LWR@minipagedepth}{1}%
4053 \setlength{\LWR@minipageheight}{\textheight}%
4054 \IfValueTF{#2}{\setlength{\LWR@minipageheight}{#2}}{}}
```

L^AT_EX 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.

```
4055 \LWR@stopars%
```

Create the `<div>` tag with optional alignment style:

```
4056 \LWR@traceinfo[minipage: creating div class]%
4057 \LWR@orignewpage%
4058 \LWR@htmlltag{div class="minipage" style="%"
4059 \ifthenelse{\equal{#1}{t}}{vertical-align: bottom ; }{}%
4060 \ifthenelse{\equal{#1}{c}}{vertical-align: middle ; }{}%
4061 \ifthenelse{\equal{#1}{b}}{vertical-align: top ; }{}%
4062 \ifthenelse{\equal{#3}{t}}{justify-content: flex-start ; }{}%
4063 \ifthenelse{\equal{#3}{c}}{justify-content: center ; }{}%
4064 \ifthenelse{\equal{#3}{b}}{justify-content: flex-end ; }{}%
4065 \ifthenelse{\equal{#3}{s}}{justify-content: space-between ; }{}}
```

Print the width and optional height styles:

```
4066 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
4067 \uselengthunit{PT}%
4068 \ifbool{\LWR@minipagefullwidth}{%
4069 {\boolfalse{\LWR@minipagefullwidth}}%
4070 {%
4071 \ifthenelse{\lengthtest{#4}=\linewidth}{%
4072 {}{%
4073 {\width:\rndprintlength{\LWR@minipagewidth} ; }%
4074 }%
4075 \LWR@traceinfo{minipage: about to print the height}}%
4076 \IfValueTF{#2}{height:\rndprintlength{\LWR@minipageheight} ; }{}%
4077 "{}"}%
```

Finish with an empty line to start L^AT_EX minipage processing on a new line. Use a large minipage area to avoid the unnecessary wrapping of tags.

```
4078
4079 \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.

```
4080 \setlength{\linewidth}{#4}%. the original width
4081 \setlength{\textwidth}{6in}%
4082 \setlength{\textheight}{9in}%
```

\raggedright cancels hyphenation, which will be done by HTML instead.

```
4083 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
4084 \LWR@startpars%
4085 \LWR@traceinfo{minipage: finished starting the minipage}%
4086 }
```

End the environment with L^AT_EX processing and closing tag:

```
4087 {%
4088 \LWR@stoppars%
4089 \LWR@endminipage%. The following empty line is required:
4090
4091 \LWR@htmldivclassend{minipage}%
4092 \vspace{1\baselineskip}%. required for subcaption
4093 \addtocounter{\LWR@minipagedepth}{-1}%
4094 }
```

```
4094 \LWR@startpars%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
4095 \global\booltrue{LWR@minipagethispar}%
4096 }
```

65.4 Parbox, makebox, framebox, fbox, raisebox

```
\parbox [⟨pos⟩] [⟨height⟩] [⟨inner-pos⟩] {⟨width⟩} {⟨text⟩}
```

A parbox uses the minipage code:

```
4097 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}
4098 {
4099 \LWR@traceinfo{parbox of width #4}%
4100 \begin{minipage}[#1][#2][#3]{#4}
4101 #5
4102 \end{minipage}
4103 }
```

```
\makebox [⟨width⟩] [⟨pos⟩] {⟨text⟩}
```

Width and position are ignored.

```
4104 \let\LWR@origmakebox\makebox
4105
4106 \RenewDocumentCommand{\makebox}{o o m}{%
4107 \mbox{#3}}
4108 }
```

```
\framebox [⟨width⟩] [⟨pos⟩] {⟨text⟩}
```

Width and position are ignored.

```
4109 \RenewDocumentCommand{\framebox}{o o m}{%
4110 \fbox{#3}
4111 }
```

```
\fbox {⟨text⟩}
```

```
4112 \let\LWR@origfbox\fbox
4113 %
4114 \renewcommand*\fbox[1]{%
```

```

4115 \InlineClass{framebox}{#1}%
4116 }

\raisebox {⟨raiselen⟩} [⟨height⟩] [⟨depth⟩] {⟨text⟩}

4117 \RenewDocumentCommand{\raisebox}{m o o m}{%
4118 #4%
4119 }

4120 \end{warpHTML}

```

66 Direct formatting

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

For high-level block and inline custom CSS classes, see section [33.7](#).

for HTML output: 4121 \begin{warpHTML}

```

\emph {⟨text⟩}

4122 \renewcommand{\emph}[1]{\LWR@htmlspan{em}{#1}}


\textmd {⟨text⟩}

4123 \renewcommand{\textmd}[1]{\LWR@htmlspan{textmd}{#1}}


\textbf {⟨text⟩}

4124 \renewcommand{\textbf}[1]{\LWR@htmlspan{b}{#1}}


\textrm {⟨text⟩}

4125 \renewcommand{\textrm}[1]{\InlineClass{textrm}{#1}}


\textsf {⟨text⟩}

4126 \renewcommand{\textsf}[1]{\InlineClass{textsf}{#1}}


\texttt {⟨text⟩}

4127 \renewcommand{\texttt}[1]{\LWR@htmlspan{kbd}{#1}}

```

```
\textup {⟨text⟩}
4128 \renewcommand{\textup}[1]{\InlineClass{textup}{#1}}


\textit {⟨text⟩}
4129 \renewcommand{\textit}[1]{\LWR@htmlspan{i}{#1}}


\textsc {⟨text⟩}
4130 \renewcommand{\textsc}[1]{\InlineClass{textsc}{#1}}


\textnormal {⟨text⟩}
4131 \renewcommand{\textnormal}[1]{\textmd{\textrm{\textup{#1}}}}{}
```

\mdseries

```
4132 \renewcommand*{\mdseries}{}{}
```

\bfseries

```
4133 \renewcommand*{\bfseries}{}{}
```

\rmfamily

```
4134 \renewcommand*{\rmfamily}{}{}
```

\sffamily

```
4135 \renewcommand*{\sffamily}{}{}
```

\ttfamily

```
4136 \renewcommand*{\ttfamily}{}{}
```

\upshape

```
4137 \renewcommand*{\upshape}{}{}
```

\itshape

```
4138 \renewcommand*{\itshape}{}{}
```

\scshape

4139 \renewcommand*{\scshape}{}{}

\scshape

4140 \renewcommand*{\normalfont}{}{}

\sp {\langle text\rangle}

For **siunitx**. Must work in math mode.

4141 \renewcommand{\sp}[1]{\text{\#1}}

\sb {\langle text\rangle}

For **siunitx**. Must work in math mode.

4142 \renewcommand{\sb}[1]{\text{\textsubscript{\#1}}}

\textsuperscript {\langle text\rangle}

4143 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}

\textsubscript {\langle text\rangle}

4144 \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.

4145 \AtBeginDocument{\let\up\textsuperscript}

\fup {\langle text\rangle} Prints superscript.

Supports **fmtcount** package.

This is \let at the beginning of the document in case some other package has changed the definition.

4146 \AtBeginDocument{\let\fup\textsuperscript}

```
\hfill
4147 \renewcommand*{\hfill}{\qquad}

\hrulefill
4148 \renewcommand*{\hrulefill}{\rule{1in}{1pt} }

\dotfill
4149 \renewcommand*{\dotfill}{\dots}

4150 \end{warpHTML}
```

67 Skips, spaces, font sizes

for HTML output: 4151 \begin{warpHTML}

```
\, must be redefined after \RequirePackage{printlen}

4152 \let\LWR@origcomma|,
4153 \let\LWR@origtilde~
4154 \let\LWR@origskip\enskip
4155 \let\LWR@origquad\quad
4156 \let\LWR@origqquad\qquad
4157 \let\LWR@orighspace\hspace
4158 \let\LWR@origrule\rule
4159 \let\LWR@origmedskip\medskip
```

Direct-formatting space commands become HTML entities:

```
4160 \renewcommand*{\,}{\textnormal{}} % HTML thin non-breakable space

4161 \renewcommand*{\-}{\textnormal{nbsp}}
4162 \renewcommand*{\textellipsis}{\textnormal{...}}
```

Direct-formatting font sizes are ignored:

```
4163 \let\LWR@orignormalsize\normalsize
4164 \let\LWR@origsmall\small
4165 \let\LWR@origfootnotesize\footnotesize
4166 \let\LWR@origscriptsize\scriptsize
4167 \let\LWR@origtiny\tiny
4168 \let\LWR@origlarge\large
```

```

4169 \let\LWR@origLarge\Large
4170 \let\LWR@origLARGE\LARGE
4171 \let\LWR@orighuge\huge
4172 \let\LWR@origHuge\Huge
4173 \renewcommand*\normalsize{}
4174 \renewcommand*\small{}
4175 \renewcommand*\footnotesize{}
4176 \renewcommand*\scriptsize{}
4177 \renewcommand*\tiny{}
4178 \renewcommand*\large{}
4179 \renewcommand*\Large{}
4180 \renewcommand*\LARGE{}
4181 \renewcommand*\huge{}
4182 \renewcommand*\Huge{}
4183
4184 \renewcommand*\onecolumn{}
4185
4186 \renewcommand{\twocolumn}[1][]{%
4187
4188 #1
4189
4190 }

```

\newline Uses HTML
 tag

```

4191 \newcommand*{\LWR@newlinebr}{\unskip\LWR@htmntag{br /}\LWR@orignewline}%
4192 \let\newline\LWR@newlinebr

```

\\" Redefined to \LWR@endofline or \LWR@tabularendofline.

\LWR@endofline * [<len>]

\\" is assigned to \LWR@endofline at \LWR@LwarpStart.

Inside tabular, \\" is temporarily changed to \LWR@tabularendofline.

```

4193 \let\LWR@origendofline\
4194 \NewDocumentCommand{\LWR@endofline}{s o}
4195 {%
4196 \newline%
4197 }

```

\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, l warp correctly suppresses the paragraph tags between the minipages, unless some other

\hspace 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. **l warp** tries to do this by remembering that a minipage has been seen, in which case paragraph tags are suppressed around **\hspace**, **\enskip**, **\quad**, and **\qquad** until the end of the paragraph, when the closing p tag is created.

When a minipage is seen, the boolean **LWR@minipagethispar** is set, telling the following horizontal whitespace commands to try to suppress their surrounding paragraph tags. **LWR@minipagethispar** is cleared at the next end of paragraph, when the HTML paragraph closing tag is generated.

Placed just before **\hspace**, **\quad**, or **\qquad**'s HTML output.

```
4198 \newcommand*{\LWR@minipagestartpars}{%
4199 \ifbool{LWR@minipagethispar}{%
4200 {%
4201 \LWR@startpars%
4202 }{}}%
4203 }
```

\LWR@minipagestopars Placed just after **\hspace**, **\quad**, or **\qquad**'s HTML output.

```
4204 \newcommand*{\LWR@minipagestopars}{%
4205 \ifbool{LWR@minipagethispar}{%
4206 {%
4207 \LWR@stopars%
4208 }{}}%
4209 }
```

\quad Handles special minipage & horizontal space interactions.

```
4210 \renewcommand*{\quad}{%
4211 \LWR@minipagestopars%
4212 \HTMLunicode{2001}%
4213 \LWR@minipagestartpars%
4214 }
```

\qquad Handles special minipage & horizontal space interactions.

```
4215 \renewcommand*{\qquad}{\quad\quad}
```

\enskip Handles special minipage & horizontal space interactions.

```
4216 \renewcommand*{\enskip}{%
4217 \LWR@minipagestopars%
4218 \HTMLunicode{2000}%
4219 }
```

```
4219 \LWR@minipagestartpars%
4220 }
```

Len \WR@tempwidth Used to compute span width, height, raise for \hspace and \rule:

```
Len \WR@tempheight 4221 \newlength{\LWR@tempwidth}
Len \WR@tempraise 4222 \newlength{\LWR@tempheight}
4223 \newlength{\LWR@tempraise}
```

\LWR@hspace * {*length*}

Handles special minipage & horizontal space interactions.

Prints a span of a given width. Ignores the optional star.

\hspace{\fill} is converted to \hspace{2em}, equal to \qquad.

```
4224 \NewDocumentCommand{\LWR@hspace}{s m}{%
4225 \setlength{\LWR@tempwidth}{#2}}%
```

If \fill, change to \qquad:

```
4226 \ifnum\gluestretchorder\LWR@tempwidth>0%
4227 \setlength{\LWR@tempwidth}{2em}%
4228 \fi%
```

Only if the width is not zero:

```
4229 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{0pt}}{}{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4230 \LWR@minipagestopars%
```

Support the HTML thin wrappable space:

```
4231 \ifthenelse{\dimtest{\LWR@tempwidth}{=.16667em}}%
4232 {}%
4233 \HTMLunicode{2009}% thin breakable space
4234 }%
```

Print the span with the converted width. Not rounded.

```
4235 {}%
4236 \uselengthunit{PT}%
4237 \LWR@htmllagc{%
4238 span style="width:\printlength{\LWR@tempwidth}; display:inline-block;"%}
```

```
4239 }%
4240 \LWR@htmltagc{/span}%
4241 }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4242 \LWR@minipagestartpars%
4243 }%
4244 }
```

\hspace * {*<length>*}

Handles special minipage & horizontal space interactions.

```
4245 \let\hspace\LWR@hspace
```

\linebreak [*num*] Inserts an HTML br tag.

```
4246 \renewcommand*{\linebreak}[1][]{\newline}
```

\nolinebreak [*num*]

```
4247 \renewcommand*{\nolinebreak}[1]{}{}
```

\pagebreak [*num*] Starts a new paragraph.

```
4248 \renewcommand*{\pagebreak}[1][]{%
4249 %
4250 }
```

\nopagebreak [*num*]

```
4251 \renewcommand*{\nopagebreak}[1]{}{}
```

\enlargethispage * {*<len>*}

```
4252 \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.

```
4253 \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:

```
4254 \setlength{\LWR@tempwidth}{#2}%
```

If it's zero-width then skip the entire rule:

```
4255 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}%
4256 {}% zero- width
4257 {}% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
4258 \ifthenelse{\lengthtest{\LWR@tempwidth>0pt}\AND%
4259 \lengthtest{\LWR@tempwidth<1pt}}%
4260 {\setlength{\LWR@tempwidth}{1pt}}{}%
```

Likewise with height:

```
4261 \setlength{\LWR@tempheight}{#3}%
4262 \ifthenelse{\lengthtest{\LWR@tempheight>0pt}\AND%
4263 \lengthtest{\LWR@tempheight<1pt}}%
4264 {\setlength{\LWR@tempheight}{1pt}}{}%
```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
4265 \LWR@minipagestopars%
```

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.

```
4266 \uselengthunit{PT}%
4267 \LWR@htmlltagc{%
4268 span
4269 style="" %
```

The background color is used to draw the filled rule. The color may be changed by \textcolor.

```
4270 background:\LWR@currenttextcolor; %
```

The width and height are printed, converted to PT:

```
4271 width:\printlength{\LWR@tempwidth}; %
4272 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.

```
4273 \IfValueTF{#1}%
4274 {%
4275 \setlength{\LWR@tempraise}{\Opt-#1}%
4276 \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
4277 \LWR@orignewline%
4278 -ms-transform: translate(\Opt,\printlength{\LWR@tempraise}); %
4279 \LWR@orignewline%
4280 -webkit-transform: translate(\Opt,\printlength{\LWR@tempraise}); %
4281 \LWR@orignewline%
4282 transform: translate(\Opt,\printlength{\LWR@tempraise}); %
4283 \LWR@orignewline%
4284 }{}}%
```

Display inline-block to place the span inline with the text:

```
4285 display:inline-block;"%
4286 }%
4287 \LWR@htmltagc{/span}%"
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4288 \LWR@minipagestartpars%
4289 }% non-zero width
4290 }
```

\rule [*raise*] {[*width*} {[*height*}]

Handles special minipage & horizontal space interactions.

```
4291 \let\rule\LWR@rule
4292 \end{warpHTML}
```

68 \phantomsection

for HTML output: 4293 \begin{warpHTML}

\phantomsection Emulate the hyperref \phantomsection command, often used to insert the bibliography into table of contents:

4294 \newcommand*\{\phantomsection\}{\section*{}}

4295 \end{warpHTML}

69 \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>

69.1 HTML logos

for HTML output: 4296 \begin{warpHTML}

\TeX{} \TeX{}

latexlogo is a CSS class used to properly typeset the E and A in L^AT_EX{} and friends.

latexlogofont is a CSS class used to select the font for the rest of the logo in L^AT_EX{}, LuaT_EX{}, ConT_EXt, etc.

4297 \newcommand*\{\LWR@TeX\}
 4298 {\InlineClass{latexlogofont}}%
 4299 {\InlineClass{latexlogo}{T\textsubscript{e}X}}}}

\LaTeX{} L^AT_EX{}, L^AT_EX{} 2_ε
 \LaTeXe
 4300 \newcommand*\{\LWR@LaTeX\}
 4301 {\InlineClass{latexlogofont}}%

```

4302 {\InlineClass{latexlogo}%
4303 {L\textsuperscript{a}T\textsubscript{e}X}%
4304 %
4305 \renewcommand*\{\LaTeXe}%
4306 {\LaTeX\InlineClass{latexlogofont}%
4307 {\textit{\HTMLunicode{3B5}}}}}

```

\LuaTeX \LuaTeX, \LuaLATEX

\LuaLaTeX

```

4308 \newcommand*\{\LWR@LuaTeX\}{\InlineClass{latexlogofont}{Lua}\TeX}%
4309 \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.

```

4310 \newcommand*\{\Xe}%
4311 {X\textsubscript{\HTMLunicode{18e}}}%
4312 \newcommand*\{\LWR@XeTeX\}{\InlineClass{xetexlogo}{\Xe}\TeX}%
4313 \newcommand*\{\LWR@XeLaTeX\}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}

```

\ConTeXt \ConTeXt

```

4314 \newcommand*\{\LWR@ConTeXt\}%
4315 {\InlineClass{latexlogofont}{Con}\TeX\%}
4316 \InlineClass{latexlogofont}{t}}

```

\BibTeX \BIBTeX, *MakeIndex*

\MakeIndex

```

4317 \providecommand*\{\BibTeX\}%
4318 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}%
4319 %
4320 \newcommand*\{\MakeIndex\}%
4321 {\InlineClass{latexlogofont}{\textit{MakeIndex}}}

```

\AmS \AMS

amslogo is a CSS class used for the \AMS logo.

```

4322 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}%
4323 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}

```

\MiKTeX MiKTeX

```
4324 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}
```

\LyX LyX

lyxlogo is a CSS class used for the LyX logo.

```
4325 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
```

```
4326 \end{warpHTML}
```

69.2 Print logos

for PRINT output:

```
4327 \begin{warpprint}
4328 \newcommand*{\XeTeXrevE}{%
4329   {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}}
4330 \providecommand*{\XeTeX}{\mbox{X}\XeTeXrevE\TeX}}
4331 \providecommand*{\XeLaTeX}{\mbox{X}\XeTeXrevE\LaTeX}
4332 \providecommand*{\AmS}{\%
4333 \leavevemode\hbox{$\mathcal A$\kern-.2em\lower.376ex\%}
4334 \hbox{$\mathcal M$\kern-.2em$\mathcal S$}}
4335 \newcommand*{\LyX}{\textsf{LyX}}
4336 \providecommand*{\LuaTeX}{\mbox{Lua}\TeX}
4337 \providecommand*{\LuaLaTeX}{\mbox{Lua}\LaTeX}
4338 \providecommand*{\BibTeX}{\mbox{B}\textit{TeX}}
4339 \providecommand*{\MakeIndex}{\mbox{\textit{Index}}}
4340 \providecommand*{\ConTeXt}{\mbox{Con}\TeX{}t}
4341 \providecommand*{\MiKTeX}{\mbox{MiK}\TeX}
4342 \end{warpprint}
```

70 \AtBeginDocument, \AtEndDocument

for HTML output:

```
4343 \begin{warpHTML}
```

\LWR@LwarpStart Automatically sets up the HTML-related actions for the start and end of the
 \LWR@LwarpEnd document.

```
4344 \AfterEndPreamble{\LWR@LwarpStart}
4345 \AtEndDocument{\LWR@LwarpEnd}
```

```
4346 \end{warpHTML}
```

71 lwarpmk.conf

File `lwarpmk.conf` `lwarpmk.conf` is automatically (re-)created by the `lwarf` 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
```

Defaults unless already over-ridden by the user:

```
4347 \providecommand*\{\HomeHTMLFileName\}{\BaseJobname}
4348 \providecommand*\{\HTMLfiveName\}{}
4349 \providecommand*\{\UseLatexmk\}{false}

for PRINT output: 4350 \begin{warpprint}
4351 \ifcsdef{LWR@file}{}{\newwrite{\LWR@file}}
4352 \immediate\openout\LWR@file=lwarpmk.conf
4353 \ifwindows
4354 \immediate\write{\LWR@file}{opsystem = "Windows"}
4355 \else
4356 \immediate\write{\LWR@file}{opsystem = "Unix"}
4357 \fi
4358 \ifPDFTeX
4359 \immediate\write{\LWR@file}{latexname = "pdflatex"}
4360 \fi
4361 \ifXeTeX
4362 \immediate\write{\LWR@file}{latexname = "xelatex"}
4363 \fi
4364 \ifLuaTeX
4365 \immediate\write{\LWR@file}{latexname = "lualatex"}
4366 \fi
4367 \immediate\write{\LWR@file}{sourcename = "\jobname"}
4368 \immediate\write{\LWR@file}{%
4369 homehtmlfilename = "\HomeHTMLFileName"%
4370 }
4371 \immediate\write{\LWR@file}{htmlfilename = "\HTMLfiveName"}
4372 \immediate\write{\LWR@file}{uselatexmk = "\UseLatexmk"}
4373 \immediate\write{\LWR@file}{language = "\LWR@lwarpmklang"}
4374 \immediate\closeout{\LWR@file}
4375 \end{warpprint}
```

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-newproject.sty

73 lwarp-newproject

Pkg **lwarp-newproject** *lwarp-newproject* is used to create *lwarp*-specific system files in a new project.

Opt **lwarp-newproject warpprint** Configuration files are only created if the *warpprint* option was used. When this is the case, *lwarp-newproject* knows that *\jobname* is the source code's filename (as opposed to the *<project>.html.tex* filename).

Opt **lwarp-newproject warpHTML** Does not create configuration files. *warpprint* or *warpHTML* are passed to both *lwarp* and *lwarp-newproject* by *lwarpmk*, depending on whether *lwarpmk print* or *lwarpmk html* was used.

Opt **lwarp-newproject BaseJobName** *BaseJobname=somename* sets *\BaseJobname* to *somename*, which is used to set *\HomeHTMLFileName*, which is then written to the *lwarpmk.conf* and *<project>.lwarpmkconf* configuration files.

\BaseJobname is also written to *<project>.html.tex* as an option given for the *lwarp* and *lwarp-newproject* packages during the creation of HTML output.

\BaseJobname is the *\jobname* of the printed version, even while compiling the HTML version, where *\jobname* has *_html* appended.

Opt **lwarp-newproject lwarpmk** Tells *lwarp-newproject* 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*.

Opt **lwarp-newproject lwarpmklang** Sets the language to be assigned in *lwarpmk*'s configuration files. This is then used by *lwarpmk* while processing the glossary.

In the document source:

```
\documentclass{article} % or book, report
...
(font selection, input encoding)
...
\newcommand*{\HomeHTMLFileName}{index}, or {projectname}
\newcommand*{\HTMLfiveName}{}%
\usepackage{lwarp-newproject}
\usepackage{lwarp}
...
(load other packages, the rest of the preamble)
```

```

...
\NewCSS{project.css}
\NewHTMLdescription{Summarize this webpage.}
...
\begin{document}
...
\end{document}
```

Place `\usepackage{lwarf-newproject}` just before `\usepackage{lwarf}` in the document source. The operating system and TeX engine will be auto-detected, and a customized `lwarfpmk.conf` file will be generated, along with the other support files (`*.css`, `lwerp_html.xdy`, `lwerp_mathjax.txt`). Rename `sample_project.css` to your own custom `projectname.css` and edit it if desired.

After the first compile, `\usepackage{lwarf-newproject}` may be commented out unless the configuration changes. For example, it may be re-enabled to switch between `pdflatex`, `xelatex`, and `lualatex`, in which case a new manual compile (without using `lwarfpmk`) will auto-configure and regenerate the files. It may also be left enabled at all times, in which case the configuration files will be regenerated each time. Note that your own `projectname.css` file will not be over-written by the newly regenerated `sample_project.css`.

Discard all options for `lwarf-newproject`:

```

1 \ProvidesPackage{lwarf-newproject}
2
3 \RequirePackage{etoolbox}
4 \RequirePackage{comment}
5 \RequirePackage{fancyvrb}
6 \RequirePackage{ifplatform}%
7 \RequirePackage{iftex}%
8 \RequirePackage{kvoptions}
9 \SetupKeyvalOptions{family=LWRNP,prefix=LWRNP@}
```

Generate config files?

```

10 \newbool{LWRNP@genconf}
11 \boolfalse{LWRNP@genconf}
```

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

```
12 \excludecomment{LWR@createlwarfpmk}
```

Process options:

```

13 \DeclareVoidOption{warpprint}{\booltrue{LWRNP@genconf}}
14 \DeclareVoidOption{warpHTML}{\boolfalse{LWRNP@genconf}}
```

```

15 \DeclareVoidOption{lwarpmk}{\includecomment{LWR@createlwarpmk}}
16 \DeclareStringOption[\jobname]{BaseJobname}
17 \DeclareStringOption[english]{lwarpmklang}
18
19 \booltrue{LWRNP@genconf}%
20 \warpprint
21 \ProcessKeyvalOptions*\relax

```

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

```
22 \providecommand*\BaseJobname{\LWRNP@BaseJobname}
```

Defaults unless already over-ridden by the user:

```

23 \% \ifbool{LWRNP@genconf}%
24 \providecommand*\HomeHTMLFileName{\BaseJobname}%
25 \% }{}%
26 \providecommand*\HTMLfiveName{}%
27 \providecommand*\UseLatexmk{false}

```

73.1 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.

```

28 \ifbool{LWRNP@genconf}%
29 \ifcsdef{LWR@file}{}{\newwrite{\LWR@file}}
30 \immediate\openout\LWR@file=\jobname_html.tex
31 \immediate\write\LWR@file{%
32 \detokenize{\PassOptionsToPackage}%
33 {warpHTML,BaseJobname=\jobname}\{l warp\}%
34 }
35 \immediate\write\LWR@file{%
36 \detokenize{\PassOptionsToPackage}%
37 {warpHTML,BaseJobname=\jobname}\{l warp-newproject\}%
38 }
39 \immediate\write\LWR@file{%
40 \detokenize{\input}\string{\jobname.tex\string }%
41 }
42 \immediate\closeout\LWR@file
43 }{}%

```

73.2 project.lwarpmkconf

File `project.lwarpmkconf` The configuration file for `lwarpmk`.

```

44 \ifbool{LWRNP@genconf}{
45 \ifcsdef{LWR@file}{}{\newwrite{\LWR@file}}
46 \immediate\openout\LWR@file=\jobname.lwarpmkconf
47 \ifwindows
48 \immediate\write{\LWR@file}{opsystem = "Windows"}
49 \else
50 \immediate\write{\LWR@file}{opsystem = "Unix"}
51 \fi
52 \ifPDFTeX
53 \immediate\write{\LWR@file}{latexname = "pdflatex"}
54 \fi
55 \ifXeTeX
56 \immediate\write{\LWR@file}{latexname = "xelatex"}
57 \fi
58 \ifLuaTeX
59 \immediate\write{\LWR@file}{latexname = "lualatex"}
60 \fi
61 \immediate\write{\LWR@file}{sourcename = "\jobname"}
62 \immediate\write{\LWR@file}{%
63 homehtmlfilename = "\HomeHTMLFileName"%
64 }
65 \immediate\write{\LWR@file}{htmlfilename = "\HTMLfiveName"}
66 \immediate\write{\LWR@file}{uselatexmk = "\UseLatexmk"}
67 \immediate\write{\LWR@file}{language = "\LWRNP@lwarpmklang"}
68 \immediate\closeout{\LWR@file}
69 }{}
```

73.3 lwarf.css

File `lwarf.css` This is the base CSS layer used by `lwarf`.

This must be present both when compiling the project and also when distributing the HTML files.

```

70 \begin{VerbatimOut}{lwarf.css}
71 /*
72   CSS stylesheet for the LaTeX lwarf package
73   Copyright 2016-2017 Brian Dunn -- BD Tech Concepts LLC
74 */
75
76
77 /* a fix for older browsers: */
78 header, section, footer, aside, nav, main,
79   article, figure { display: block; }
80
81
82 A:link {color:#000080 ; text-decoration: none ; }
```

```
83 A:visited {color:#800000 ; }
84 A:hover {color:#000080 ; text-decoration: underline ;}
85 A:active {color:#800000 ; }
86
87 a.tocpart {display: inline-block ; margin-left: 0em ;
88     font-weight: bold ;}
89 a.tocchapter {display: inline-block ; margin-left: 0em ;
90     font-weight: bold ;}
91 a.tocsection {display: inline-block ; margin-left: 1em ;
92     text-indent: -.5em ; font-weight: bold ;}
93 a.tocsubsection {display: inline-block ; margin-left: 2em ;
94     text-indent: -.5em ;}
95 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
96     text-indent: -.5em ;}
97 a.tocparagraph {display: inline-block ; margin-left: 4em ;
98     text-indent: -.5em ;}
99 a.toc subparagraph {display: inline-block ; margin-left: 5em ;
100    text-indent: -.5em ;}
101 a.tocfigure {margin-left: 0em}
102 a.tocsubfigure {margin-left: 2em}
103 a.toctable {margin-left: 0em}
104 a.tocsubtable {margin-left: 2em}
105 a.toctheorem {margin-left: 0em}
106 a.toclstlisting {margin-left: 0em}
107
108
109 body {
110     font-family: "DejaVu Serif", "Bitstream Vera Serif",
111         "Lucida Bright", Georgia, serif;
112     background: #FAF7F4 ;
113     color: black ;
114     margin:0em ;
115     padding:0em ;
116     font-size: 100% ;
117     line-height: 1.2 ;
118 }
119
120 p {margin: 1.5ex 0em 1.5ex 0em ;}
121
122 /* Holds a section number to add space between it and the name */
123 span.sectionnumber { margin-right: .6em }
124
125 /* Inserted in front of index lines */
126 span.indexitem {margin-left: 0em}
127 span.indexsubitem {margin-left: 2em}
128 span.indexsubsubitem {margin-left: 4em}
129
130 div.hidden { display: none ; }
131
132 kbd {
```

```
133     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
134         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
135         "Courier New", monospace;
136     font-size: 100% ;
137 }
138
139 span.strong { font-weight: bold; }
140
141 span.textmd { font-weight: normal; }
142
143 span.textsc { font-variant: small-caps; }
144
145 span.textup { font-variant: normal; }
146
147 span.textrm {
148     font-family: "DejaVu Serif", "Bitstream Vera Serif",
149     "Lucida Bright", Georgia, serif;
150 }
151
152 span.textsf {
153     font-family: "DejaVu Sans", "Bitstream Vera Sans",
154     Geneva, Verdana, sans-serif ;
155 }
156
157 span.attribution {
158     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
159 }
160
161 span.citetitle {
162     margin-left: 1em ; font-size: 80% ; font-style: oblique;
163 }
164
165 span.poemtitle {
166     font-size: 120% ; font-weight: bold;
167 }
168
169 blockquote {
170 margin-left: 0px ;
171 margin-right: 0px ;
172 }
173
174 blockquote p {
175     line-height: 1.5;
176     text-align: left ;
177     font-size: .85em ;
178     margin-left: 3em ;
179 margin-right: 3em ;
180 }
181
182 blockquotation {
```

```
183 margin-left: 0px ;
184 margin-right: 0px ;
185 }
186
187 blockquotation p {
188     line-height: 1.5;
189     text-align: left ;
190     font-size: .85em ;
191     margin-left: 3em ;
192 margin-right: 3em ;
193 }
194
195 div.epigraph {
196     line-height: 1.2;
197     text-align: left ;
198     padding: 3ex 1em 0ex 1em ;
199 /*      margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
200     margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
201 /*      margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
202     font-size: .85em ;
203     max-width: 27em ;
204 }
205
206
207
208 div.epigraphsource{
209     text-align:right ;
210     margin-left:auto ;
211 /*      max-width: 50% ; */
212     border-top: 1px solid #AOAOAO ;
213     padding-bottom: 3ex ;
214     line-height: 1.2;
215 }
216
217 div.epigraph p { padding: .5ex ; margin: 0ex ;}
218 div.epigraphsource p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
219
220
221 /* lettrine package: */
222 span.lettrine { font-size: 3ex ; float: left ; }
223 span.lettrinetext { font-variant: small-caps ; }
224
225 /* ulem and soul packages: */
226 span.uline {
227     text-decoration: underline ;
228     text-decoration-skip ;
229 }
230
231 span.uline {
232     text-decoration: underline ;
```

```
233     text-decoration-skip ;
234     text-decoration-style: double ;
235 }
236
237 span.uwave {
238     text-decoration: underline ;
239     text-decoration-skip ;
240     text-decoration-style: wavy ;
241 }
242
243 span.sout {
244     text-decoration: line-through ;
245 }
246
247 span.xout {
248     text-decoration: line-through ;
249 }
250
251 span.dashuline {
252     text-decoration: underline ;
253     text-decoration-skip ;
254     text-decoration-style: dashed ;
255 }
256
257 span.dotuline {
258     text-decoration: underline ;
259     text-decoration-skip ;
260     text-decoration-style: dotted ;
261 }
262
263 span.letterspacing { letter-spacing: .2ex ; }
264
265 span.capsspacing {
266     font-variant: small-caps ;
267     letter-spacing: .1ex ;
268 }
269
270 span.highlight { background: #F8E800 ; }
271
272
273
274
275 html body {
276     margin: 0 ;
277     line-height: 1.2;
278 }
279
280
281 body div {
282     margin: 0ex;
```

```
283 }
284
285
286 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
287 {
288     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
289     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
290     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
291     "Times New Roman", serif;
292     font-style: normal ;
293     font-weight: bold ;
294     text-align: left ;
295 }
296
297 h1 { /* title of the entire website, used on each page */
298     text-align: center ;
299     font-size: 2.5em ;
300     padding: .4ex 0em 0ex 0em ;
301 }
302 h2 { font-size: 2.25em }
303 h3 { font-size: 2em }
304 h4 { font-size: 1.75em }
305 h5 { font-size: 1.5em }
306 h6 { font-size: 1.25em }
307 span.paragraph {font-size: 1em ; font-variant: normal ;
308     margin-right: 1em ; }
309 span.subparagraph {font-size: 1em ; font-variant: normal ;
310     margin-right: 1em ; }
311
312
313
314 /* Title of the file */
315 h1 {
316     margin: 0ex 0em 0ex 0em ;
317     line-height: 1.3;
318     text-align: center ;
319 }
320
321 /* Part */
322 h2 {
323     margin: 1ex 0em 1ex 0em ;
324     line-height: 1.3;
325     text-align: center ;
326 }
327
328 /* Chapter */
329 h3 {
330     margin: 3ex 0em 1ex 0em ;
331     line-height: 1.3;
332 }
```

```
333
334 /* Section */
335 h4 {
336   margin: 3ex 0em 1ex 0em ;
337   line-height: 1.3;
338 }
339
340 /* Sub-Section */
341 h5 {
342   margin: 3ex 0em 1ex 0em ;
343   line-height: 1.3;
344 }
345
346 /* Sub-Sub-Section */
347 h6 {
348   margin: 3ex 0em 1ex 0em ;
349   line-height: 1.3;
350 }
351
352
353 div.titlepage {
354   text-align: center ;
355 }
356
357 .footnotes {
358   font-size: .85em ;
359   margin: 3ex 1em 0ex 1em ;
360   padding-bottom: 1ex ;
361 border-top: 1px solid silver ;
362 }
363
364 .marginpar {
365   max-width:50%;
366   float:right;
367   text-align:left;
368   margin: 1ex 0.5em 1ex 1em ;
369   padding: 1ex 0.5em 1ex 0.5em ;
370   font-size: 85% ;
371   border-top: 1px solid silver ;
372   border-bottom: 1px solid silver ;
373   overflow-x: auto;
374 }
375
376 .marginpar br { margin-bottom: 2ex ; }
377
378 div.marginblock {
379   max-width:50%;
380   float:right;
381   text-align:left;
382   margin: 1ex 0.5em 1ex 1em ;
```

```
383     padding: 1ex 0.5em 1ex 0.5em ;
384     overflow-x: auto;
385 }
386
387 div.marginblock div.minipage {
388     display: block ;
389     margin: 0pt auto 0pt auto ;
390 }
391
392 div.marginblock div.minipage p { font-size: 85%}
393
394 div.marginblock br { margin-bottom: 2ex ; }
395
396
397 section.textbody div.footnotes{
398     margin: 3ex 0em 0ex 0em ;
399     border-bottom: 2px solid silver ;
400 }
401
402 .footnoteheader {
403     border-top: 2px solid silver ;
404     margin-top: 3ex ;
405     padding-top: 1ex ;
406     font-weight: bold ;
407 }
408
409 .mpfootnotes {
410     text-align: left ;
411     font-size: .85em ;
412     margin-left: 1em ;
413     border-top: 1px solid silver ;
414 }
415
416 /* Remove footnote top border in the title page. */
417 div.titlepage div.mpfootnotes {
418     border-top: none ;
419 }
420
421
422
423 ol {
424     margin: 1ex 1em 1ex 0em;
425     line-height: 1.2;
426 }
427
428 ul, body dir, body menu {
429     margin: 1ex 1em 1ex 0em;
430     line-height: 1.2;
431 }
432
```

```
433 li { margin: 0ex 0em 1ex 0em; }
434
435 html {
436   margin: 0;
437   padding: 0;
438 }
439
440 .programlisting {
441   font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
442             "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
443             "Courier New", monospace;
444   margin: 1ex 0ex 1ex 0ex ;
445   padding: .5ex 0pt .5ex 0pt ;
446   overflow-x: auto;
447 }
448
449 section.textbody>pre.programlisting {
450 border-top: 1px solid silver ;
451 border-bottom: 1px solid silver ;
452 }
453
454
455 .inlineprogramlisting {
456   font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
457             "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
458             "Courier New", monospace;
459   overflow-x: auto;
460 }
461
462
463 div.abstract {
464   margin: 2em 5% 2em 5% ;
465   padding: 1ex 1em 1ex 1em ;
466 /*  font-weight: bold ; */
467   font-size: 90% ;
468 }
469
470 div.abstract dl {line-height:1.5;}
471 div.abstract dt {color:#304070;}
472
473 div.abstracttitle{
474   font-family: "URW Classico", Optima, "Linux Biolinum O",
475             "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
476             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
477   font-weight:bold;
478   font-size:1.25em;
479   text-align: center ;
480 }
481
482 span.abstractrunintitle{
```

```
483     font-family: "URW Classico", Optima, "Linux Biolinum 0",
484         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
485         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
486     font-weight:bold;
487 }
488
489
490 .verbatim {
491     overflow-x: auto ;
492 }
493
494 .alltt {
495     overflow-x: auto ;
496 }
497
498
499 .bverbatim {
500     margin: 1ex 0pt 1ex 0pt ;
501     padding: .5ex 0pt .5ex 0pt ;
502     overflow-x: auto ;
503 }
504
505 .lverbatim {
506     margin: 1ex 0pt 1ex 0pt ;
507     padding: .5ex 0pt .5ex 0pt ;
508     overflow-x: auto ;
509 }
510
511 .fancyvrb {
512     font-size:.85em ;
513     margin: 3ex 0pt 3ex 0pt
514 }
515
516 .fancyvrblabel {
517     font-weight:bold;
518     text-align: center ;
519 }
520
521
522 .verse {
523     font-family: "Linux Libertine Mono 0", "Lucida Console",
524         "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
525         "Liberation Mono", "FreeMono", "Andale Mono",
526         "Nimbus Mono L", "Courier New", monospace;
527     margin-left: 1em ;
528 }
529
530
531 div.singlespace { line-height: 1.2 ; }
532 div.onehalfspace { line-height: 1.5 ; }
```

```
533 div.doublespace { line-height: 2 ; }
534
535
536
537
538
539 /* Minipage environments, vertically aligned to top, center, bottom: */
540 .minipage {
541     /* display: inline-block ; */
542     /* Mini pages which follow each other will be tiled. */
543     margin: .25em .25em .25em .25em;
544     padding: .25em .25em .25em .25em;
545     display: inline-flex;
546     flex-direction: column ;
547     overflow: auto;
548 }
549
550 /* Paragraphs in the flexbox did not collapse their margins. */
551 /* Have not yet researched this. */
552 .minipage p {margin: .75ex 0em .75ex 0em ;}
553
554
555
556 .framebox {
557     margin: 0ex ;
558     padding: 0ex ;
559     border: 1px solid black;
560     border-radius: 0px ;
561     padding: .3ex .2em 0ex .2em ;
562     margin: .1ex ;
563     display: inline-block ;
564 }
565
566
567 .mdframed {
568     /* padding: 0ex ; */
569     /* border: 1px solid blaefck; */
570     /* border-radius: 0px ; */
571     padding: 0ex ;
572     margin: 3ex 5% 3ex 5% ;
573     /* display: inline-block ; */
574 }
575
576 .mdframed p { padding: 0ex .5em 0ex .5em ; }
577
578 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
579
580 .mdframedtitle {
581     padding: .5em ;
582     display: block ;
```

```
583     font-size: 130%  
584 }  
585  
586 .mdframedsubtitle {  
587     padding: 0ex .5em 0ex .5em ;  
588     display: block ;  
589     font-size: 115% ;  
590 }  
591  
592 .mdframedsubsubtitle {  
593     padding: 0ex .5em 0ex .5em ;  
594     display: block ;  
595 }  
596  
597 .mdtheorem {  
598     padding: 0ex .5em 0ex .5em ;  
599     margin: 3ex 5% 3ex 5% ;  
600 /*   display: inline-block ; */  
601 }  
602  
603  
604 /* framed package */  
605 .framed {  
606     margin: 3ex 0em 3ex 0em ;  
607     border: 1px solid black;  
608     border-radius: 0px ;  
609     padding: .3ex 1em 0ex 1em ;  
610     display: block ;  
611 }  
612  
613 .snugframed {  
614     margin: 3ex 0em 3ex 0em ;  
615     border: 1px solid black;  
616     border-radius: 0px ;  
617     display: block ;  
618 }  
619  
620 .framedleftbar {  
621     margin: 3ex 0em 3ex 0em ;  
622     border-left: 3pt solid black;  
623     border-radius: 0px ;  
624     padding: .3ex .2em .3ex 1em ;  
625     display: block ;  
626 }  
627  
628 .framedtitle {  
629 margin: 0em ;  
630 padding: 0em ;  
631     font-size: 130%  
632 }
```

```
633
634 .framedtitle p { padding: .3em }
635
636
637
638 dl {
639   margin: 1ex 2em 1ex 0em;
640   line-height: 1.3;
641 }
642
643 dl dt {
644   margin-top: 1ex;
645   font-weight: bold;
646 }
647
648 dl dd p { margin-top: 0em; }
649
650
651 nav.toc, nav.lof, nav.lot, nav.lol, nav.lothm {
652   font-family: "URW Classico", Optima, "Linux Biolinum 0",
653   "DejaVu Sans", "Bitstream Vera Sans",
654   Geneva, Verdana, sans-serif ;
655   margin-bottom: 4ex ;
656 }
657
658 nav.toc p, nav.lof p, nav.lot p, nav.lol p, nav.lothm p {
659   line-height: 1.2 ;
660   margin-top:.5ex ;
661   margin-bottom:.5ex;
662   font-size: .9em ;
663 }
664
665
666
667 img, img.hyperimage, img.borderimage {
668   max-width: 600px;
669   border: 1px solid silver;
670   box-shadow: 3px 3px 3px #808080 ;
671   padding: .5% ;
672   margin: .5% ;
673   background: none ;
674 }
675
676 img.inlineimage{
677   padding: 0px ;
678   box-shadow: none ;
679   border: none ;
680   background: none ;
681   margin: 0px ;
682   display: inline-block ;
```

```
683     border-radius: 0px ;
684 }
685
686 img.logoimage{
687     max-width: 300px ;
688     box-shadow: 3px 3px 3px #808080 ;
689     border: 1px solid black ;
690     background:none ;
691     padding:0 ;
692     margin:.5ex ;
693     border-radius: 10px ;
694 }
695
696
697 .section {
698 /*
699     To have each section float relative to each other:
700 */
701 /*
702     display: block ;
703     float: left ;
704     position: relative ;
705     background: white ;
706     border: 1px solid silver ;
707     padding: .5em ;
708 */
709     margin: 0ex .5em 0ex .5em ;
710     padding: 0 ;
711 }
712
713
714 figure {
715     margin: 3ex auto 3ex auto ;
716     padding: 1ex 1em 1ex 1em ;
717     overflow-x: auto ;
718 }
719
720
721 /* To automatically center images in figures: */
722 /*
723 figure img.inlineimage {
724     margin: 0ex auto 0ex auto ;
725     display: block ;
726 }
727 /*
728
729 /* To automatically center minipages in figures: */
730 /*
731 figure div.minipage, figure div.minipage div.minipage {
732     margin: 1ex auto 1ex auto ;
```

```
733     display: block ;
734 }
735 */
736
737 figure div.minipage p { font-size: 85% ; }
738
739 figure.subfigure, figure.subtable {
740     display: inline-block ; margin: 3ex 1em 3ex 1em ;
741 }
742
743 figcaption .minipage { margin:0 ; padding: 0 }
744
745 div.floatrow { text-align: center; }
746
747 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
748
749 div.floatfoot { font-size: .85em ;
750     border-top: 1px solid silver ; line-height: 1.2 ; }
751
752 figcaption , .lstlistingtitle {
753     font-size: .85em ;
754     text-align: center ;
755     font-weight: bold ;
756 margin-top: 1ex ;
757 margin-bottom: 1ex ;
758 }
759
760 figure.subfigure figcaption, figure.subtable figcaption {
761     border-bottom: none ; background: none ;
762 }
763
764 div.nonfloatcaption {
765     margin: 1ex auto 1ex auto ;
766     font-size: .85em ;
767     text-align: center ;
768     font-weight: bold ;
769 }
770
771 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
772 figure div.floatrow div.minipage figcaption {
773 border: none ;
774 background: none ;
775 }
776
777
778 table {
779     margin: 1ex auto 1ex auto ;
780     border-collapse: collapse ;
781     border-spacing: 0px ;
782     line-height: 1.3 ;
```

```
783      }
784
785 tr.hline {border-top: 1px solid silver ; margin-top: 0ex ;
786     margin-bottom: 0ex ; } /* for \hline */
787
788 tr.tbrule {border-top: 1px solid black ; margin-top: 0ex ;
789     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
790
791 td {padding: 1ex .5em 1ex .5em ;}
792
793 table td.tdl { text-align: left ; vertical-align: middle ; }
794 table td.tdc { text-align: center ; vertical-align: middle ; }
795 table td.tdr { text-align: right ; vertical-align: middle ; }
796 table td.tdp { text-align: left ; vertical-align: bottom ; }
797 table td.tdm { text-align: left ; vertical-align: middle ; }
798 table td.tdb { text-align: left ; vertical-align: top ; }
799 table td.tdP { text-align: center ; vertical-align: bottom ; }
800 table td.tdM { text-align: center ; vertical-align: middle ; }
801 table td.tdB { text-align: center ; vertical-align: top ; }
802 table td.tdlrule { text-align: left ; border-top: 1px solid silver ;
803     vertical-align: middle ; } /* for cmidrule */
804 table td.tdcrule { text-align: center ; border-top: 1px solid silver ;
805     vertical-align: middle ; }
806 table td.tdrrule { text-align: right ; border-top: 1px solid silver ;
807     vertical-align: middle ; }
808 table td.tdprule { text-align: left ; border-top: 1px solid silver ;
809     vertical-align: bottom ; }
810 table td.tdmrule { text-align: left ; border-top: 1px solid silver ;
811     vertical-align: middle ; }
812 table td.tdbrule { text-align: left ; border-top: 1px solid silver ;
813     vertical-align: top ; }
814 table td.tdPrule { text-align: center ; border-top: 1px solid silver ;
815     vertical-align: bottom ; }
816 table td.tdMrule { text-align: center ; border-top: 1px solid silver ;
817     vertical-align: middle ; }
818 table td.tdBrule { text-align: center ; border-top: 1px solid silver ;
819     vertical-align: top ; }
820
821 /* Margins of paragraphs inside table cells: */
822 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
823     padding-bottom: 1ex ; margin: 0ex ; }
824 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
825     padding-bottom: 1ex ; margin: 0ex ; }
826 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
827     padding-bottom: 1ex ; margin: 0ex ; }
828
829 td.tdp , td.tdprule , td.tdP , td.tdPrule
830     { padding: 0ex .5em 0ex .5em ; }
831 td.tdm , td.tdmrule , td.tdM , td.tdMrule
832     { padding: 0ex .5em 0ex .5em ; }
```

```
833 td.tdb , td.tdbrule , td.tdB , td.tdBrule
834     { padding: 0ex .5em 0ex .5em ; }
835
836
837 /* table notes: */
838 .tnotes {
839     margin: 0ex 5% 1ex 5% ;
840     padding: 0.5ex 1em 0.5ex 1em;
841     font-size:.85em;
842     text-align: left ;
843 }
844
845 .tnotes dl dt p {margin-bottom:0px;}
846
847 .tnoteitemheader {margin-right: 1em;}
848
849
850
851 /* center, flushleft, flushright environments */
852 div.center{text-align:center;}
853 div.center table {margin-left:auto;margin-right:auto;}
854 div.flushleft{text-align:left;}
855 div.flushleft table {margin-left:0em ; margin-right:auto;}
856 div.flushright{text-align:right;}
857 div.flushright table {margin-left:auto ; margin-right: 0em ;}
858
859
860
861
862 /* program listing callouts: */
863 span.callout {
864     font-family: "DejaVu Sans", "Bitstream Vera Sans",
865             Geneva, Verdana, sans-serif ;
866     border-radius: .5em;
867     background-color:black;
868     color:white;
869     padding:0px .25em 0px .25em;
870 margin: 0 ;
871     font-weight: bold;
872     font-size:.72em ;
873 }
874
875 div.programlisting pre.verbatim span.callout{
876 font-size: .85em ;
877 }
878
879
880
881
882
```

```
883 div.published
884 {
885     text-align: center ;
886     font-variant: normal ;
887     font-style: italic ;
888     font-size: 1em ;
889     margin: 3ex 0em 3ex 0em ;
890 }
891
892 div.subtitle
893 {
894     text-align: center ;
895     font-variant: normal ;
896     font-style: italic ;
897     font-size: 1.25em ;
898     margin: 3ex 0em 3ex 0em ;
899 }
900
901 div.subtitle p { margin: 1ex ; }
902
903 div.author
904 {
905     font-variant: normal ;
906     font-style: normal ;
907     font-size: 1em ;
908     margin: 3ex 0em 3ex 0em ;
909 }
910
911 div.author table {
912     margin: 3ex auto 0ex auto ;
913     background: none ;
914 }
915
916 div.author table tbody tr td { padding: .25ex ; }
917
918 span.affiliation {font-size: .85em ; font-variant: small-caps; }
919
920 div.titledate {
921     text-align: center ;
922     font-size: .85em ;
923     font-style: italic;
924     margin: 6ex 0em 6ex 0em ;
925 }
926
927
928 nav.topnavigation{
929     text-align: left ;
930     padding: 0.5ex 1em 0.5ex 1em ;
931 /*     margin: 2ex 0em 3ex 0em ; */
932     margin: 0 ;
```

```
933     border-bottom: 1px solid silver ;
934     border-top: 1px solid silver ;
935     clear:right ;
936 }
937
938 nav.botnavigation{
939     text-align: left ;
940     padding: 0.5ex 1em 0.5ex 1em ;
941 /*     margin: 3ex 0em 2ex 0em ; */
942     margin: 0 ;
943     border-top: 1px solid silver ;
944     border-bottom: 1px solid silver ;
945     clear:right ;
946 }
947
948
949 header{
950     line-height: 1.2 ;
951     font-size: 1em ;
952 /*     border-bottom: 2px solid silver ; */
953     margin: 0px ;
954     padding: 0ex 1em 0ex 1em ;
955     text-align:center ;
956 }
957
958 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
959
960
961 footer{
962     font-size: .85em ;
963     line-height: 1.2 ;
964     margin-top: 1ex ;
965     border-top: 2px solid silver ;
966     padding: 2ex 1em 2ex 1em ;
967     clear:right ;
968     text-align:left ;
969 }
970
971
972 a.linkhome { font-weight:bold ; font-size: 1em ;}
973
974
975 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
976
977 img.lateximage{
978     padding: 0px 0px 0px 0px ;
979     box-shadow: none ;
980     border: none ;
981     background: none ;
982     margin: 0px 0px -.15ex 0px ;
```

```
983     /* pdfcrop leaves a slight margin, adjust to baseline */
984     max-width: 100% ;
985     border-radius: 0ex ;
986     border: none ;
987 }
988
989
990
991 nav.sidetoc {
992     font-family: "DejaVu Serif", "Bitstream Vera Serif",
993         "Lucida Bright", Georgia, serif;
994     float:right ;
995     width: 20%;
996     border-left: 1px solid silver;
997     border-top: 1px solid silver;
998     border-bottom: 1px solid silver;
999 /*     border-top: 2px solid #808080 ; */
1000    background: #FAF7F4 ;
1001    padding: 2ex 0em 2ex 1em ;
1002    margin: 0ex 0em 2ex 1em ;
1003    font-size:.9em ;
1004    border-radius: 20px 0px 0px 20px ;
1005 }
1006
1007 div.sidetoccontents {
1008 /*     border-top: 1px solid silver ; */
1009     overflow-y: auto ;
1010     width: 100% ;
1011     text-align: left ;
1012 }
1013
1014 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
1015     text-indent: 0 ; }
1016 nav.sidetoc p a {color:black ; font-size: .7em ;}
1017 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
1018     border-bottom: 1px solid silver ; }
1019 nav.sidetoc a:hover {text-decoration: underline ; }
1020
1021
1022
1023 section.textbody { margin: 0ex 1em 0ex 1em ;}
1024
1025
1026 div.multicolsheading { -webkit-column-span: all;
1027     -moz-column-span: all; column-span: all; }
1028 div.multicols { -webkit-columns: 3 380px ;
1029     -moz-columns: 3 380px ; columns: 3 380px ; }
1030 div.multicols p {margin-top: 0ex}
1031
1032
```

```
1033
1034 /* Used to support algorithmicx: */
1035 span.floatright { float: right ; }
1036
1037
1038
1039 /* Native LaTeX theorems: */
1041
1042 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
1043 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
1044
1045
1046 /* theorem, amsthm, and ntheorem packages */
1047
1048 span.theoremheader,
1049 span.theoremheaderplain,
1050 span.theoremheaderdefinition,
1051 span.theoremheaderbreak,
1052 span.theoremheadermarginbreak,
1053 span.theoremheaderchangebreak,
1054 span.theoremheaderchange,
1055 span.theoremheadermargin
1056 {
1057 font-style:normal ; font-weight: bold ; margin-right: 1em ;
1058 }
1059
1060 span.amsthmnameplain,
1061 span.amsthmnamedefinition,
1062 span.amsthmnumberplain,
1063 span.amsthmnumberdefinition
1064 {
1065 font-style:normal ; font-weight: bold ;
1066 }
1067
1068
1069 span.amsthmnameremark,
1070 span.amsthmnumberremark
1071 {font-style:italic ; font-weight: normal ; }
1072
1073
1074 span.amsthmnoteplain,
1075 span.amsthmnotedefinition
1076 {font-style:normal ;}
1077
1078
1079 span.theoremheaderremark,
1080 span.theoremheaderproof,
1081 span.amsthmproofname
1082 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
```

```
1083
1084 span.theoremheadersc
1085 {
1086 font-style:normal ;
1087 font-variant: small-caps ;
1088 font-weight: normal ;
1089 margin-right: 1em ;
1090 }
1091
1092 .theoremendmark {float:right}
1093
1094 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
1095 div.theorembodybreak, div.theorembodynonumberbreak,
1096 div.theorembodymarginbreak,
1097 div.theorembodychangebreak,
1098 div.theorembodychange,
1099 div.theorembodymargin
1100 {
1101 font-style:italic;
1102 margin-top: 3ex ; margin-bottom: 3ex ;
1103 }
1104
1105 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
1106 div.theorembodyplainupright, nonumberplainuprightsc,
1107 div.amsthmbodydefinition, div.amsthmbodyremark,
1108 div.amsthmproof
1109 {
1110 font-style: normal ;
1111 margin-top: 3ex ; margin-bottom: 3ex ;
1112 }
1113
1114 span.amsthmnoteremark {}
1115
1116
1117
1118 /*
1119 For CSS LaTeX and related logos:
1120 Based on:
1121 http://edward.oconnor.cx/2007/08/tex-poshlet
1122 http://nitens.org/taraborelli/texlogo
1123 */
1124
1125 .latexlogofont {
1126     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1127             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1128     font-variant: normal ;
1129 }
1130
1131 .latexlogo {
1132     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
```

```
1133     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1134     letter-spacing: .03em ;
1135     font-size: 1.1em;
1136 }
1137
1138 .latexlogo sup {
1139   text-transform: uppercase;
1140   letter-spacing: .03em ;
1141   font-size: 0.85em;
1142   vertical-align: 0.15em;
1143   margin-left: -0.36em;
1144   margin-right: -0.15em;
1145 }
1146
1147 .latexlogo sub {
1148   text-transform: uppercase;
1149   vertical-align: -0.5ex;
1150   margin-left: -0.1667em;
1151   margin-right: -0.125em;
1152   font-size: 1em;
1153 }
1154
1155 .xetexlogo {
1156   font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1157   "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1158   letter-spacing: .03em ;
1159   font-size: 1.1em;
1160 }
1161
1162 /* A smaller gap between Xe and Tex v.s. LaTeX: */
1163 .xetexlogo sub {
1164   text-transform: uppercase;
1165   vertical-align: -0.5ex;
1166   margin-left: -0.0667em;
1167   margin-right: -0.2em;
1168   font-size: 1em;
1169   letter-spacing: .03em ;
1170 }
1171
1172 /* A large gap between Xe and LaTeX v.s. TeX: */
1173 .xelatexlogo sub {
1174   text-transform: uppercase;
1175   vertical-align: -0.5ex;
1176   margin-left: -0.0667em;
1177   margin-right: -.05em;
1178   font-size: 1em;
1179   letter-spacing: .03em ;
1180 }
1181
1182 .amslogo {
```

```
1183     font-family: "TeXGyreChorus", "URW Chancery L",
1184         "Apple Chancery", "ITC Zapf Chancery", "Monotype Corsiva",
1185         "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
1186         "Hoefler Text", Times, "Times New Roman", serif;
1187     font-style: italic;
1188 }
1189
1190 .lyxlogo {
1191     font-family: "URW Classico", Optima, "Linux Biolinum O",
1192         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
1193 Verdana, sans-serif ;
1194 }
1195
1196
1197
1198
1199 /* Only display top and bottom navigation if a small screen: */
1200 /* Hide the sidetoc if a small screen: */
1201 nav.topnavigation { display:none; }
1202 nav.botnavigation { display:none; }
1203
1204 @media screen and (max-width: 45em) {
1205 /*      nav.sidetoc {display:none;} */
1206     nav.sidetoc {
1207         float: none ;
1208         width: 100% ;
1209         margin: 5ex 0px 5ex 0px ;
1210         padding: 0 ;
1211         border-radius: 0 ;
1212         border-bottom: 1px solid black ;
1213         border-top: 1px solid black ;
1214         box-shadow: none ;
1215     }
1216 /*      nav.topnavigation { display:block } */
1217     nav.botnavigation { display:block }
1218     .marginpar {
1219         max-width: 100%;
1220         float: none;
1221         display:block ;
1222         margin: 1ex 1em 1ex 1em ;
1223     }
1224 }
1225
1226 @media print {
1227     body {
1228         font-family: "Linux Libertine O",
1229             "DejaVu Serif", "Bitstream Vera Serif",
1230             "Liberation Serif", "Nimbus Roman No 9 L",
1231             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1232     }
```

```

1233     nav.sidetoc { display:none; }
1234     nav.topnavigation { display: none; }
1235     nav.botnavigation { display: none; }
1236 }
1237 @media handheld {
1238     nav.sidetoc { display:none; }
1239     nav.topnavigation { display:block }
1240     nav.botnavigation { display:block }
1241 }
1242 }
1243 @media projection {
1244     nav.sidetoc { display:none; }
1245     nav.topnavigation { display:block }
1246     nav.botnavigation { display:block }
1247 }
1248 }
1249 \end{VerbatimOut}
1250 % \end{Verbatim} for syntax highlighting

```

73.4 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.

```

1251 \begin{VerbatimOut}{lwarp_sagebrush.css}
1252 @import url("lwarp.css") ;
1253
1254
1255 A:link {color:#105030 ; text-decoration: none ; }
1256 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
1257 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
1258 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
1259
1260
1261
1262 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1263 {
1264     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1265             "Linux Libertine 0", "Liberation Serif",
1266             "Nimbus Roman No 9 L", "FreeSerif",
1267             "Hoefler Text", Times, "Times New Roman", serif;
1268     font-variant: small-caps ;
1269 font-weight: normal ;
1270     color: #304070 ;
1271     text-shadow: 2px 2px 3px #808080;

```

```
1272 }
1273
1274 h1 { /* title of the entire website, used on each page */
1275     font-variant: small-caps ;
1276     color: #304070 ;
1277     text-shadow: 2px 2px 3px #808080;
1278     background-color: #F7F7F0 ;
1279     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
1280 }
1281
1282 h1 {
1283     border-bottom: 1px solid #304070;
1284     border-top: 2px solid #304070;
1285 }
1286
1287 h2 {
1288     border-bottom: 1px solid #304070;
1289     border-top: 2px solid #304070;
1290     background-color: #F7F7F0 ;
1291     background-image: linear-gradient(to bottom, #F7F7F0, #DADOC0);
1292 }
1293
1294
1295
1296 div.abstract {
1297     background: #f5f5eb ;
1298     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1299
1300     border: 1px solid silver;
1301     border-radius: 1em ;
1302 }
1303
1304 div.abstract dl {line-height:1.5;}
1305 div.abstract dt {color:#304070;}
1306
1307 div.abstracttitle{
1308     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1309         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
1310         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1311     font-weight:bold;
1312     font-variant: small-caps ;
1313     font-size:1.5em;
1314     border-bottom: 1px solid silver ;
1315     color: #304070 ;
1316     text-align: center ;
1317     text-shadow: 1px 1px 2px #808080;
1318 }
1319
1320 span.abstractrunintitle{
1321     font-family: "URW Classico", Optima, "Linux Biolinum 0",
```

```
1322     "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1323     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1324     font-weight:bold;
1325 }
1326
1327
1328 div.epigraph {
1329     background: #f5f5eb ;
1330     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1331
1332     border: 1px solid silver ;
1333     border-radius: 1ex ;
1334     box-shadow: 3px 3px 3px #808080 ;
1335 }
1336
1337
1338 .example {
1339     background-color: #f5f5eb ;
1340     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1341
1342 }
1343
1344 div.exampletitle{
1345     font-family: "URW Classico", Optima, "Linux Biolinum O",
1346     "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1347     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1348     font-weight:bold;
1349     font-variant: small-caps ;
1350     border-bottom: 1px solid silver ;
1351     color: #304070 ;
1352     text-align: center ;
1353     text-shadow: 1px 1px 2px #808080;
1354 }
1355
1356
1357 .sidebar {
1358     background-color: #f5f5eb ;
1359     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1360
1361 }
1362
1363 div.sidebartitle{
1364     font-family: "URW Classico", Optima, "Linux Biolinum O",
1365     "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1366     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1367     font-weight:bold;
1368     font-variant: small-caps ;
1369     border-bottom: 1px solid silver ;
1370     color: #304070 ;
1371     text-align: center ;
```

```
1372     text-shadow: 1px 1px 2px #808080;
1373 }
1374
1375
1376 .fancyvrblabel {
1377     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1378         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
1379         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1380     font-weight:bold;
1381     font-variant: small-caps ;
1382 font-size: 1.5em ;
1383     color: #304070 ;
1384     text-align: center ;
1385     text-shadow: 1px 1px 2px #808080;
1386 }
1387
1388
1389
1390 .minipage {
1391     background-color: #eeeeee7 ;
1392     border: 1px solid silver ;
1393 border-radius: 1ex ;
1394 }
1395
1396 .framed .minipage , .framedleftbar .minipage {
1397 border: none ;
1398 background: none ;
1399 padding: 0ex ;
1400 margin: 0ex ;
1401 }
1402
1403 figure.figure .minipage, figcaption .minipage { border: none; }
1404
1405 div.marginblock div.minipage { border: none; }
1406
1407 figure , div.marginblock {
1408     background-color: #eeeeee7 ;
1409     border: 1px solid silver ;
1410     border-radius: 1ex ;
1411     box-shadow: 3px 3px 3px #808080 ;
1412 }
1413
1414 figure figure {
1415     border: 1px solid silver ;
1416     margin: 0em ;
1417 box-shadow: none ;
1418 }
1419
1420 /*
1421 figcaption {
```

```
1422     border-top: 1px solid silver ;
1423     border-bottom: 1px solid silver ;
1424     background-color: #e8e8e8 ;
1425 }
1426 */
1427
1428
1429 div.table {
1430     box-shadow: 3px 3px 3px #808080 ;
1431 }
1432
1433 /*
1434 .tnotes {
1435     background: #e8e8e8;
1436     border: 1px solid silver;
1437 }
1438 */
1439
1440
1441 nav.topnavigation{
1442     background-color: #b0b8b0 ;
1443     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
1444 }
1445
1446 nav.botnavigation{
1447     background-color: #b0b8b0 ;
1448     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
1449 }
1450
1451
1452
1453 header{
1454     background-color: #F7F7F0 ;
1455     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
1456 }
1457
1458 footer{
1459     background-color: #F7F7F0 ;
1460     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
1461 }
1462
1463
1464
1465 nav.sidetoc {
1466     background-color: #F7F7F0 ;
1467     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
1468     box-shadow: 3px 3px 3px #808080 ;
1469     border-radius: 0px 0px 0px 20px ;
1470 }
1471
```

```

1472 div.sidetootitle {color: #304070 ; }
1473 nav.sidetoc a:hover {color:#006000 ; text-decoration: none ; text-shadow:0px 0px 2px #a0a0a0;}
1474
1475
1476 @media screen and (max-width: 45em) {
1477     nav.sidetoc { border-radius: 0 ; }
1478 }
1479
1480
1481 \end{VerbatimOut}
1482 % \end{Verbatim}% for syntax highlighting

```

73.5 lwarf_formal.css

File `lwarf_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.

```

1483 \begin{VerbatimOut}{lwarf_formal.css}
1484 @import url("lwarf.css") ;
1485
1486
1487
1488 A:link {color:#802020 ; text-decoration:none; }
1489 A:visited {color:#802020 ; text-shadow:none ;}
1490 A:hover {color:#400000 ; text-shadow:none ;}
1491 A:active {color:#C00000 ; text-shadow:none ;}
1492
1493
1494 body {
1495     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1496                 "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1497                 "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1498                 "Times New Roman", serif;
1499     background: #fffcf5;
1500 }
1501
1502 span.textrm {
1503     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1504                 "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1505                 "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1506                 "Times New Roman", serif;
1507 }
1508
1509 span.textsf {
1510     font-family: "DejaVu Sans", "Bitstream Vera Sans",

```

```
1511     Geneva, Verdana, sans-serif ;
1512 }
1513
1514
1515
1516 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1517 {
1518     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1519         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1520         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1521         "Times New Roman", serif;
1522     color: #800000 ;
1523     text-shadow: none ;
1524 }
1525
1526 h1, h2 {
1527     background-color: #ffffcf5 ;
1528     background-image: none ;
1529     border-bottom: 1px solid #808080;
1530     border-top: 2px solid #808080;
1531 }
1532
1533 div.abstracttitle {
1534     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1535         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1536         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1537         "Times New Roman", serif;
1538     color: black ;
1539     text-shadow: none ;
1540 }
1541
1542 span.abstractrunintitle {
1543     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1544         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1545         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1546         "Times New Roman", serif;
1547     color: black ;
1548     text-shadow: none ;
1549 }
1550
1551 div.abstract { font-size: 100% }
1552
1553 .sidebar {
1554     background: #ffffcf5;
1555     background-image: none ;
1556     margin: 2em 5% 2em 5%;
1557     padding: 0.5em 1em;
1558     border: none ;
1559     border-top : 1px solid silver;
1560     border-bottom : 1px solid silver;
```

```
1561   font-size: 90% ;
1562 }
1563
1564 div.sidebartitle{
1565   font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1566     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1567     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1568     "Times New Roman", serif;
1569   color: #800000 ;
1570   text-shadow: none ;
1571   border: none ;
1572 }
1573
1574 .example {
1575   background: #ffffcf5;
1576   background-image: none ;
1577   margin: 2em 5% 2em 5%;
1578   padding: 0.5em 1em;
1579   border: none ;
1580   border-top : 1px solid silver;
1581   border-bottom : 1px solid silver;
1582 }
1583
1584 div.exampletitle{
1585   font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1586     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1587     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1588     "Times New Roman", serif;
1589   color: #800000 ;
1590   text-shadow: none ;
1591   border: none ;
1592 }
1593
1594 div.fancyvrblabel{
1595   font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1596     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1597     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1598     "Times New Roman", serif;
1599   color: #800000 ;
1600   text-shadow: none ;
1601   border: none ;
1602 }
1603
1604
1605
1606 .verse {
1607   font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1608     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1609     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1610     "Times New Roman", serif;
```

```
1611 }
1612
1613
1614 figure {
1615     margin: 3ex 5% 3ex 5% ;
1616     padding: 1ex 1em 1ex 1em ;
1617     background-color: #ffffcf5 ;
1618     overflow-x: auto ;
1619     border: none ;
1620 /*     border-top: 1px solid silver; */
1621 /*     border-bottom: 1px solid silver; */
1622 }
1623
1624
1625 figcaption , .lstlisting {
1626     border: none ;
1627 /*     border-top: 1px solid silver ; */
1628 /*     border-bottom: 1px solid silver ; */
1629     background-color: #ffffcf5 ;
1630 }
1631
1632 .tnotes {
1633     background: #ffffcf5 ;
1634 }
1635
1636 .theorem {
1637     background: none ;
1638 }
1639
1640 .minipage {
1641     background-color: #ffffcf5 ;
1642     border: none ;
1643 }
1644
1645 div.floatrow figure { border: none ; }
1646
1647 figure figure { border: none ; }
1648
1649
1650 nav.toc, nav.lof, nav.lot, nav.lol {
1651     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1652         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1653         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1654         "Times New Roman", serif;
1655 }
1656
1657 nav.sidetoc {
1658     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1659         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1660         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
```

```
1661     "Times New Roman", serif;
1662     background-image: linear-gradient(to bottom, #ffffcf5, #C0C0C0);
1663     border-radius: 0px 0px 0px 20px ;
1664 }
1665
1666 div.sidetotitle{
1667     color: #800000 ;
1668 }
1669
1670 header{
1671     background-color: #e0e0e0 ;
1672     background-image: linear-gradient(to top, #ffffcf5, #b0b0b0);
1673     text-align:center ;
1674 }
1675
1676 footer{
1677     background-color: #e0e0e0 ;
1678     background-image: linear-gradient(to bottom, #ffffcf5, #b0b0b0);
1679     padding: 2ex 1em 2ex 1em ;
1680     clear:right ;
1681     text-align:left ;
1682 }
1683
1684 nav.botnavigation {
1685     background: #dedcd5 ;
1686     border-top: 1px solid black ;
1687 }
1688 \end{VerbatimOut}
1689 % \end{Verbatim}% for syntax highlighting
```

73.6 sample_project.css

File `sample_project.css` The project-specific CSS file. Use with `\NewCSS`.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```
1690 \begin{VerbatimOut}{sample_project.css}
1691 /* ( --- Start of project.css --- ) */
1692 /* A sample project-specific CSS file for lwarp --- ) */
1693
1694 /* Load default lwarp settings: */
1695 @import url("lwarp.css") ;
1696 /* or lwarp_formal.css, lwarp_sagebrush.css */
1697
1698 /* Project-specific CSS setting follow here. */
1699 /* . . . */
```

```

1700
1701 /* ( --- End of project.css --- ) */
1702 \end{VerbatimOut}
1703 % \end{Verbatim}% for syntax highlighting

```

73.7 lwarf_html.xdy

File `lwarf_html.xdy` Used to modify the index for `lwarf`.

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

```

1704 \begin{VerbatimOut}{lwarf_html.xdy}
1705 (require "tex/inputenc/latin.xdy")
1706 (merge-rule "\PS *" "Postscript")
1707 (require "texindy.xdy")
1708 (require "page-ranges.xdy")
1709 (require "book-order.xdy")
1710 (markup-locref :open "\hyperindexref{" :close "}")
1711 \end{VerbatimOut}
1712 % \end{Verbatim}% for syntax highlighting

```

73.8 lwarf_mathjax.txt

File `lwarf_mathjax.txt` Used by `lwarf` when using MathJax.

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

```

1713 \begin{VerbatimOut}{lwarf_mathjax.txt}
1714 <!-- https://groups.google.com/forum/#!topic/
1715           mathjax-users/jUtewUcE2bY -->
1716 <script type="text/x-mathjax-config">
1717 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
1718     var seteqsectionDefault = {name: "", num: 0};
1719     var seteqsections = {}, seteqsection = seteqsectionDefault;
1720     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
1721     var AMS = MathJax.Extension["TeX/AMSmash"];
1722     TEX.Definitions.Add({
1723         macros: {
1724             seteqsection: "mySection",
1725             seteqnumber: "mySetEqNumber"
1726         }
1727     });
1728

```

```
1729     PARSE.Augment({
1730       mySection: function (name) {
1731         seteqsection.num = AMS.number;
1732         var n = this.GetArgument(name);
1733         if (n === "") {
1734           seteqsection = seteqsectionDefault;
1735         } else {
1736           if (!seteqsections["_"+n])
1737             seteqsections["_"+n] = {name:n, num:0};
1738           seteqsection = seteqsections["_"+n];
1739         }
1740         AMS.number = seteqsection.num;
1741       },
1742       mySetEqNumber: function (name) {
1743         var n = this.GetArgument(name);
1744         if (!n || !n.match(/^\ *[0-9]+ *\$/))
1745           n = ""; else n = parseInt(n)-1;
1746         <!-- $ syntax highlighting -->
1747         if (n === "" || n < 1)
1748           TEX.Error
1749           ("Argument to "+name+" should be a positive integer");
1750         AMS.number = n;
1751       }
1752     });
1753     MathJax.Hub.Config({
1754       TeX: {
1755         equationNumbers: {
1756           formatTag: function (n)
1757             {return "("+(seteqsection.name+"."+n).replace(/\.\./,"")+")"},
1758           formatID: function (n) {
1759             n = (seteqsection.name+'.'+n).replace
1760               (/[:<>]/g,"").replace(/\.\./,"");
1761             return 'mjax-eqn-' + n;
1762           }
1763         }
1764       }
1765     });
1766   });
1767 </script>
1768
1769 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
1770 <script type="text/x-mathjax-config">
1771   MathJax.Ajax.config.path["Contrib"] =
1772     "https://cdn.mathjax.org/mathjax/contrib";
1773 </script>
1774
1775 <!-- https://github.com/mathjax/MathJax-third-party-extensions/
1776                               tree/master/siunitx -->
1777 <script type="text/x-mathjax-config">
1778   MathJax.Hub.Config({
```

```

1779     extensions: ["tex2jax.js", "[Contrib]/siunitx/siunitx.js"],
1780     jax: ["input/TeX", "output/HTML-CSS"],
1781     tex2jax: {inlineMath: [[ "$", "$"], ["\\(", "\\)"]]},
1782     TeX: {extensions: ["AMSmath.js", "AMSsymbols.js", "sinuitx.js"]}
1783   );
1784 </script>
1785
1786 <script type="text/x-mathjax-config">
1787 MathJax.Hub.Config({
1788   TeX: {
1789     equationNumbers: {
1790       autoNumber: "AMS"
1791     }
1792   }
1793 });
1794 </script>
1795
1796 <!-- Alternative CDN provider: -->
1797 <script type="text/javascript" async
1798   src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.0/MathJax.js?config=TeX-AMS_HTML-full"
1799 </script>
1800
1801 <!-- No longer supported after April 30, 2017: -->
1802 <!--
1803 <script
1804   src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
1805 </script>
1806 -->
1807
1808 \end{VerbatimOut}
1809 % \end{Verbatim}% for syntax highlighting

```

73.9 Lwarpmk option

The following is only generated if the `lwarpmk` option was given to `lwarf-newproject`.

```
1810 \begin{LWR@createlwarpmk}
```

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

```

1811 \begin{VerbatimOut}{lwarpmk.lua}
1812 #!/usr/bin/env texlua
1813
1814 -- Copyright 2016–2017 Brian Dunn
1815
1816 -- Print the usage of the lwarpmk command:
1817

```

```
1818 printversion = "v0.29"
1819
1820 function printhelp ()
1821 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
1822 end
1823
1824 function printusage ()
1825 print ( [[
1826
1827 lwarpmk print [project]: Compile a print version.
1828 lwarpmk printindex [project]: Process the index for the print version.
1829 lwarpmk printglossary [project]: Process the glossary for the print version.
1830 lwarpmk html [project]: Compile an HTML version.
1831 lwarpmk htmlindex [project]: Process the index for the html version.
1832 lwarpmk htmlglossary [project]: Process the glossary for the html version.
1833 lwarpmk again [project]: Touch the source code to trigger recompiles.
1834 lwarpmk limages [project]: Process the "lateximages" created by l warp.sty.
1835 lwarpmk pdftohtml [project]:
1836     For use with latexmk or a Makefile:
1837     Convert project_html.pdf to project_html.html and
1838     individual HTML files.
1839 lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, .gl*
1840 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
1841 lwarpmk -h: Print this help message.
1842 lwarpmk --help: Print this help message.
1843
1844 ]] )
1845 printconf ()
1846 end
1847
1848 -- Print the format of the configuration file lwarpmk.conf:
1849
1850 function printconf ()
1851 print ( [[
1852 An example lwarpmk.conf or <project>.lwarpmkconf project file:
1853 --
1854 opsystem = "Unix"  (or "Windows")
1855 latexname = "pdflatex" (or "lualatex", or "xelatex")
1856 sourcename = "projectname" (the source-code filename w/o .tex)
1857 homehtmlfilename = "index" (or perhaps the project name)
1858 htmlfilename = "" (or "projectname" - filename prefix)
1859 uselatexmk = "false" (or "true" to use latexmk to build PDFs)
1860 languge = "english" (use a language supported by xindy)
1861 --
1862 Filenames must contain only letters, numbers, underscore, or dash.
1863 Values must be in "quotes".
1864
1865 ]] );
1866 end
1867
```

```
1868
1869 -- Split one large sourcefile into a number of files,
1870 -- starting with destfile.
1871 -- The file is split at each occurrence of <!--|Start file|newfilename|*
1872
1873 function splitfile (destfile,sourcefile)
1874 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
1875 io.input(sourcefile)
1876 io.output(destfile)
1877 for line in io.lines() do
1878 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
1879 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
1880 io.output(newfilename) ;
1881 else -- not a splitpoint
1882 io.write (line .. "\n") ;
1883 end
1884 end -- do
1885 end -- function
1886
1887 -- Incorrect value, so print an error and exit.
1888
1889 function cvalueerror ( line, linenum , cvalue )
1890     print ( linenum .. " : " .. line ) ;
1891     print ("lwarpmk: incorrect variable value \" .. cvalue .. \" in lwarpmk.conf.\n" ) ;
1892     printconf () ;
1893     os.exit(1) ;
1894 end
1895
1896 -- Load settings from the project's "lwarpmk.conf" file:
1897
1898 function loadconf ()
1899 -- Default configuration filename:
1900 local conffile = "lwarpmk.conf"
1901 -- Optional configuration filename:
1902 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
1903 -- Default language:
1904 language = "english"
1905 -- Verify the file exists:
1906 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
1907 print("lwarpmk: " .. conffile .. " does not exist.")
1908 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
1909 printhelp () ;
1910 os.exit(1) -- exit the entire lwarpmk script
1911 else -- file exists
1912 -- Read the file:
1913 print ("lwarpmk: Reading " .. conffile ..".")
1914 io.input(conffile) ;
1915 -- Scan each line:
1916 local linenum = 0
1917 for line in io.lines() do -- scan lines
```

```
1918 linenum = linenum + 1
1919 i,j,cvarname,cvalue = string.find (line,"([%w_]*)%s*=%s*\\"([%w_]*)\\\"") ;
1920 -- Error if incorrect enclosing characters:
1921 if ( i == nil ) then
1922 print ( linenum .. " : " .. line ) ;
1923 print ( "lwarfpmk: Incorrect entry in " .. conffile .."\n" ) ;
1924 printconf () ;
1925 os.exit(1) ;
1926 end
1927 if ( cvarname == "opsystem" ) then
1928     -- Verify choice of opsystem:
1929     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
1930         opsystem = cvalue
1931     else
1932         cvalueerror ( line, linenum , cvalue )
1933     end
1934 elseif ( cvarname == "latexname" ) then
1935     -- Verify choice of LaTeX compiler:
1936     if (
1937         (cvalue == "pdflatex") or
1938         (cvalue == "xelatex") or
1939         (cvalue == "lualatex")
1940     ) then
1941         latexname = cvalue
1942     else
1943         cvalueerror ( line, linenum , cvalue )
1944     end
1945 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
1946 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
1947 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
1948 elseif ( cvarname == "uselatexmk" ) then uselatexmk = cvalue
1949 elseif ( cvarname == "language" ) then language = cvalue
1950 else
1951 print ( linenum .. " : " .. line ) ;
1952 print ( "lwarfpmk: Incorrect variable name \" .. cvarname .. \" in " .. conffile .."\n" ) ;
1953 printconf () ;
1954 os.exit(1) ;
1955 end
1956 end -- do scan lines
1957 end -- file exists
1958 -- Select some operating-system commands:
1959 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
1960 rmname = "rm"
1961 mvname = "mv"
1962 touchnamepre = "touch"
1963 touchnamepost = ""
1964 dirslash = "/"
1965 opquote= "\'"
1966 elseif opsystem=="Windows" then -- For Windows
1967 rmname = "DEL"
```

```
1968 mvname = "MOVE"
1969 touchnamepre = "COPY /b"
1970 touchnamepost = "+,,"
1971 dirslash = "\\"
1972 opquote= "\""
1973 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
1974 end --- for Windows
1975 end -- loadconf
1976
1977
1978 function refreshdate ()
1979 os.execute(touchnamepre .. " " .. sourcename .. ".tex" .. touchnamepost)
1980 end
1981
1982
1983 -- Scan the LaTeX log file for the phrase "Rerun to get",
1984 -- indicating that the file should be compiled again.
1985 -- Return true if found.
1986
1987 function reruntoget (filesource)
1988 io.input(filesource)
1989 for line in io.lines() do
1990 if ( string.find(line,"Rerun to get") ~= nil ) then return true end
1991 end
1992 return false
1993 end
1994
1995 -- Compile one time, return true if should compile again.
1996 -- fsuffix is "" for print, "_html" for HTML output.
1997
1998 function onetime (fsuffix)
1999 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
2000 err = os.execute(
2001 --     "echo " ..
2002     latexname .. " " .. sourcename..fsuffix )
2003 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2004 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
2005 end
2006
2007 -- Compile up to five times.
2008 -- fsuffix is "" for print, "_html" for HTML output
2009
2010 function manytimes (fsuffix)
2011 if onetime(fsuffix) == true then
2012 if onetime(fsuffix) == true then
2013 if onetime(fsuffix) == true then
2014 if onetime(fsuffix) == true then
2015 if onetime(fsuffix) == true then
2016 end end end end end
2017 end
```

```
2018
2019 -- Exit if the given file does not exist.
2020
2021 function verifyfileexists (filename)
2022 if (lfs.attributes ( filename , "modification" ) == nil ) then
2023 print ( "lwarpmk: " .. filename .. " not found." ) ;
2024 os.exit (1) ;
2025 end
2026 end
2027
2028 -- Convert <project>_html.pdf into HTML files:
2029
2030 function pdftohtml ()
2031     -- Convert to text:
2032     print ("lwarpmk: Converting " .. sourcename .. "_html.pdf to " .. sourcename .. "_html.html")
2033     os.execute("pdftotext -enc UTF-8 -nopgbrk -layout " .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
2034     -- Split the result into individual HTML files:
2035     splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
2036 end
2037
2038 -- Remove auxiliary files:
2039
2040 function removeaux ()
2041     os.execute ( rname .. " " ..
2042                 sourcename .. ".aux" .. sourcename .. "_html.aux" ..
2043                 sourcename .. ".toc" .. sourcename .. "_html.toc" ..
2044                 sourcename .. ".lof" .. sourcename .. "_html.lof" ..
2045                 sourcename .. ".lot" .. sourcename .. "_html.lot" ..
2046                 sourcename .. ".idx" .. sourcename .. "_html.idx" ..
2047                 sourcename .. ".ind" .. sourcename .. "_html.ind" ..
2048                 sourcename .. ".log" .. sourcename .. "_html.log" ..
2049                 sourcename .. ".gl*" .. sourcename .. "_html.gl*" )
2050
2051 end
2052
2053
2054
2055 -- Create lateximages based on lateximages.txt:
2056 function createlateximages ()
2057 print ("lwarpmk: Creating lateximages.")
2058 io.input("lateximages.txt")
2059 -- Create the lateximages directory, ignore error if already exists
2060 err = os.execute("mkdir lateximages")
2061 -- Scan lateximages.txt
2062 for line in io.lines() do
2063     -- lwimgpage is the page number in the PDF which has the image
2064     -- lwimgnum is the sequential lateximage number to assign for the image
2065     i,j,lwimgpage,lwimgnum = string.find (line,"|(.*)|(.*)|")
2066     -- For each entry:
2067     if ( (i~=nil) ) then
```

```
2068 -- Separate out the image into its own single-page pdf:
2069 err = os.execute(
2070 "pdfseparate -f " .. lwimgpage .. " -l " ..
2071 lwimgpage .. " " .. sourcename .."_html.pdf lateximagetemp-%d.pdf")
2072 -- Crop the image:
2073 err = os.execute(
2074 "pdfcrop --hires lateximagetemp-" .. lwimgpage ..".pdf lateximage-" .. lwimgnum ..".pdf")
2075 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2076 -- Convert the image to svg:
2077 err = os.execute(
2078 "pdftocairo -svg lateximage-" .. lwimgnum ..".pdf lateximage-" .. lwimgnum ..".svg")
2079 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2080 -- Move the result into lateximages/:
2081 err = os.execute(
2082 mvname .. " lateximage-" .. lwimgnum ..".svg lateximages" .. dirslash )
2083 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2084 -- Remove the temporary files:
2085 err = os.execute(
2086 rmname .. " lateximage-" .. lwimgnum ..".pdf lateximagetemp-" .. lwimgpage ..".pdf")
2087 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2088 end
2089 end -- do
2090 end -- function
2091
2092
2093
2094
2095 -- lwarpmk --version :
2096
2097 if (arg[1] == "--version") then
2098 print ( "lwarpmk: " .. printversion )
2099
2100 else -- not -- version
2101
2102 -- print intro:
2103
2104 print ("lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package.")
2105
2106 -- lwarpmk print:
2107
2108 if arg[1] == "print" then
2109 loadconf ()
2110 if ( uselatexmk == "true" ) then
2111     os.execute ( "latexmk -pdf -dvi- -ps- -pdflatex=\"\" .. latexname .. "%0 %S\" " .. sourcename .. ".tex" )
2112     print ("lwarpmk: Done.")
2113 else -- not latexmk
2114     verifyfileexists (sourcename .. ".tex");
2115     -- See if up to date:
2116     if (
2117         ( lfs.attributes (sourcename .. ".pdf", "modification") == nil ) or
```

```
2118      (
2119          lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2120          lfs.attributes ( sourcename .. ".pdf" , "modification" )
2121      )
2122  ) then
2123      -- Recompile if not yet up to date:
2124      manytimes("")
2125      print ("lwarpmk: Done.") ;
2126  else
2127      print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
2128  end
2129 end -- not latexmk
2130
2131 -- lwarf printindex:
2132 -- Compile the index then touch the source
2133 -- to trigger a recompile of the document:
2134
2135 elseif arg[1] == "printindex" then
2136 loadconf ()
2137 print ("lwarpmk: Processing the index.")
2138 os.execute("texindy -M lwarf_html.xdy " .. sourcename .. ".idx")
2139 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2140 refreshdate ()
2141 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2142 print ("lwarpmk: Done.")
2143
2144 -- lwarf printglossary:
2145 -- Compile the glossary then touch the source
2146 -- to trigger a recompile of the document:
2147
2148 elseif arg[1] == "printglossary" then
2149 loadconf ()
2150 print ("lwarpmk: Processing the glossary.")
2151
2152 os.execute("xindy -L " .. language .. " -C utf8 -I xindy -M " .. sourcename ..
2153     " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
2154     .. sourcename .. ".glo")
2155 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2156 refreshdate ()
2157 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2158 print ("lwarpmk: Done.")
2159
2160 -- lwarpmk html:
2161
2162 elseif arg[1] == "html" then
2163 loadconf ()
2164 if ( uselatexmk == "true" ) then
2165     -- The recorder option is required to detect changes in <project>.tex
2166     -- while we are loading <project>.html.tex.
2167     err=os.execute ( "latexmk -pdf -dvi- -ps- --recorder "
```

```
2168      .. "-e "
2169      .. opquote .. "$makeindex = q/texindy -M lwarf_html.xdy/" .. opquote
2170      .. " -pdflatex="" .. latexname .. "%0 %S\" "
2171      .. sourcename .. "_html.tex" ) ;
2172      if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2173      pdftohtml ()
2174      print ("lwarpmk: Done.")
2175 else -- not latexmk
2176   verifyfileexists ( sourcename .. ".tex" ) ;
2177   -- See if exists and is up to date:
2178   if (
2179     ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
2180     (
2181       lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2182       lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
2183     )
2184   ) then
2185     -- Recompile if not yet up to date:
2186     manytimes("_html")
2187     pdftohtml ()
2188     print ("lwarpmk: Done.")
2189   else
2190     print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
2191   end
2192 end -- not latexmk
2193
2194 elseif arg[1] == "pdftohtml" then
2195   loadconf ()
2196   pdftohtml ()
2197
2198 -- lwarpmk htmlindex:
2199 -- Compile the index then touch the source
2200 -- to trigger a recompile of the document:
2201
2202 elseif arg[1] == "htmlindex" then
2203 loadconf ()
2204 print ("lwarpmk: Processing the index.")
2205 os.execute("texindy -M lwarf_html.xdy " .. sourcename .. "_html.idx")
2206 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2207 refreshdate ()
2208 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2209 print ("lwarpmk: Done.")
2210
2211 -- lwarpmk htmlglossary:
2212 -- Compile the glossary then touch the source
2213 -- to trigger a recompile of the document:
2214
2215 elseif arg[1] == "htmlglossary" then
2216 loadconf ()
2217 print ("lwarpmk: Processing the glossary.")
```

```
2218
2219 os.execute("xindy -L .. language .. " -C utf8 -I xindy -M ..sourcename ..
2220     "_html -t .. sourcename .. "_html.glg -o ..sourcename ..
2221     "_html.gls ..sourcename .. "_html.glo")
2222
2223 print ("lwarpmk: Forcing an update of .. sourcename ..".tex.")
2224 refreshdate ()
2225 print ("lwarpmk: .. sourcename ..".tex is ready to be recompiled.")
2226 print ("lwarpmk: Done.")
2227
2228 -- lwarpmk limages:
2229 -- Scan the lateximages.txt file to create lateximages,
2230 -- then touch the source to trigger a recompile.
2231
2232 elseif arg[1] == "limages" then
2233 loadconf ()
2234 print ("lwarpmk: Processing images.")
2235 createlateximages ()
2236 print ("lwarpmk: Forcing an update of .. sourcename ..".tex.")
2237 refreshdate ()
2238 print ("lwarpmk: .. sourcename ..".tex is ready to be recompiled.")
2239 print ("lwarpmk: Done.")
2240
2241 -- lwarpmk again:
2242 -- Touch the source to trigger a recompile.
2243
2244 elseif arg[1] == "again" then
2245 loadconf ()
2246 print ("lwarpmk: Forcing an update of .. sourcename ..".tex.")
2247 refreshdate ()
2248 print ("lwarpmk: .. sourcename ..".tex is ready to be recompiled.")
2249 print ("lwarpmk: Done.")
2250
2251 -- lwarpmk clean:
2252 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, .gl*
2253
2254 elseif arg[1] == "clean" then
2255 loadconf ()
2256 removeaux ()
2257 print ("lwarpmk: Done.")
2258
2259 -- lwarpmk cleanall
2260 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, .gl*
2261 --     and also project.pdf, *.html
2262
2263 elseif arg[1] == "cleanall" then
2264 loadconf ()
2265 removeaux ()
2266 os.execute ( rmname .. " " ..
2267     sourcename .. ".pdf" .. sourcename .. "_html.pdf" ..
```

```
2268      "*.html"
2269      )
2270 print ("lwarpmk: Done.")
2271
2272 -- lwarpmk with no argument :
2273
2274 elseif (arg[1] == nil) then
2275 printhelp ()
2276
2277 -- lwarpmk -h or lwarpmk --help :
2278
2279 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
2280 printusage ()
2281
2282 else
2283 print ("lwarpmk: Unknown command \"..arg[1]..\".\n")
2284 printhelp ()
2285 end
2286
2287 end -- not --version
2288 \end{VerbatimOut}
2289 % \end{Verbatim}% for syntax highlighting

2290 \end{LWR@createlwarpmk}
```

Package 3

lwarf-abstract.sty

74 Abstract

(Based on original code by PETER WILSON.)

Pkg **abstract** abstract is supported and patched by lwarf.

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 lwarf-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{\@bsruntitl}{%
11 \hspace*{\abstitleskip}%
12 {\abstractnamefont}%
13 \InlineClass{abstractruntitle}{\abstractname}%
14 \@bslabeldelim}%
15 }
16
17 \if@titlepage
18   \renewenvironment{abstract}{%
19 %     \titlepage
20     \null\vfil
21     \begin{parpenalty}\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\OM
33      \end\absnamepos%
34 %%      \vspace{\abstitleskip}%
35      \fi
36      \fi
37      \vspace{\abstitleskip}%
38  \fi
39  \put@bsintoc%
40  \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
41  {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
42 }
43 \else
44  \renewenvironment{abstract}{%
45    \if@bsrunin
46    \else
47      \if@bsstyle
48        \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
49    \else
50      \ifnumber@bs
51        \num@bs
52      \else
53 \begin{\absnamepos}%
54 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
55 \end\absnamepos%
56 %%      \vspace{\abstitleskip}%
57      \fi
58      \fi
59      \vspace{\abstitleskip}%
60    \fi
61    \put@bsintoc%
62    \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
63  {\par\end{@bstr@ctlist}}
64 \fi
65
```

Package 4

l warp-afterpage.sty

75 Afterpage

Pkg **afterpage** Not used.

for HTML output: Discard all options for l warp-afterpage:

```
1 \LWR@ProvidesPackageDrop{afterpage}
2 \newcommand{\afterpage}[1]{#1}
```

Package 5

lwarf-algorithmicx.sty

76 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 [159.1](#).

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 6

l warp-alltt.sty

77 Alltt

Pkg alltt alltt is patched for use by l warp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{alltt}

2 \AfterEndPreamble{
3 \AtBeginEnvironment{alltt}{%
4 \LWR@forcenewpage
5 \LWR@atbeginverbatim{alltt}\unskip\vspace*{-\baselineskip}%
6 }
7 \AfterEndEnvironment{alltt}{\unskip\vspace*{-\baselineskip}\LWR@afterendverbatim}
8 }
```

Package 7

l warp-amsthm.sty

78 AMSthm

(Based on original code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)

Pkg **amsthm** amsthm is patched for use by l warp.

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}%
8 }{%
9   \thm@style{#1}%
10 \renewcommand{\LWR@newtheoremstyle}{#1}%
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@\oplus6\p@\relax
70 \trivlist
71 \item[\hspace*{1em}\labelsep
72 \quad \InLineClass{amsthmproofname}{\#1\@addpunct{.}}]\ignorespaces% changes
73 }{%
74 \InLineClass{theoremendmark}{\popQED}\endtrivlist%
75 \endBlockClass% new
76 \@endpefalse
77 }
```

Package 8

lwarf-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 lwarf-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

lwarf-booktabs.sty

80 Booktabs

Pkg booktabs booktabs is emulated during HTML output, and the booktabs package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{booktabs}

Booktabs emulation is spread among the tabular code.

Emulated for source compatibility.

2 \newcommand*{\addlinespace}[1]{}
3 \newcommand*{\morecmidrules}{}
4 \newcommand*{\specialrule}[3]{}
5 \newcommand*{\cprotect}[1]{#1}

Package 10

lwarf-ccaption.sty

81 Ccaption

Pkg ccaption ccaption is not used. The user is recommended to use caption instead.

for HTML output: 1 \LWR@loadnever{ccaption}{caption}

Package 11

l warp-changepage.sty

82 Changepage

Pkg `changepage` `changepage` is ignored.

for HTML output: Discard all options for `l warp-changepage`:

```
1 \LWR@ProvidesPackageDrop{changepage}
2 \newif\ifoddpage
3 \DeclareRobustCommand{\checkoddpage}{\oddpage=true}
4 \DeclareRobustCommand{\changetext}[5]{}
5 \DeclareRobustCommand{\changepage}[9]{}
6 \newenvironment{adjustwidth}[2]{}{}
7 \newenvironment{adjustwidth*}[2]{}{}
```

Package 12

lwarf-cutwin.sty

83 Cutwin

Pkg **cutwin** Emulated.

for HTML output: Discard all options for lwarf-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 13

l warp-dcolumn.sty

84 Dcolumn

Pkg **dcolumn** **dcolumn** is emulated during HTML output, and the **dcolumn** package is ignored.

```
1 \LWR@ProvidesPackageDrop{dcolumn}
```

Package 14

lwarf-draftwatermark.sty

85 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 15

lwarf-ellipsis.sty

86 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 16

l warp-emptypage.sty

87 Emptypage

Pkg **emptypage** **emptypage** is ignored.

for HTML output: Discard all options for **l warp-emptypage**:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

Package 17

lwarf-endnotes.sty

88 Endnotes

(Based on original code by JOHN LAVAGNINO.)

Pkg endnotes

Discard all options for lwarf-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 18

l warp-enumerate.sty

89 Enumerate

Pkg **enumerate** `enumerate` is ignored. `enumitem` is then modified per the `shortlabels` option.

`enumerate` conflicts with `enumitem` if both are loaded at the same time, but `l warp` does not actually load `enumerate`. While generating HTML, `l warp` only loads `enumitem`, and `enumerate` is simulated by `enumitem` using the functionality of the `shortlabels` option.

A problem may occur during print output if `enumitem` is loaded, either manually or by some other package such as `siunitx`. If these are used, `enumerate` will conflict with `enumitem` during print output.

for HTML output: Discard all options for `l warp-enumerate`:

```
1 \LWR@ProvidesPackageDrop{enumerate}

2 % \DeclareOption{shortlabels}
3 %
4 \def\enit@shl#1{%
5   \ifnum\enit@type=\tw@
6     \enit@toks{\#1}%
7   \else
8     \def\enit@c{\#1}%
9     \enit@first#1,\@nil\@nil % Returns \enit@toks
10   \fi}
11 }
```

Package 19

lwarf-epigraph.sty

90 Epigraph

Pkg **epigraph** `epigraph` is emulated during HTML output, and the `epigraph` package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{epigraph}

2 \newcommand{\qitem}[2]
3 {
4 \begin{BlockClass}{qitem}
5 #1
6 \begin{BlockClass}{epigraphsource}
7 #2
8 \end{BlockClass}
9 \end{BlockClass}
10 }

11 \newcommand{\epigraph}[2]
12 {
13 \begin{BlockClass}{epigraph}
14 \qitem{#1}{#2}
15 \end{BlockClass}
16 }
17
18 \newenvironment*{epigraphs}
19 {\BlockClass{epigraph}}
20 {\endBlockClass}
```

Use CSS to format epigraphs.

The following are null commands for source compatibility:

```

21 \newlength{\epigraphwidth}
22 \setlength{\epigraphwidth}{.5\linewidth}
23 \newenvironment*{\flushepinormal}{}{}
24 \newcommand{\textflush}[1]{\flushepinormal}
25 \newcommand{\epigraphflush}[1]{\flushright}
26 \newcommand{\sourceflush}[1]{\flushright}
27 \newcommand*{\epigraphsize}{\small}
28 \newlength{\epigraphrule}
29 \newlength{\beforeepigraphskip}
30 \newlength{\afterepigraphskip}
31 \newcommand{\epigraphhead}[2][0]{#2}
```

```
32 \newcommand{\dropchapter}[1]{}
33 \newcommand*{\undodrop}{}
34 \newcommand{\cleartoevenpage}[1][]{}
```

Package 20

lwarf-eso-pic.sty

91 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 \let\AddToShipoutPicture\AddToShipoutPictureBG
14 \NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
15 \newcommand*{\ClearShipoutPictureBG}{}
16 \newcommand*{\ClearShipoutPicture}{}
17 \newcommand*{\ClearShipoutPictureFG}{}
18 \newcommand{\gridSetup}[6][]{}
```

Package 21

l warp-everypage.sty

92 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 22

l warp-extramarks.sty

93 Extramarks

Pkg extramarks extramarks is not used.

for HTML output: Discard all options for l warp-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 23

lwarf-fancyhdr.sty

94 Fancyhdr

Pkg fancyhdr fancyhdr is nullified.

for HTML output: Discard all options for lwarf-fancyhdr:

```
1 \LWR@ProvidesPackageDrop{fancyhdr}

2 \newcommand*{\fancyhead}[2] []
3 \newcommand*{\fancyfoot}[2] []
4 \newcommand*{\fancyhf}[2] []
5 \newcommand*{\fancypagestyle}[2] []
6 \newcommand*{\lhead}[1] {}
7 \newcommand*{\chead}[1] {}
8 \newcommand*{\rhead}[1] {}
9 \newcommand*{\lfoot}[1] {}
10 \newcommand*{\cfoot}[1] {}
11 \newcommand*{\rfoot}[1] {}
12 \newcommand*{\headrulewidth} {}
13 \newcommand*{\footrulewidth} {}
14 \newcommand*{\fancyheadoffset}[2] []
15 \newcommand*{\fancyfootoffset}[2] []
16 \newcommand*{\fancyhfoffset}[2] []
17 \newcommand*{\iffloatpage}[2]{#2}
18 \newcommand*{\ifftopfloat}[2]{#2}
19 \newcommand*{\iffbotfloat}[2]{#2}
```

Package 24

lwarf-float.sty

95 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 56.2 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 {%
5 \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}%
6 }%
7 {\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.

```
8 \cslet{listof#1s}\relax
9 \cslet{listof#1es}\relax
10 }
```

\floatname {\langle type\rangle} {\langle name\rangle}

Sets the text name of the float, such as “Figure”.

```
11 \NewDocumentCommand{\floatname}{m +m}{%
12 \SetupFloatingEnvironment{#1}{name=#2}%
13 }
```

\floatplacement {\langle type\rangle} {\langle placement\rangle}

Float placement is ignored.

```
14 \newcommand*{\floatplacement}[2]{%
15 \SetupFloatingEnvironment{#1}{placement=#2}%
16 }

\floatstyle {⟨style⟩}
```

Float styles are ignored.

```
17 \newcommand{\floatstyle}[1]{%
18 }
```

```
\restylefloat * {⟨style⟩}
```

Float styles are ignored.

```
19 \NewDocumentCommand{\restylefloat}{s m}{%
20 }
```

Package 25

l warp-floatfl t.sty

96 Floatfl t

Pkg **floatfl t** Emulated.

for HTML output: Discard all options for l warp-floatfl t:

```
1 \LWR@ProvidesPackageDrop{floatfl t}
```

Borrowed from the l warp version of keyfloat:

```
2 \DeclareDocumentEnvironment{KFLT@marginfloat}{O{-1.2ex} m}
3 {%
4 \LWR@maybeinthisfloat%
5 \LWR@forcenewpage%
6 \LWR@stoppars%
7 \LWR@htmlltag{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 {%
17 \begin{KFLT@marginfloat}{figure}%
18 \end{KFLT@marginfloat}%
19 \DeclareDocumentEnvironment{floatingtable}{o +m}
20 {%
21 \begin{KFLT@marginfloat}{table}#2%
22 \end{KFLT@marginfloat}%
23 }
```

Package 26

l warp-floatrow.sty

97 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. l warp 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 }

\floatingbox [\langle 1 preamble\rangle] [\langle 2 captype\rangle] [\langle 3 width\rangle] [\langle 4 height\rangle] [\langle 5 vert pos\rangle]
[\langle 6 caption\rangle] [\langle 7 object\rangle]
```

Only parameters for captype, width, caption, and object are used.

LWR@insubfloatrow is true if inside a **subfloatrow** environment.

There are two actions, depending on the use of `subcaption` or `subfig`.

```
9 \NewDocumentCommand{\floatbox}{o m o o o +m +m}{%
10 \ifbool{LWR@subcaptionloaded}{%
11 {%
12 \ifbool{LWR@insubfloatrow}{%
13 {%
14 \IfValueTF{#3}{%
15 {\@nameuse{sub#2}{#3}}{%
16 {\@nameuse{sub#2}{\linewidth}}{%
17 }{%
18 {%
19 \@nameuse{#2}{%
20 }{%
21 #6{%
22 {%
23 #7{%
```

For `subcaption`:

```
12 \ifbool{LWR@insubfloatrow}{%
13 {%
14 \ifbool{LWR@subcaption}{%
15 {%
16 {%
17 {%
18 {%
19 {%
20 {%
21 {%
22 {%
23 {%
```

`subfigure` and `subtable` environments take width as an argument.

```
14 \IfValueTF{#3}{%
15 {\@nameuse{sub#2}{#3}}{%
16 {\@nameuse{sub#2}{\linewidth}}{%
17 }{%
18 {%
19 {%
20 {%
21 {%
22 {%
23 {%
```

`figure` and `table` environments do not take a width argument.

```
19 \@nameuse{#2}{%
20 }{%
21 #6{%
22 {%
23 #7{%
```

End the environments:

```
24 \ifbool{LWR@insubfloatrow}{%
25 {\@nameuse{endsub#2}}{%
26 {\@nameuse{end#2}}{%
27 }{%
28 {%
29 {%
30 {%
```

For `subfig`:

```
29 \ifbool{LWR@insubfloatrow}{%
30 {%
31 {%
32 {%
```

`\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 \Cnameuse{#2}
38 #6
39
40 #7
41 \Cnameuse{end#2}
42 }% subfig package, but not a subfig
43 }% assume subfig
44 }

```

Not used:

```

45 \newcommand*\nocapbeside(){}
46 \newcommand*\capbeside(){}
47 \newcommand*\capttop(){}
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 \Cnamedef{#1}{%
59 \floatbox{#2}{%
60 }%
61 }

```

Preamble and default width are ignored.

```

62 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
63 \Cnamedef{#1}{%
64 \floatbox{#2}{%
65 }%
66 }

```

```
67 \newfloatcommand{ffigbox}{figure}{\nocapbeside}[]
```

```
68 \newfloatcommand{ttabbox}{table}[\caption][\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@insubfloatrow}
134 \boolfalse{LWR@insubfloatrow}
```

```
135 \newenvironment*{subfloatrow}[1][2]
136 {
```

The row of floats is placed into a <div> of class **floatrow**:

```
137 \LWR@forcenewpage
138 \BlockClass{floatrow}
```

While inside the floatrow, LWR@insubfloatrow is set true, which tells \floatbox to use \subfigure or \subtable.

```
139 \begingroup
140 \booltrue{LWR@insubfloatrow}
141 }
142 {
143 \endgroup
144 \endBlockClass
145 \boolfalse{LWR@insubfloatrow}
146 }
```

Package 27

l warp- fontenc.sty

98 Fontenc

Pkg **fontenc** Error if fontenc is loaded after l warp.

Discard all options for l warp- fontenc:

for HTML output: 1 \LWR@ProvidesPackageDrop{fontenc}
2 \LWR@loadbefore{fontenc}

Package 28

l warp- fonts spec.sty

99 Fonts spec

Pkg **fonts spec** Error if fonts spec is loaded after l warp.

Discard all options for l warp- fonts spec:

for HTML output: 1 \LWR@ProvidesPackageDrop{fonts spec}
2 \LWR@loadbefore{fonts spec}

Package 29

l warp- footmisc.sty

100 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\p0 \hrule \kern2.6\p0}
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     \c@xmpfootnotemark
22   {%
23     \stepcounter\c@mpfn
24     \protected@xdef\@thefnmark{\thempfn}%
25     \c@footnotemark
26   }%
27 }
28 \def\c@xmpfootnotemark[#1]{%
29   \begingroup
30   \csname c@\c@mpfn\endcsname #1\relax
31   \unrestored@protected@xdef\@thefnmark{\thempfn}%
32   \endgroup
33   \c@footnotemark
34 }
```

Package 30

l warp-footnote.sty

101 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 l warp
6 }
7
8 % \let\fn@endnote\color@endgroup% *** conflicts with l warp
9 \newcommand*{\fn@endnote}{\LWR@htmlltagc{/}\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@htmlltagc{/}\LWR@tagregularparagraph}%
18   \fn@postfntext%
19   \egroup%
20   \begingroup%
21   \let\@makefntext\@empty%
22   \let\@finalstrut\@gobble%
23   \let\@rule\@gobbletwo%
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 31

l warp-footnotehyper.sty

102 Footnotehyper

Pkg **footnotehyper** footnotehyper is a hyperref-safe version of footnote. For l warp, footnotehyper is emulated.

for HTML output: Discard all options for l warp-footnotehyper:

```
1 \RequirePackage{footnote}  
2 \LWR@ProvidesPackageDrop{footnotehyper}
```

Package 32

l warp-framed.sty

103 Framed

(Based on original code by DONALD ARSENEAU.)

Pkg **framed** **framed** is supported and patched by l warp.

for HTML output: Accept all options for l warp-framed:

```
1 \LWR@ProvidesPackagePass{framed}

2
3 \renewenvironment{framed}{%
4 \LWR@forcenewpage
5 \BlockClass{framed}%
6 }
7 {\endBlockClass}
8
9 \renewenvironment{o framed}{%
10 \LWR@forcenewpage
11 \BlockClass{framed}%
12 }
13 {\endBlockClass}
14
15
16 \renewenvironment{shaded}{%
17 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
18 \LWR@forcenewpage
19 \BlockClass{framed}[background: \#\LWR@tempcolor]%
20 }
21 {\endBlockClass}
22
23 \renewenvironment{shaded*}{%
24 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
25 \LWR@forcenewpage
26 \BlockClass{framed}[background: \#\LWR@tempcolor]%
27 }
28 {\endBlockClass}
29
30
31 \renewenvironment{leftbar}{%
32 \LWR@forcenewpage
33 \BlockClass{framedleftbar}
34 \def\FrameCommand{}%
```

```
35 \MakeFramed {}
36 }%
37 {\endMakeFramed\endBlockClass}
38
39
40 \renewenvironment{snugshade}{%
41 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
42 \LWR@forcenewpage
43 \BlockClass{snugframed}[background: \#\LWR@tempcolor]%
44 }
45 {\endBlockClass}
46
47 \renewenvironment{snugshade*}{%
48 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
49 \LWR@forcenewpage
50 \BlockClass{snugframed}[background: \#\LWR@tempcolor]%
51 }
52 {\endBlockClass}
53
54 \let\oframed\framed
55 \let\endoframed\endframed
56
57
58 \RenewEnviron{titled-frame}[1]{%
59 \CustomFBox{#1}{}{0pt}{0pt}{0pt}{0pt}{\BODY}
60 }

\CustomFBox {\langle toptitle\rangle} {\langle bottitle\rangle} {\langle thickness top\rangle} {\langle bottom\rangle} {\langle left\rangle} {\langle right\rangle}
{\langle text contents\rangle}

61 \renewcommand{\CustomFBox}[7]{%
62 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
63 \LWR@forcenewpage
64 \begin{BlockClass}{framed}[border: 3px solid \#\LWR@tempcolor]%
65 \ifthenelse{\isempty{#1}}{}{%
66 \begin{BlockClass}{framedtitle}[background: \#\LWR@tempcolor]%
67 \textcolor{TFTTitleColor}{\textbf{#1}}%
68 \end{BlockClass}
69 }% not empty
70
71 #7
72
73 \ifthenelse{\isempty{#2}}{}{%
74 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
75 \begin{BlockClass}{framedtitle}[background: \#\LWR@tempcolor]%
76 \textcolor{TFTTitleColor}{\textbf{#2}}%
77 \end{BlockClass}
78 }% not empty
79 \end{BlockClass}
```

```
80 }

\TitleBarFrame [⟨marker⟩] {⟨title⟩} {⟨contents⟩}

81 \renewcommand\TitleBarFrame[3] []{
82 \CustomFBox
83   {#2}{}}%
84   \fboxrule\fboxrule\fboxrule\fboxrule
85   {#3}%
86 }

87 \renewcommand{\TF@Title}[1]{#1}

MakeFramed {⟨settings⟩}

88 \let\MakeFramed\relax
89 \let\endMakeFramed\relax
90
91 \NewEnviron{MakeFramed}[1]{%
92 \FrameCommand{\begin{minipage}{\linewidth}\BODY\end{minipage}}%
93 }

\fbox@put@frame {⟨frame cmd no split⟩} {⟨frame cmd split⟩}

94 \renewcommand*{\fbox@put@frame}[2]{%
95 \relax%
96 \tempboxa%
97 }
```

Package 33

l warp-ftnright.sty

104 Ftnright

Pkg **ftnright** ftnright is ignored.

for HTML output: Discard all options for l warp-ftnright:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

Package 34

l warp-geometry.sty

105 Geometry

Pkg **geometry** geometry is preloaded by l warp, but must be nullified as seen by the user's source code.

for HTML output: Discard all options for l warp-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 35

lwarf-glossaries.sty

106 Glossaries

Pkg **glossaries** **xindy** is required for glossaries.

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

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

lwarfmk has the commands **printglossary** and **htmlglossary** to process the glossaries created by **glossaries** using **xindy**.

Opt **lwarf-newproject** The packages **lwarf-newproject** and **lwarf** each take an option **lwarfmklang=english** to set the language used by **xindy**. Set the same option for both packages. This is passed to **xindy** using its **-L** option.
Opt **lwarf lwarfmklang**

for HTML output:

```
1 \PassOptionsToPackage{xindy}{glossaries}
2 \LWR@ProvidesPackagePass{glossaries}
3 \setupglossaries{nonumberlist}
4 \setglossarystyle{index}
```

Package 36

lwarf-graphics.sty

107 Graphics

Pkg `graphics` `graphics` is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphics}

Package 37

lwarf-graphicx.sty

108 Graphicx

Pkg `graphicx` `graphicx` is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphicx}

Package 38

lwarf-hyperref.sty

109 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 lwarf html version of package 'hyperref' -- discarding options.}
3 \typeout{    Are not using ProvidesPackage, so that other packages}
4 \typeout{    do not attempt to patch lwarf's version of 'hyperref'.}
5 \% \ProvidesPackage{lwarf-\#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@htmntag{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@htmntag{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}{0{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 39

l warp-indentfirst.sty

110 Indentfirst

Pkg indentfirst indentfirst is ignored.

Discard all options for l warp-indentfirst:

for HTML output: 1 \LWR@ProvidesPackageDrop{indentfirst}

Package 40

l warp-inputenc.sty

111 Inputenc

Pkg inputenc Error if inputenc is loaded after l warp.

Discard all options for l warp-inputenc:

for HTML output: 1 \LWR@ProvidesPackageDrop{inputenc}

2 \LWR@loadbefore{inputenc}

Package 41

lwarf-keyfloat.sty

112 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}{O{-1.2ex} m}
15 {%
16 \LWR@maybeinthisfloat%
17 \LWR@forcenewpage
18 \LWR@stoppars%
19 \LWR@htmntag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
20 \LWR@startpars%
21 \captionsetup{type=#2}%
22 }
23 {
24 \LWR@htmldivclassend{div}
25 }

26 }% AtBeginDocument
```

Package 42

l warp-layout.sty

113 Layout

Pkg **layout** layout is ignored.

for HTML output: Discard all options for l warp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}
2 \NewDocumentCommand{\layout}{s}{}{}
```

Package 43

l warp-letterspace.sty

114 Letterspace

Pkg **letterspace** letterspace is a subset of microtype, which is pre-loaded by l warp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for l warp-letterspace:

```
1 \LWR@ProvidesPackageDrop{letterspace}
2 \newcommand*\lsstyle{}
3 \newcommand\textls[2][]{}
4 \def\textls#1{}
5 \newcommand*\lslig[1]{#1}
```

Package 44

lwarf-lettrine.sty

115 Lettrine

(Based on original code by DANIEL FLIPO.)

Pkg **lettrine** Emulated.

for HTML output: Discard all options for lwarf-lettrine:

```
1 \LWR@ProvidesPackageDrop{lettrine}
```

The initial letter is in a `` of class `lettrine`, and the following text is in a `` of class `lettrinetext`. `\lettrine [<keys>] {<letter>} {<additional text>}`

```
2 \DeclareDocumentCommand{\lettrine}{o m m}{%
3   \InlineClass{lettrine}{#2}\InlineClass{lettrinetext}{#3} %
4 }
5
6 \newcounter{DefaultLines}
7 \setcounter{DefaultLines}{2}
8 \newcounter{DefaultDepth}
9 \newcommand*{\DefaultOptionsFile}{\relax}
10 \newcommand*{\DefaultLoversize}{0}
11 \newcommand*{\DefaultLraise}{0}
12 \newcommand*{\DefaultLhang}{0}
13 \newdimen\DefaultFindent
14 \setlength{\DefaultFindent}{\z@}
15 \newdimen\DefaultNindent
16 \setlength{\DefaultNindent}{0.5em}
17 \newdimen\DefaultSlope
18 \setlength{\DefaultSlope}{\z@}
19 \newdimen\DiscardVskip
20 \setlength{\DiscardVskip}{0.2\p@}
21 \newif\ifLettrineImage
22 \newif\ifLettrineOnGrid
23 \newif\ifLettrineRealHeight
24
25 \newcommand*{\LettrineTextFont}{\scshape}
26
27 \newcommand*{\LettrineFontHook}{}
28
29 \newcommand*{\LettrineFont}[1]{\InlineClass{lettrine}{#1}}
30 \newcommand*{\LettrineFontEPS}[1]{\includegraphics[height=1.5ex]{#1}}
```

Package 45

lwarf-lips.sty

116 Lips

Pkg lips lips is emulated during HTML output, and the lips package is ignored.

```
1 % \LWR@ProvidesPackageDrop{lips}
2 \PackageInfo{lwarf}{Using the lwarf version of package 'lips'.}%
3 \ProvidesPackage{lwarf-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 46

lwarf-listings.sty

117 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@origlst@EveryPar\lst@EveryPar
7
8 \renewcommand{\lstlisting}[2]{\hypertocfloat{1}{\lstlisting}{\liststyle}{#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{\liststyle}%
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\lst@EveryPar\relax%
17 \LWR@forcenewpage
18 \LWR@atbeginverbatim{programlisting}%
19
20 \let\lst@EveryPar\LWR@origlst@EveryPar%
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 #1%
41 \ifx\lst@@caption\empty\expandafter\lst@HRefStepCounter \else
42 \expandafter\refstepcounter
43 \fi {lstlisting}%
44 \LWR@traceinfo{About to assign label: !\lst@label!}%
45 \ifx\lst@label\empty\else
46 \label{\lst@label}\fi
47 \LWR@traceinfo{Finished assigning the label.}%
48 \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
49 \global\let\lst@name\lst@arg \global\let\lst@name\lst@name
50 \lst@ifnolol\else
51 \ifx\lst@@caption\empty
```

```

52           \ifx\lst@caption\@empty
53             \ifx\lst@intname\@empty \else \def\lst@temp{ }%
54               \ifx\lst@intname\lst@temp \else

```

This code places a contents entry for a non-float. This would have to be modified for `lwarp`:

```

55 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
56 %
57           \addcontentsline{lol}{lstlisting}{\lst@name}
58           \fi\fi
59           \fi
      \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           {\begin{group}
71 \LWR@traceinfo{at the selected position}%

```

These space and box commands are not needed for HTML output:

```

72 %
73           \let\@vskip\vskip
74           \def\vskip{\afterassignment\lst@vskip \tempskipa}%
75           \def\lst@vskip{\nobreak\@vskip\tempskipa\nobreak}%
76           \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
77           \ifx #1\allowbreak \fi
      \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 %
81           \lst@maketitle\lst@title % (AS)
82 \LWR@traceinfo{Making title: \lst@title}%
83 \begin{BlockClass}{lstlistingtitle}%
84 \lst@maketitle\lst@title% lwarp
85 \end{BlockClass}%

```

```

85           \fi
86 \LWR@traceinfo{About to assign label: !\lst@label!}%
87         \ifx\lst@label\empty\else
88 \leavevmode% gets rid of bad space factor error
89 \GetTitleStringExpand{\lst@caption}%
90 \edef\LWR@lntemp{\GetTitleStringResult}%
91 \edef@\currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
92 \label{\lst@label}\fi
93 \LWR@traceinfo{Finished assigning the label.}%

```

Not needed for lwarp:

```

94 %           \ifx #1b\allowbreak \fi
95           \endgroup}{}%
96       \fi
97 \LWR@traceinfo{end of making a listings display caption}%
98   \else
99 \LWR@traceinfo{INLINE}%
100  \fi
101 \LWR@traceinfo{DONE WITH CAPTION at #1}%
102 }

```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```

103 \lst@Key{numbers}{none}{%
104   \let\lst@PlaceNumber\empty
105   \lstKV@SwitchCases{#1}%
106   {none&\\%
107     left&\def\lst@PlaceNumber{%
108   \% \llap{%
109 \LWR@orignormalfont%
110 \lst@numberstyle{\the\lstnumber}\kern\lst@numbersep%
111 }%
112 }%
113 \\%
114   right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont%
115           \kern\VerbatimHTMLWidth \kern\lst@numbersep%
116           \lst@numberstyle{\the\lstnumber}}}%
117 }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}%
118 \end{warpHTML}

```

Package 47

lwarf-longtable.sty

118 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** * [*horizontalignment*] {*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{LTcaptiontype}%
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{\LTcaptype}\%
17 \caption@setoptions{longtable}\%
18 \caption@setoptions{@longtable}\%
19 \caption@LT@setup\%
20 \refstepcounter{\LTcaptype}\%
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}%
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 48

l warp-l scape.sty

119 L scape

Pkg l scape l scape is nullified.

for HTML output: Discard all options for l warp-l scape.

```
1 \LWR@ProvidesPackageDrop{l scape}  
2 \newenvironment*{landscape}{}{}
```

Package 49

l warp-l tcaption.sty

120 L tcaption

Pkg l tcaption l tcaption is emulated during HTML output, and the l tcaption package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{l tcaption}

\LTcaptype is already defined by l warp.

longtable* is already defined by l warp-longtable.

```
2 \newlength{\LTcapskip}  
3 \newlength{\LTcapleft}  
4 \newlength{\LTcapright}  
5 \newcommand*{\LTcapmarginsfalse}{}{}
```

Package 50

l warp-marginfix.sty

121 Marginfix

Pkg **marginfix** Not used.

for HTML output: Discard all options for l warp-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*{\unlockmargin}[1][]{}
11 \newcommand*{\marginphantom}[2][]{}
```

Package 51

l warp-marginnote.sty

122 Marginnote

Pkg **marginnote** Emulated.

for HTML output: Discard all options for l warp-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 52

l warp-mcaption.sty

123 Mcaption

Pkg **mcaption** mcaption is nullified.

for HTML output: Discard all options for l warp-mcaption:

```
1 \LWR@ProvidesPackageDrop{mcaption}

2 \newenvironment{margincap}{}{}
3 \newcommand*{\margincapalign}={}
4 \newlength{\margincapskip}
```

Package 53

l warp-mdframed.sty

124 Mdframed

Pkg **mdframed** **mdframed** is loaded with options forced to **framemethod=none**.

for **HTML output**: 1 \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, **l warp** 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 **l warp** 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**

2 \LWR@origRequirePackage{amsthm}

Do not require Tikz or pstricks:

```
3 \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.

```
4 \mdfsetup{
5 startcode={\LWR@mdframedstart\LWR@origraggedright},
6 endcode={\LWR@mdframedend},
7 startinnercode={\LWR@startpars\LWR@origraggedright},
8 endinnercode={\LWR@stoppars},
9 }
```

Given the `mdframed` key, print the color.

```
10 \newcommand*{\LWR@mdfprintcolor}[1]{%
11 \convertcolorspec{named}{\csuse{mdf@\#1}}{HTML}\LWR@tempcolor%
12 \#\LWR@tempcolor
13 }
```

Given the `mdframed` key, print the length.

```
14 \newcommand*{\LWR@mdfprintlength}[1]{%
15 \rndprintlength{\csuse{mdf@\#1@length}}
16 }
```

Actions before an mdframe starts.

Encapsulate a frame inside a `<div>` of the desired `class`.

```
17 \newcommand*{\LWR@mdframedstart}{%
```

Turn off paragraph handling during the generation of the encapsulating tags:

```
18 \LWR@stoppars%
```

Below, print HTML pt units:

```
19 \uselengthunit{PT}%
```

Open a `<div>` and with custom `class` and custom `style`:

```
20 \LWR@htmntagc{div class="\LWR@mdthisenv" \LWR@orignewline
21 style=" \LWR@orignewline
```

Convert and print the background color:

```
22 background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```
23 border: \LWR@mdfprintlength{linewidth} solid
24 \LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
25 border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
26 \ifbool{mdf@shadow}{%
27 box-shadow:
28 \LWR@mdfprintlength{shadowsize}
29 \LWR@mdfprintlength{shadowsize}
30 \LWR@mdfprintlength{shadowsize}
31 \LWR@mdfprintcolor{shadowcolor} ;
32 }
33 {box-shadow: none ;}
34 \LWR@orignewline

35 "}
36 % \LWR@htmldivclass{\LWR@mdthisenv}
```

mdframed environment may not work with the modified `\hspace` and `\rule`, so restore them to their originals while inside **mdframed**:

```
37 \let\hspace\LWR@orighspace%
38 \let\rule\LWR@origrule%
39 }
```

Actions after an mdframe ends.

After closing the `<div>`, globally restore to the default environment type:

```
40 \newcommand*{\LWR@mdframedend}{
```

Close the custom `<div>`:

```
41 \LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
42 \gdef\LWR@mdthisenv{mdframed}
```

Resume paragraph handling:

```
43 \LWR@startpars%
44 }
```

Encapsulation of the original which places the title inside a `` of class `mdframedtitle`:

```
45 \let\LWR@origmdfframedtitleenv\mdfframedtitleenv
46
47 \newlength{\LWR@titleroundcorner}
48
49 \renewrobustcmd\mdfframedtitleenv[1]{%
50 \LWR@origmdfframedtitleenv{%
```

Below, print HTML pt lengths:

```
51 \uselengthunit{PT}{%
```

Open a `` with a custom `class` and custom `style`:

```
52 \LWR@htmntagc{span class="mdframedtitle" \LWR@orignewline
53 style=" \LWR@orignewline
```

Convert and print the title background color:

```
54 background:
55 \LWR@mdfprintcolor{frametitlebackgroundcolor}
56 ; \LWR@orignewline
```

Convert and print the title rule:

```
57 \ifbool{mdf@frametitlerule}{%
58 border-bottom:
59 \LWR@mdfprintlength{frametitlerulewidth}
60 solid
61 \LWR@mdfprintcolor{frametitlerulecolor}
62 ; \LWR@orignewline
63 }{}
```

The title's top border radius is adjusted for the line width:

```
64 border-radius:
65 \setlength{\LWR@titleroundcorner}
66 {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{0pt}}
67 \rndprintlength{\LWR@titleroundcorner}
68 \rndprintlength{\LWR@titleroundcorner}
69 Opt Opt
70 \LWR@orignewline
```

Finish the custom style and the opening span tag:

```
71 " \LWR@orignewline
72 }% span
```

Restrict paragraph tags inside a span:

```
73 \begin{LWR@nestspan}%
```

Print the title inside the span:

```
74 #1%
```

Cloosee the span and unnest the paragraph tag restriction:

```
75 \LWR@htmlltagc{/span}%
76 \end{LWR@nestspan}%
77 }
78 }
```

Common code for `\LWR@mdfsubtitle` and `\LWR@mdfsubsubtitle`.

Encapsulate the subtitle inside a `` of class `mdframedsubtitle`:

```
79 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}%
80 {%
81   the following empty line is required
82 }
```

Special handling for mdframed: Subtitles have `\pars` around them, so temporarily disable them here.

```
82 \let\par\LWR@origpar%
```

Open a `` with a custom `class` and custom `style`:

```
83 \LWR@htmlltagc{span class="mdframed#1title"
84 style=" \LWR@orignewline
```

Convert and print the background color:

```
85 background:
86 \LWR@mdfprintcolor{#1titlebackgroundcolor}
87 ; \LWR@orignewline
```

Convert and print the above line:

```
88 \ifbool{mdf@#1titleaboveline}{%
89 border-top:
90 \LWR@mdfprintlength{#1titleabovelinewidth}
91 solid
92 \LWR@mdfprintcolor{#1titleabovelinecolor}
93 ; \LWR@orignewline
94 }{}}
```

Convert and print the below line:

```

95 \ifbool{mdf@#1titlebelowline}{%
96 border-bottom:
97 \LWR@mdfprintlength{#1titlebelowlinewidth}
98 solid
99 \LWR@mdfprintcolor{#1titlebelowlinecolor}
100 ; \LWR@orignewline
101 }{}%
```

Finish the custom style and the opening span tag:

```
102 "}% span
```

Restrict paragraph tags inside a span:

```
103 \begin{LWR@nestspan}%
```

Perform the original subtitle action:

```

104 \IfNoValueTF{#2}
105 {\csuse{LWR@origmdf#1title}{#3}}%
106 {\csuse{LWR@origmdf#1title}[#2]{#3}}%
```

Close the span and unnest the paragraph tag restriction:

```

107 \LWR@htmntagc{/span}%
the following empty line is required
108 \end{LWR@nestspan}%
must follow the /span or an extra <p> appears
109
110 }

111 \let\LWR@origmdfsubtitle\mdfsubtitle
112
113 \newcommand*\{\LWR@mdfsubtitle}{%
114 \LWR@mdfsubtitlecommon{sub}%
115 }
116 \let\mdfsubtitle\LWR@mdfsubtitle

117 \let\LWR@origmdfsubsubtitle\mdfsubsubtitle
118
119 \newcommand*\{\LWR@mdfsubsubtitle}{%
120 \LWR@mdfsubtitlecommon{subsub}%
121 }
122 \let\mdfsubsubtitle\LWR@mdfsubsubtitle
```

Stores the environment of the frame about to be created:

```
123 \newcommand*\{\LWR@mdthisenv}{mdframed}
```

Modified from the original to remember the environment.

```

124 \renewrobustcmd*\newmdenv[2] [] {%
125 \newenvironment{#2}%
126 {%
127 \mdfsetup{#1}%
128 \renewcommand*{\LWR@mdthisenv}{md#2}%
129 \begin{mdframed}%
130 }%
131 {\end{mdframed}}%
132 }

```

Modified from the original to remember the environment.

```

133 \renewrobustcmd*\surroundwithmdframed}[2] [] {%
134 \BeforeBeginEnvironment{#2}%
135 \renewcommand*{\LWR@mdthisenv}{md#2}%
136 \begin{mdframed}[#1]}%
137 \AfterEndEnvironment{#2}{\end{mdframed}}%
138 }

```

[*numberedlike*] {[*caption*] [*within*]}

Modified from the original to remember the environment.

```

139 \let\LWR@origmdtheorem\mdtheorem
140
141 \DeclareDocumentCommand{\LWR@mdtheorem}{O{} m o m o}{%
142 \LWR@origmdtheorem[#1]{#2}[#3]{#4}[#5]%
143 \BeforeBeginEnvironment{#2}{\renewcommand*{\LWR@mdthisenv}{md#2}}%
144 }%
145
146 \let\mdtheorem\LWR@mdtheorem

```

[*numberedlike*] {[*caption*] [*within*]}

Modified from the original to remember the environment.

```

147 \DeclareDocumentCommand\newmdtheoremenv{O{} m o m o }{%
148 \ifboolexpr{ test {\IfNoValueTF{#3}} and test {\IfNoValueTF{#5}} }{%
149   {\newtheorem{#2}{#4}}{%
150     \IfValueTF{#3}{\newtheorem{#2}{#3}{#4}}{%
151       \IfValueTF{#5}{\newtheorem{#2}{#4}{#5}}{%
152         }%
153 \BeforeBeginEnvironment{#2}{%
154 \renewcommand*{\LWR@mdthisenv}{md#2}%
155 \begin{mdframed}[#1]}%
156 \AfterEndEnvironment{#2}{%
157 \end{mdframed}}%
158 }

```

Package 54

lwarf-microtype.sty

125 Microtype

Pkg **microtype** microtype is pre-loaded by lwarf. All user options and macros are ignored and disabled.

for **HTML output**: Discard all options for lwarf-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 55

l warp-mparhack.sty

126 Mparhack

Pkg mparhack Not used.

for HTML output: Discard all options for l warp-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```

Package 56

l warp-multicol.sty

127 Multicol

Pkg `multicol` `multicol` is emulated during HTML output, and the `multicol` package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{multicol}[2015/09/13]
```

Multicols are converted into a 1–3 column display, browser-supported.

The optional multicols heading is placed inside a `<div>` of class `multicolsheading`.

The content is placed inside a `<div>` of class `multicols`.

```
2 \begin{warpHTML}
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}{multicolsheading}#3\end{BlockClass}}{}}
```

When done with the environment, close the div:

```
8 {\endBlockClass}
```

Emulated null functions which are not used in HTML:

```
9 \newcommand*{\columnbreak}{}%
10 \newcommand*{\RLmulticolcolumns}{}%
11 \newcommand*{\LRmulticolcolumns}{}%
12
13 \newlength{\premulticols}
14 \newlength{\postmulticols}
15 \newlength{\multicolsep}
16 \newlength{\multicolbaselineskip}
17 \newlength{\multicoltolerance}
```

```
18 \newlength{\multicolpretolerance}
19 \newcommand*{\columnseprulecolor}{\normalcolor}
20 \newcounter{columnbadness}
21 \newcounter{finalcolumnbadness}
22 \newcounter{collectmore}
23 \newcounter{unbalance}
24 \newlength{\multicolovershoot}
25 \newlength{\multicolundershoot}

26 \end{warpHTML}
```

Package 57

l warp-multirow.sty

128 Multirow

Pkg multirow multirow is emulated during HTML output, and the multirow package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{multirow}

Package 58

l warp-nameref.sty

129 Nameref

Pkg nameref nameref is emulated by l warp.

for HTML output: Discard all options for l warp-nameref:

```
1 \typeout{Using the l warp html version of package 'nameref' -- discarding options.}
2 \typeout{    Are not using ProvidesPackage, so that other packages}
3 \typeout{    do not attempt to patch l warp's version of 'nameref'.}
4 \DeclareOption*{}
5 \ProcessOptions\relax
```

Package 59

l warp-needspace.sty

130 Needspace

Pkg **needspace** needspace is not used during HTML conversion.

for HTML output: Discard all options for l warp-needspace:

```
1 \LWR@ProvidesPackageDrop{needspace}
2
3 \newcommand*{\needspace}[1]{}
4 \DeclareDocumentCommand{\Needspace}{s m}{}  
}
```

Package 60

l warp-newclude.sty

131 Newclude

Pkg **newclude** Error if newclude is loaded after l warp.

Discard all options for l warp-newclude:

for HTML output: 1 \LWR@ProvidesPackageDrop{newclude}
2 \LWR@loadbefore{newclude}

Package 61

l warp-newunicodechar.sty

132 Newunicodechar

Pkg **newunicodechar** Error if `newunicodechar` is loaded after `l warp`.

Discard all options for `l warp-newunicodechar`:

for HTML output:

- 1 \LWR@ProvidesPackageDrop{newunicodechar}
- 2 \LWR@loadbefore{newunicodechar}

Package 62

l warp-nextpage.sty

133 Nextpage

Pkg **nextpage** `nextpage` is nullified.

for HTML output: Discard all options for `l warp-nextpage`.

- 1 \LWR@ProvidesPackageDrop{nextpage}
- 2 \newcommand{\cleartoevenpage}[1] {}
- 3 \newcommand{\movetoevenpage}[1] {}
- 4 \newcommand{\cleartooddpage}[1] {}
- 5 \newcommand{\movetooddpage}[1] {}

Package 63

l warp-nowidow.sty

134 Nowidow

Pkg nowidow nowidow is not used during HTML conversion.

Discard all options for l warp-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 64

lwarf-ntheorem.sty

135 Ntheorem

(Based on original code by WOLFGANG MAY, ANDREAS SCHEDLER.)

Pkg **ntheorem** ntheorem is patched for use by lwarf.

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 AMS environments when the option **thref** is used. lwarf does not share this bug, so equations with \split, etc, are numbered correctly with lwarf'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 lwarf 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@ntheoremmarks}
9 \newif\ifsetendmark\setendmarktrue
10 }
11
12
13 \newbool{LWR@ntheoremamsthm}
14 \boolfalse{LWR@ntheoremamsthm}
15
16 \DeclareOption{amsthm}{\booltrue{LWR@ntheoremamsthm}}
17
18
19 \DeclareOption{amsmath}={}
20 \DeclareOption{hyperref}={}
21
22
23 \LWR@ProvidesPackagePass{ntheorem}

```

Storage for the style being used for new theorems.

```
24 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```

25 \gdef\theoremstyle#1{%
26   \@ifundefined{th@#1}{\@warning
27     {Unknown theoremstyle '#1'. Using 'plain'}%
28     \theorem@style{plain}
29     \renewcommand{\LWR@newtheoremstyle}{plain}}% new
30 }%
31   {
32 \theorem@style{#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\xnithm#1#2[#3]{%
38   \ifthm@tempif
39     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}}% new
40   \expandafter\ifundefined{c@#1}%
41     {\@definecounter{#1}}{}%
42   \newctr{#1}[#3]%
43   \expandafter\xdef\csname the#1\endcsname{%
44     \expandafter\noexpand\csname the#3\endcsname \thmcOUNTERsep
45     {\noexpand\csname\the\theoremnumbering\endcsname{#1}}}%
46   \expandafter\gdef\csname mkheader@#1\endcsname
```

```

47      {\csname setparms@\#1\endcsname
48      \@thm{\#1}{\#1}{\#2}
49 }%
50      \global\@namedef{end#1}{\@endtheorem}
51      \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
52  \fi
53 }
54
55 \gdef\@ynthm#1#2{%
56  \ifthm@tempif
57    \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
58    \expandafter\@ifundefined{c@#1}%
59      {\@definecounter{#1}}{%
60        \expandafter\xdef\csname the#1\endcsname
61          {\noexpand\csname the\theoremnumbering\endcsname{#1}}%
62        \expandafter\gdef\csname mkheader@#1\endcsname
63          {\csname setparms@\#1\endcsname
64            \@thm{\#1}{\#1}{\#2}}
65 }%
66    \global\@namedef{end#1}{\@endtheorem}
67    \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
68  \fi
69 }
70
71 \gdef\@othm#1[#2]#3{%
72  \@ifundefined{c@#2}{\@nocounterr{#2}}{%
73    \ifthm@tempif
74      \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
75      \global\@namedef{the#1}{\@nameuse{the#2}}%
76      \expandafter\protected@xdef\csname num@addtheoremline#1\endcsname{%
77        \noexpand\@num@addtheoremline{#1}{#3}}%
78      \expandafter\protected@xdef\csname nonum@addtheoremline#1\endcsname{%
79        \noexpand\@nonum@addtheoremline{#1}{#3}}%
80    \theoremkeyword{#3}%
81    \expandafter\protected@xdef\csname #1Keyword\endcsname
82      {\the\theoremkeyword}%
83    \expandafter\gdef\csname mkheader@#1\endcsname
84      {\csname setparms@\#1\endcsname
85        \@thm{\#1}{\#2}{#3}}
86 }%
87    \global\@namedef{end#1}{\@endtheorem}
88    \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\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@htmlltag{a id="autofloat-\arabic{LWR@thisfloat}"{} }\LWR@htmlltag{/a}%
95 \LWR@startpars%
96 }

```

The following are patched for CSS.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc. They are gathered together here.

Each theorem is encased in a `BlockClass` environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader<style>`.

```

97 \gdef\newtheoremstyle#1#2#3{%
98   \expandafter\@ifundefined{th@#1}%
99   { \expandafter\gdef\csname th@#1\endcsname{%
100     \def\@begintheorem####1####2{%
101       \LWR@forcenewpage% new
102     \BlockClass{theorembody#1}\LWR@thisthmstyle% new
103     \LWR@inctheorem% new
104     #2}%
105     \def\@opargbegintheorem####1####2####3{%
106       \LWR@forcenewpage% new
107     \BlockClass{theorembody#1}\LWR@thisthmstyle% new
108     \LWR@inctheorem% new
109     #3}%
110   }%
111 }%
112 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
113 }%
114 %
115 \renewtheoremstyle{plain}{%
116   {\item[\hspace{-0.2em}\hskip\labelsep \theorem@headerfont
117 \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
118   {\item[\hspace{-0.2em}\hskip\labelsep \theorem@headerfont
119 \InlineClass{theoremheaderplain}{##1\ ##2\ (###3)\theorem@separator}}}%
120 %
121 \renewtheoremstyle{break}{%
122   {\item[%
123 % \rlap{\vbox{\hbox{
124 \hspace{-0.2em}\hskip\labelsep \theorem@headerfont
125 \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
126 % }\hbox{\strut}}}
127 ]}%
128   {\item[%
129 % \rlap{\vbox{\hbox{
130 \hspace{-0.2em}\hskip\labelsep \theorem@headerfont
131 \InlineClass{theoremheaderbreak}{##1\ ##2\ (###3)\theorem@separator}\newline

```

```
132 % } \hbox{\strut}}}
133 ]}
134
135 \renewtheoremstyle{change}%
136   {\item[\hspace{\labelsep}
137 \theorem@headerfont
138 \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]}%
139   {\item[\hspace{\labelsep}
140 \theorem@headerfont
141 \InlineClass{theoremheaderchange}{##2\ ##1\ (#3)\theorem@separator}]}%
142
143 \renewtheoremstyle{changebreak}%
144   {\item[
145 % \rlap{\vbox{\hbox{
146 \hspace{\labelsep} \theorem@headerfont
147 \InlineClass{theoremheaderchangebreak}{##2\ ##1\theorem@separator}\newline
148 % } \hbox{\strut}}}
149 ]}%
150   {\item[
151 % \rlap{\vbox{\hbox{
152 \hspace{\labelsep} \theorem@headerfont
153 \InlineClass{theoremheaderchangebreak}{##2\ ##1\ (#3)\theorem@separator}\newline
154 % } \hbox{\strut}}}
155 ]}%
156
157 \renewtheoremstyle{margin}%
158   {\item[\hspace{\labelsep}\theorem@headerfont
159 \InlineClass{theoremheadermargin}{##2 \quad ##1\theorem@separator}
160 ]}%
161   {\item[\hspace{\labelsep}\theorem@headerfont
162 \InlineClass{theoremheadermargin}{##2 \quad ##1\ (#3)\theorem@separator}
163 ]}%
164
165 \renewtheoremstyle{marginbreak}%
166   {\item[\hspace{\labelsep}\theorem@headerfont
167 \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\theorem@separator}\newline
168 ]}%
169   {\item[\hspace{\labelsep}\theorem@headerfont
170 \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\ (#3)\theorem@separator}\newline
171 ]}%
172
173 \renewtheoremstyle{nonumberplain}%
174   {\item[\theorem@headerfont\hspace{\labelsep}
175 \InlineClass{theoremheaderplain}{##1\theorem@separator}]}%
176   {\item[\theorem@headerfont\hspace{\labelsep}
177 \InlineClass{theoremheaderplain}{##1\ (#3)\theorem@separator}]}%
178
179 \renewtheoremstyle{nonumberbreak}%
180   {\item[
181 % \rlap{\vbox{\hbox{
```

```

182 \hskip\labelsep \theorem@headerfont
183 \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline
184 % } \hbox{\strut}}}
185 ]}%
186 {\item[
187 % \rlap{\vbox{\hbox{
188 \hskip\labelsep \theorem@headerfont
189 \InlineClass{theoremheaderbreak}{##1\ (###3)\theorem@separator}\newline
190 % } \hbox{\strut}}}}
191 ]}%
192
193 \renewtheoremstyle{empty}%
194 {\item[]}%
195 {\item[\theorem@headerfont \hskip\labelsep\relax
196 \InlineClass{theoremheaderplain}{##3}]}%
197
198 \renewtheoremstyle{emptybreak}%
199 {\item[]}%
200 {\item[\theorem@headerfont \hskip\labelsep\relax
201 \InlineClass{theoremheaderplain}{##3}] \ \newline}

```

The following manually adjust the css for the standard configuration objects which are not a purely plain style:

```

202 \ifbool{LWR@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{\ensuremath{\_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{\ensuremath{\blacksquare}}
234 \renewtheorem{Proof}{Proof}
235 \renewtheorem{proof}{Proof}
236 \renewtheorem{Beweis}{Beweis}
237 \renewtheorem{beweis}{Beweis}
238 \qedsymbol{\ensuremath{\blacksquare}}
239
240 \theoremsymbol{}
241 }% not amsthm

```

Only if the `amsthm` option was given:

```

242 \ifbool{LWR@ntheoremamsthm}{
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@inctheorem% new
250     \item[\hspace{\labelsep}
251       \theorem@headerfont
252       \InlineClass{theoremheaderplain}{##1\ ##2.}%
253     ]}%
254   \def\@opargbegingroup{%
255     \LWR@forcenewpage% new
256     \BlockClass{theorembodyplain}% new
257     \LWR@inctheorem% new
258     \item[\hspace{\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@inctheorem% new
269     \item[\hspace{\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@inctheorem% new
277 \item[\hspace{\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@inctheorem% new
288 \item[\hspace{\labelsep}
289 % \theorem@headerfont
290 \InlineClass{theoremheaderdefinition}{##1\ ##2.}
291 ]}%
292 \def\@opargbegintheorem##1##2##3{%
293 \LWR@forcenewpage% new
294 \BlockClass{theorembodydefinition}% new
295 \LWR@inctheorem% new
296 \item[\hspace{\labelsep}
297 % \theorem@headerfont
298 \InlineClass{theoremheaderdefinition}{##1\ ##2\ (###3).}
299 ]}%
300
301 \gdef\th@nonumberdefinition{%
302 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
303 \def\@begintheorem##1##2{%
304 \LWR@forcenewpage% new
305 \BlockClass{theorembodydefinition}% new
306 \LWR@inctheorem% new
307 \item[\hspace{\labelsep}
308 % \theorem@headerfont
309 \InlineClass{theoremheaderdefinition}{##1.}
310 ]}%
311 \def\@opargbegintheorem##1##2##3{%
312 \LWR@forcenewpage% new
313 \BlockClass{theorembodydefinition}% new
314 \LWR@inctheorem% new
315 \item[\hspace{\labelsep}
316 % \theorem@headerfont
317 \InlineClass{theoremheaderdefinition}{##1\ (###3).}
318 ]}%
319
320 \gdef\th@remark{%
```

```
321 \def\theorem@headerfont{\itshape}\normalfont%
322 \def\@begintheorem##1##2{%
323 \LWR@forcenewpage% new
324 \BlockClass{theorembodyremark}% new
325 \LWR@inctheorem% new
326 \item[\hspace{\labelsep}
327 % \theorem@headerfont
328 \InlineClass{theoremheaderremark}{##1\ ##2.}%
329 ]}%
330 \def\@opargbegintheorem##1##2##3{%
331 \LWR@forcenewpage% new
332 \BlockClass{theorembodyremark}% new
333 \LWR@inctheorem% new
334 \item[\hspace{\labelsep}
335 % \theorem@headerfont
336 \InlineClass{theoremheaderremark}{##1\ ##2\ (#3).}%
337 ]}%
338
339 \gdef\th@nonumberremark{%
340 \def\theorem@headerfont{\itshape}\normalfont%
341 \def\@begintheorem##1##2{%
342 \LWR@forcenewpage% new
343 \BlockClass{theorembodyremark}% new
344 \LWR@inctheorem% new
345 \item[\hspace{\labelsep}
346 % \theorem@headerfont
347 \InlineClass{theoremheaderremark}{##1.}%
348 ]}%
349 \def\@opargbegintheorem##1##2##3{%
350 \LWR@forcenewpage% new
351 \BlockClass{theorembodyremark}% new
352 \LWR@inctheorem% new
353 \item[\hspace{\labelsep}
354 % \theorem@headerfont
355 \InlineClass{theoremheaderremark}{##1\ (#3).}%
356 ]}%
357
358 \gdef\th@proof{%
359 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
360 \def\@begintheorem##1##2{%
361 \LWR@forcenewpage% new
362 \BlockClass{theorembodyproof}% new
363 \LWR@inctheorem% new
364 \item[\hspace{\labelsep}
365 % \theorem@headerfont
366 \InlineClass{theoremheaderproof}{##1.}%
367 ]}%
368 \def\@opargbegintheorem##1##2##3{%
369 \LWR@forcenewpage% new
370 \BlockClass{theorembodyproof}% new
```

```

371 \LWR@inctheorem% new
372     \item[\hspace{\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{\ensuremath{\blacksquare}}%
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@ntheoremmarks}{%
402 \ifsetendmark{%
403 \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
404 \setendmarkfalse{%
405 \fi}%
406 }{}%
407 \LWR@origendtheorem%
408 \ifbool{LWR@ntheoremmarks}{\global\setendmarktrue}{%
409 \endBlockClass{%
410 }%
411 \gdef\NoEndMark{\global\setendmarkfalse}

```

Redefined to reuse the float mechanism to add list-of-theorem links:

```
\thm@thmline {\langle 1: printed type\rangle} {\langle 2: #\rangle} {\langle 3: optional\rangle} {\langle 4: page\rangle}

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@htmlelementclass{nav}{lothm}}%
422 \begingroup
423 \c@tocdepth=-2%
424 \def\thm@list{\thm@processlist
425 \endgroup
426 \LWR@htmlelementclassend{nav}{lothm}}%
427 }
```

Proof QED symbol:

```
428
429 \newcommand{\qed}{\qquad\the\qedsymbol}
430
431 \AtBeginDocument{
432 \def\openbox{\text{\HTMLunicode{25A1}}}%
433 \def\blacksquare{\text{\HTMLunicode{220E}}}%
434 \def\Box{\text{\HTMLunicode{25A1}}}%
435 }

\thref {\langle label\rangle}

436 \newcommand*\thref[1]{\cref{#1}}
```

Package 65

l warp-pagenote.sty

136 Pagenote

Pkg **pagenote** pagenote works as-is.

It is only included as an **l warp-pagenote.sty** file because past versions of **l warp** 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 66

l warp-parskip.sty

137 Parskip

Pkg **parskip** pars skip is ignored.

for HTML output: Discard all options for l warp-parskip.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

Package 67

l warp-placeins.sty

138 Placeins

Pkg **placeins** placeins is not used during HTML conversion.

Discard all options for l warp-placeins:

for HTML output: 1 \LWR@ProvidesPackageDrop{placeins}

```
2 \newcommand*{\FloatBarrier}{}{}
```

Package 68

l warp-ragged2e.sty

139 Ragged2e

Pkg **ragged2e** ragged2e is not used during HTML conversion.

Discard all options for l warp-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 69

lwarf-rotating.sty

140 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 \let\sidewaystable\table
3 \let\endsidewaystable\endtable
4
5 \let\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 70

lwarf-setspace.sty

141 Setspace

Pkg **setspace** setspace is not used during HTML conversion.

Discard all options for lwarf-setspace:

```
for HTML output: 1 \LWR@ProvidesPackageDrop{setspace}
                2
                3 \newcommand*{\setstretch}[1]{}
                4 \newcommand*{\SetSinglespace}[1]{}
                5 \newcommand*{\singespacing}{}
                6 \newcommand*{\onehalfspacing}{}
                7 \newcommand*{\doublespacing}{}
                8
                9 \newenvironment*{singlespace}
               10 {
               11 \LWR@forcenewpage
               12 \BlockClass{singlespace}
               13 }
               14 {\endBlockClass}
               15
               16 \newenvironment*{singespacing*}
               17 {
               18 \LWR@forcenewpage
               19 \BlockClass{singespacing}
               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}

clearpage
```

Package 71

l warp-showidx.sty

142 Showidx

Pkg showidx showidx is ignored.

for HTML output: Discard all options for l warp-showidx:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

Package 72

l warp-showkeys.sty

143 Showkeys

Pkg showkeys showkeys is ignored.

for HTML output: Discard all options for l warp-showkeys:

```
1 \LWR@ProvidesPackageDrop{showkeys}
```

```
2 \NewDocumentCommand{\showkeys}{s}{}{}
```

Package 73

l warp-sidecap.sty

144 Sidecap

Pkg sidecap sidecap is nullified.

for HTML output: Discard all options for l warp-sidecap.

```
1 \LWR@ProvidesPackageDrop{sidecap}
```

See:

[http://tex.stackexchange.com/questions/45401/
use-the-s-star-argument-with-newdocumentenvironment](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:cpx {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:cpx {SCfigure*} {\SCfigure*}
18 \cs_new_eq:cN {\endSCfigure*} \endSCfigure
19 \ExplSyntaxOff
20
21
22 \newenvironment*{wide}{}{}
```

Package 74

l warp-sidenotes.sty

145 Sidenotes

(Based on original code by ANDY THOMAS, OLIVER SCHEBAUM.)

Pkg **sidenotes** Patched for l warp.

for HTML output: Load the original package:

```
1 \LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with l warp:

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 l warp version of keyfloat:

```
22 \DeclareDocumentEnvironment{KFLT@marginfloat}{O{-1.2ex} m}
23 {%
24   \LWR@maybeinthisfloat%
25   \LWR@forcenewpage%
26   \LWR@stoppars%
27   \LWR@htmlltag{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{KFLT@marginfloat}{figure}}
37   {\end{KFLT@marginfloat}}
38
39 \RenewDocumentEnvironment{margintable}{o}
40   {\begin{KFLT@marginfloat}{table}}
41   {\end{KFLT@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_& article.cls source:

```
42 \renewenvironment{figure*}
43   {@dblfloat{figure}}
44   {\end{@dblfloat}}
45
46 \renewenvironment{table*}
47   {@dblfloat{table}}
48   {\end{@dblfloat}}
```

Package 75

lwarf-soul.sty

146 Soul

(Based on original code by MELCHIOR FRANZ.)

Pkg soul Emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{soul}

Storage for the colors to use:

```
2 \newcommand*{\LWR@soululcolor}{}  
3  
4 \newcommand*{\LWR@soulstcolor}{}  
5  
6 % \definecolor{LWR@soulhlcolordefault}{HTML}{F8E800}  
7 % \newcommand*{\LWR@soulhlcolor}{LWR@soulhlcolordefault}  
8 \newcommand*{\LWR@soulhlcolor}{}  
9 \newcommand{\so}[1]{\InlineClass{letterspacing}{#1}}  
10 \newcommand{\caps}[1]{\InlineClass{capsspacing}{#1}}
```

Basic markup with CSS:

```
9 \newcommand{\so}[1]{\InlineClass{letterspacing}{#1}}  
10 \newcommand{\caps}[1]{\InlineClass{capsspacing}{#1}}
```

Add colors if not empty:

```
11 \newcommand{\LWR@soulcolor}[4]{%  
12 \ifcseempty{#2}{%  
13 {  
14 \InlineClass{#3}{#1}}%  
15 {  
16 \convertcolorspec{named}{\csuse{#2}}{HTML}\LWR@tempcolor%  
17 \InlineClass{#3}{#4: \#\LWR@tempcolor}{#1}}%  
18 }%  
19 }  
20  
21 \newcommand{\ul}[1]{%  
22 \LWR@soulcolor{#1}{\LWR@soululcolor}{\underline}{text-decoration-color}}%  
23 }  
24  
25 \newcommand{\st}[1]{%  
26 \LWR@soulcolor{#1}{\LWR@soulstcolor}{\sout}{text-decoration-color}}%  
27 }
```

```
28
29 \newcommand{\hl}[1]{
30 \LWR@soulcolor{#1}{\LWR@soulhlcolor}{highlight}{background-color}%
31 }
```

Nullified:

```
32 \newcommand*\soulaccent}[1]{}
33 \newcommand*\soulregister}[2]{}
34 \newcommand{\sloppyword}[1]{#1}
35 \newcommand*\sodef}[5]{\DeclareRobustCommand*#1[1]{\so{##1}}}
36 \newcommand*\resetso}{}
37 \newcommand*\capsdef}[5]{}
38 \newcommand*\capsreset}{}
39 \newcommand*\capssave}[1]{}
40 \newcommand*\capsselect}[1]{}
41 \newcommand*\setul}[2]{}
42 \newcommand*\resetul}{}
43 \newcommand*\setuldepth}[1]{}
44 \newcommand*\setuloverlap}[1]{}
```

Set colors:

```
45 \newcommand*\setulcolor}[1]{\renewcommand{\LWR@soululcolor}{#1}}
46 \newcommand*\setstcolor}[1]{\renewcommand{\LWR@soulstcolor}{#1}}
47 \newcommand*\sethlcolor}[1]{\renewcommand{\LWR@soulhlcolor}{#1}}
```

Long versions of the user-level macros:

```
48 \let\textso\so
49 \let\textul\ul
50 \let\texthl\hl
51 \let\textcaps\caps
```

Package 76

lwarf-subfig.sty

147 Subfig

(Based on original code by STEVEN DOUGLAS COCHRAN.)

Pkg **subfig** **subfig** is supported and patched by **lwarf**.

⚠️ 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 **lwarf-subfig**:

```
1 \LWR@ProvidesPackagePass{subfig}
\sf@@@subfloat {\langle 1 type\rangle} [\langle 2 lof entry\rangle] [\langle 3 caption\rangle] {\langle 4 contents\rangle}
```

The outer minipage allows side-by-side subfloats with **\hfill** between.

```
2 \long\def\sf@@@subfloat#1[#2][#3]{%
3 \begin{minipage}{\linewidth}\new
4 \LWR@stoppars\new
5 \ifeundefined{FBsc@max}{}%
6 \FB@readaux{\let\FBsubheight\relax}%
7 \tempcpta=\one
8 \ifminipage
9 \tempcpta=z@
10 \else\ifdim \lastskip=z@ \else
11 \tempcpta=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                  \atminipagefalse  
34                  \or  
35                  \vskip\sff@top  
36                  \or  
37                  \ifdim \lastskip=\z@ \else  
38                      \attempskipb\sff@top\relax\xaddvskip  
39                  \fi  
40              \fi  
41              \sf@ifpositiontop{\%  
42                  \ifx \empty\relax \else  
43                      \sf@subcaption{\#1}{\#2}{\#3}\%  
44                      \vskip\sff@capskip  
45                      \vskip\sff@captionadj  
46                  \fi\egroup  
47                  \hrule width0pt height0pt depth0pt  
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\FBsubheight\relax  
60             \box\@tempboxa  
61 #4  
62         \else  
63             \vbox to \FBsubheight{\FBafil\box\@tempboxa\FBbfil}\%  
64 #4  
65         \fi}\%  
66 \LWR@stoppars% new  
67         \egroup  
68         \ifx \empty\relax \else  
69             \vskip\sff@capskip  
70             \hrule width0pt height0pt depth0pt  
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}%
85 \LWR@startpars%
86 \endgroup\ignorespaces%
87 }%  
  
\sf@subcaption {\langle 1 type\rangle} {\langle 2 lof entry\rangle} {\langle 3 caption\rangle}  
  
88 \long\def\sf@subcaption#1#2#3{%
89 \LWR@stoppars%
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             }%
98         \egroup
99     \fi
100    \bgroup
101        \ifx \relax#3\relax
102            \let\captionlabelsep=\relax
103        \fi
104        \setbox0\vbox{%
105            \hb@xt@\the\@tempdima{%
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}%
121 \LWR@htmlblocktag{figcaption}%
122 \caption@make
123 {\@nameuse{sub\@capttype name}}%
124 {\@nameuse{thesub\@capttype}}%
125 {#3}
126 \LWR@htmlblocktag{/figcaption}%
127 \LWR@traceinfo{sfsubcap B2}%
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}%
151     \caption@make
152     {\@nameuse{sub\@capttype name}}%
153     {\@nameuse{thesub\@capttype}}%
154     {#3}
155 \LWR@traceinfo{sfsubcap C1}%
156 % }%
157 % \hss
158
159 % }
160 %
161 }%
162 \egroup
163 \LWR@startpars%
164 }%
```

\caption@@make {\i<caption label>} {\i<caption text>}

```
165 \renewcommand\caption@@make[2]{%
166 \LWR@startpars%
```

```

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}%
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\@capttype\space%
209     \ifundefined{thechapter}{}{\nameuse{thechapter}\space}%
210     \nameuse{p@sub\@capttype}%
211     \nameuse{thesub\@capttype}.)}}}

```

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 \let\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 77

l warp-tabularx.sty

148 Tabularx

Pkg tabularx tabularx is emulated by l warp.

for HTML output: Discard all options for l warp-tabularx:

```
1 \LWR@ProvidesPackageDrop{tabularx}

2 \NewDocumentEnvironment{tabularx}{m o m}
3 {\begin{array}{#3}}
4 {\end{array}}
5
6 \NewDocumentEnvironment{tabularx*}{m o m}
7 {\begin{array}{#3}}
8 {\end{array}}
```

Package 78

lwarf-tabulary.sty

149 Tabulary

Pkg **tabulary** tabulary is emulated by lwarf.

for HTML output: Discard all options for lwarf-tabulary.

Column types L, C, R, and J are emulated by lwarf core code.

```
1 \LWR@ProvidesPackageDrop{tabulary}

2 \NewDocumentEnvironment{tabulary}{m o m}
3 {\begin{array}{#3}}
4 {\end{array}}
5
6 \NewDocumentEnvironment{tabulary*}{m o m}
7 {\begin{array}{#3}}
8 {\end{array}}
9
10 \newdimen\tymin
11 \newdimen\tymax
12 \def\tyformat{}
```

Package 79

lwarf-textpos.sty

150 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 80

lwarf-theorem.sty

151 Theorem

(Based on original code by FRANK MITTELBACH.)

Pkg theorem theorem is patched for use by lwarf.

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}%
8   }%
9   \%
10  \theorem@style{#1}%
11  \renewcommand{\LWR@newtheoremstyle}{#1}%
12 }%
13 \begin{group}
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\xntheorem#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             \thmcntersep \thmcnter{#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{\thmcnter{#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 \\qquad ##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 \\qquad ##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 \qquad ##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 \qquad ##1\ (###3)}%
151 ]}%
152 }

```

Patched for CSS:

```
153 \gdef\@endtheorem{\endBlockClass\endtrivlist}
```

Package 81

lwarf-threeparttable.sty

152 Threeparttable

Pkg **threeparttable** threeparttable is emulated during HTML output, and the threeparttable package is ignored.

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 \setlist[description]{format=\LWR@printtablenote}-%
9 \description-%
10 }%
11 {}%
12 \enddescription-%
13 \endBlockClass-%
14 }

15 \newcommand{\tnote}[1]{#1}

Package 82

l warp-tikz.sty

153 Tikz

Pkg **tikz** tikz is supported.

Accept all options for l warp-tikz:

```
1 \LWR@ProvidesPackagePass{tikz}

catcodes l warp 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, l warp must change $'s catcode itself.

for HTML output: 2 \begin{warpHTML}

3 \newboolean{\LWR@tikzbabel}
4
5 \@ifpackagelater{tikz}{2013/12/20}%
6 {\usepackage[babel]{tikz}\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 \latexitimage%
10 \ifbool{\LWR@tikzbabel}{%
11 {}%
12 {\catcode`\$=3} % dollar sign is math shift
13 }%
14
15 \AfterEndEnvironment{tikzpicture}{%
16 \endlatexitimage%
17 \ifbool{\LWR@tikzbabel}{%
18 {}%
19 {\catcode`\$=\active}%
20 }%
```

```
21 \end{warpHTML}
```

Package 83

lwarf-titleps.sty

154 Titleps

Pkg titleps titleps is loaded and used by lwarf during HTML output. All user options and macros are ignored and disabled.

Discard all options for lwarf-titleps:

for HTML output: 1 \LWR@ProvidesPackageDrop{titleps}

\pagestyle and \thispagestyle are already disabled in the lwarf code.

```
2 \RenewDocumentCommand{\newpagestyle}{m o m}{}  
3 \RenewDocumentCommand{\renewpagestyle}{m o m}{}  
  
4 \RenewDocumentCommand{\sethead}{o o o m m m}{}  
5 \RenewDocumentCommand{\setfoot}{o o o m m m}{}  
  
6 \RenewDocumentCommand{\settitemarks}{s m}{}  
  
7 \renewcommand*{\headrule}{}  
8 \renewcommand*{\footrule}{}  
  
9 \renewcommand*{\setheadrule}[1]{}  
10 \renewcommand*{\setfootrule}[1]{}  
  
11 \newcommand*{\makeheadrule}{}  
12 \newcommand*{\makefootrule}{}  
  
13 \renewcommand{\setmarkboth}[1]{}  
  
14 \RenewDocumentCommand{\widenhead}{s o o m m}{}  
  
15 \renewcommand*{\bottitemarks}{}  
16 \renewcommand*{\toptitemarks}{}  
17 \renewcommand*{\firstrtitlemarks}{}  
18 \renewcommand*{\nexttoptitlemarks}{}  
19 \renewcommand*{\outertitlemarks}{}  
20 \renewcommand*{\innertitlemarks}{}  
  
21 \RenewDocumentCommand{\newtitlemark}{s m}{}  

```

```
22 \RenewDocumentCommand{\pretitlemark}{s m m}{}  
23 \renewcommand{\ifsamemark}[4]{}  
  
24 \NewDocumentCommand{\setfloathead}{s o o o m m m m m}{}  
25 \NewDocumentCommand{\setfloatfoot}{s o o o m m m m m}{}  
  
26 \NewDocumentCommand{\nextfloathead}{s o o o m m m m m}{}  
27 \NewDocumentCommand{\nextfloatfoot}{s o o o m 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{\nexttopextramarks}[1]{}  
34 \newcommand{\outerextramarks}[1]{}  
35 \newcommand{\innerextramarks}[1]{}  

```

Package 84

lwarf-titlesec.sty

155 Titlesec

Pkg **titlesec** titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarf-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 \oplus
18   \fontdimen\thr@@\font \ominus \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 85

lwarf-titletoc.sty

156 Titletoc

Pkg **titletoc** titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarf-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 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 86

lwarf-titling.sty

157 Titling

Pkg **titling** titling is used by lwarf. The following patches are not needed by lwarf, but are required if the user requests titling.

lwarf uses page notes for footnotes, so the various titling footnote restyling commands have no effect.

Pass all options to lwarf-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 87

lwarf-tocloft.sty

158 Tocloft

Pkg **tocloft** tocloft is emulated. Most user options and macros are ignored and disabled.
\newlistof and \cftchapterprecis are supported.

Discard all options for lwarf-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*{\cftloftitlefont}{}
10 \newcommand*{\cftafterloftitle}{}

11 \newlength{\cftbeforeloftitleskip}
12 \newlength{\cftafterloftitleskip}

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{\cftpaiindent}  
84 \newlength{\cftpaiwidth}  
85 \newcommand*{\cftpafont}{}  
86 \newcommand*{\cftpapresnum}{}  
87 \newcommand*{\cftpaaftersnum}{}  
88 \newcommand*{\cftpaaftersnumb}{}  
89 \newcommand*{\cftpalaader}{}  
90 \newcommand*{\cftpadotsep}{1}  
91 \newcommand*{\cftpapagefont}{}  
92 \newcommand*{\cftpaafterpnum}{}  
  
93 \newlength{\cftbeforesubparaskip}  
94 \newlength{\cftsubparaindent}  
95 \newlength{\cftsubparanumwidth}  
96 \newcommand*{\cftsubparafont}{}  
97 \newcommand*{\cftsubparapresnum}{}  
98 \newcommand*{\cftsubparaftersnum}{}  
99 \newcommand*{\cftsubparaftersnumb}{}  
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{\pagenumberson}[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 \cnamedef{listof#2}{\listof{#2}{#4}}
157 \cnamedef{#2depth}{1}
158 \expandafter\newlength\csuse{cftbefore#2skip}
159 \expandafter\newlength\csuse{cft#2indent}
160 \expandafter\newlength\csuse{cft#2numwidth}
161 \cnamedef{cft#2font}{}
162 \cnamedef{cft#2presnum}{}
163 \cnamedef{cft#2aftersnum}{}
164 \cnamedef{cft#2aftersnumb}{}
165 \cnamedef{cft#2leader}{}
166 \cnamedef{cft#2dotsep}{1}
167 \cnamedef{cft#2pagefont}{}
168 \cnamedef{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     \begin{protect}\begin{quote}#1\end{protect}\end{quote}}}
178   }
179 }
180 }
```

Package 88

l warp-trivfloat.sty

159 Trivfloat

Pkg **trivfloat** trivfloat is forced to use the built-in l warp emulation for floats.

Discard all options for l warp-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 l warp.

for HTML output: 5 \begin{warpHTML}

6 \AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}

7 \end{warpHTML}

159.1 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 89

lwarf-ulem.sty

160 Ulem

(Based on original code by DONALD ARSENEAU.)

Pkg **ulem** Emulated.

for HTML output: Original lwarp definitions:

```
1 \let\LWR@ulemorig\emph\emph  
2 \let\LWR@ulemorig\textbf\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}{%  
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}{\let\emph\LWR@ulemorig\emph}  
40 \newcommand*{\ULforem}{\let\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 90

l warp-verse.sty

161 Verse

(Based on original code by PETER WILSON.)

Pkg **verse** `verse` is supported and patched by `l warp`.

for HTML output: Pass all options for `l warp-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. `l warp` 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 `\HTMLleftskip` and `\HTMLleftmargini` are provided to control the margins in HTML output. These new lengths may be set by the user before any `verse` environment, and persist until they are manually changed again. One reason to change `\HTMLleftmargini` is if there is a wide `\flagverse` in use, such as the word “Chorus”, in which case the value of `\HTMLleftmargini` should be set to a wide enough length to contain “Chorus”. The default is wide enough for a stanza number.

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

Env **verse** The `verse` environment will be placed inside a HTML `pre`.

```
2 \AfterEndPreamble{
```

At the beginning of the `verse` environment:

```

3 \AtBeginEnvironment{verse}
4 {%
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 5 \ifdef{\vleftskip}{%
6 \setlength{\vleftskip}{\HTMLvleftskip}
7 \setlength{\leftmargini}{\HTMLleftmargini}
8 }{}%
9 \LWR@forcenewpage
10 \LWR@atbeginverbatim{verse}
11 \unskip\vspace{-\baselineskip}
12 }%

```

After the end of the `verse` environment, which places the `pre` tag at the regular left margin:

```

13 \AfterEndEnvironment{verse}%
14 \unskip\vspace{-\baselineskip}%
15 \LWR@afterendverbatim
16 }%

```

Patch to place poemtitle inside an HTML span of class `poemtitle`:

```

17 \ifdef{\poemtitle}{%
18 \DeclareDocumentCommand{\vstypepoemtitle}{m}{%
19   \vspace{\beforepoemtitleskip}%
20   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
21   \vspace{\afterpoemtitleskip}%
22 }
23 }{}%
24
25 }%

```

Package 91

lwarf-wallpaper.sty

162 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*{\LCornerWallPaper}[2]{}
15 \newcommand*{\ThisLCornerWallPaper}[2]{}
16 \newcommand*{\ClearWallPaper}{}{}
17 \newlength{\wpXoffset}
18 \newlength{\wpYoffset}
```

Package 92

lwarf-wrapfig.sty

163 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@htmntag{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}{omm}
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 93

lwarf-xcolor.sty

164 Xcolor

Pkg **xcolor** `xcolor` is supported by `lwarf`.

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@htmtagc{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@htmtagc{/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}{O{named} m m}{%
16 \begingroup%
17 \LWR@colorstyle{color:}{#1}{#2}{#3}%
18 \endgroup%
19 }
```

\pagecolor [*model*] {*color*} is ignored. Use \NewCSS instead.

```
20 \newcommand*{\LWR@pagecolor}[2][named]{}
```

\colorbox [*model*] {*color*} {*text*} is converted into an HTML hex background color span.

```
21 \NewDocumentCommand{\LWR@colorbox}{O{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}}{%
29 \LWR@colorstyle{border:1px solid }{#1}{#2}{#5}%
30 }{%
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 \let\colorbox\LWR@colorbox
39 \let\fcolorbox\LWR@fcolorbox
```

Package 94

lwarf-xfrac.sty

165 Xfrac

Pkg **xfrac** Supported by adding `xfrac` instances.

for **HTML output:** 1 `\LWR@ProvidesPackagePass{xfrac}`

⚠ font size In the user's document preamble, `lwarf` should be loaded after font-related setup. During HTML conversion, this font is used by `lwarf` 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` [$\langle instance \rangle$] [$\langle num \rangle$] [$\langle sep \rangle$] [$\langle denom \rangle$]

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@htmntagc{span style="font-size:\xfracHTMLfontsize\{}%
8 \LWR@nestspan\%
9 %
10 }
```

```

11
12 \newcommand*{\LWR@htmlsmallfontend}{%
13 \LWR@htmlltagc{/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 \let\scalebox\LWR@noscalebox%
20 \LWR@htmlsmallfontstart{textsuperscript{#1}\,,\LWR@htmlsmallfontend},%
21 denominator-format = {%
22 \let\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 \let\scalebox\LWR@noscalebox%
29 \LWR@htmlsmallfontstart{textsuperscript{#1}\,,\LWR@htmlsmallfontend},%
30 denominator-format = {%
31 \let\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 \let\scalebox\LWR@noscalebox%
38 \LWR@htmlsmallfontstart{textsuperscript{#1}\,,\LWR@htmlsmallfontend},%
39 denominator-format = {%
40 \let\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 \let\scalebox\LWR@noscalebox%
47 \LWR@htmlsmallfontstart{textsuperscript{#1}\,\LWR@htmlsmallfontend},
48 denominator-format = {%
49 \let\scalebox\LWR@noscalebox%
50 \LWR@htmlsmallfontstart{}{#1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
51 scaling = false
52 }
53 \end{warpHTML}
```

Change History and Index

Change History

v0.10		
General:	2016/03/08 Initial version	1
v0.11		
General:	2016/03/11	1
Added section:		
Operating-System portability.	104	
Added section:	Selecting the	
operating system.	59	
Test Suite:	limages and index in	
README.txt	1	
Test Suite:	MS-Windows in	
README.txt	1	
v0.12		
\LWR@newhtmlfile:	Bugfix: TOC	
with numbered files.	135	
General:	2016/03/14	1
Global:	Uses \p@(type) in float	
captions.	1	
Test Suite:	Sub-figures	
	1	
v0.13		
\CaptionSeparator:	Fix for newer	
babel package.	219	
\LWR@LwarpStart:	\up and \fup	
\published:	Default to empty	
published.	155	
\subtitle:	Default to empty	
subtitle.	155	
General:	2016/03/24	1
Removed package:	subfig	
	1	
Test Suite:	Ordinals, Subcaption	
	1	
\tikzpicture:	Fix dollar-redefined	
bug for newer package.	445	
v0.14		
\LWR@htmlsectionfilename:	Fix:	
Links to home page.	110	
General:	2016/03/31	1
\floatrow:	Added.	
	361	
v0.15		
Docs: Commands for a successful		
HTML conversion.	61	
Docs: Commands into a		
warpprint environment.	61	
Docs: Newclude limitations.	73	
Docs: Table: Cross-referencing		
data structures.	207	
Docs: Table: Float data		
structures.	217	
Docs: Trademarks section.	286	
Docs: Troubleshooting		
cross-references.	82	
Test Suite: Assigned cleveref		
name for Test Float.	1	
Test Suite: \floatrow	1	
v0.16		
\printpublished:	No div if	
empty.	158	
\printsubtitle:	No div if empty.	
	158	
General:	2016/04/06	
Added	366	
Ampersand (&):	Fixed handling	
when passed as an argument.	182	
Docs: Added warning icons for		
items needing special attention.	86	
Docs: Clarify print/HTML		
output.	59	
Docs: Moved the supported		
functions table to the		
introduction.	25	
Files: l warp_formal.css	added.	
	1	
Fix: steps counter	365	
Fixed & handling.	363	
Test Suite: test_suite_formal.css		
file added.	1	
General:	2016/04/11	
	1	

\titlingpage:	Improved print-output spacing.	159
Added XeLaTeX, LuaLaTeX support.	93
Docs: Font and UTF-8 support.	58	
Docs: Moved location of \usepackage{l warp}.	59
Docs: Text not converting.	82	
Fix: amsmath options clash . . .	97	
Fix: newtxmath compatibility.	97	
Lwarp no longer selects fonts.	58, 93	
Removed package: suffix	1	
Test Suite: Improved titlingpage.	159	
Test Suite: Lwarp no longer selects fonts.	1	
Test Suite: Supports XeLaTeX, LuaLaTeX.	1	
xfrac: Adjusted for the use of any font:	469	
v0.17		
\LWR@htmlsectionfilename:	Fix: Links when entire doc is one HTML page.	110
General: 2016/04/14	1	
mdframed:	Added.	394
Test Suite: Fix: Print-version front-matter page numbers.	1	
Test Suite: Mdframed	1	
v0.18		
\LWR@hspace:	\hspace supported.	278
\LWR@includegraphicsb:	Add: svgz file extension.	258
em, ex, %, px dimensions preserved.	258	
Fix: \linewidth, \textwidth, \textheight inside a minipage.	258	
Improved HTML output linebreaks.	258	
\LWR@myshorttoc:	Reorganize \HomeHTMLfilename logic.	223
\LWR@newhtmlfile:	sidetOC after title, improving responsive design.	134
\LWR@requesttoc:	Reorganize \HomeHTMLfilename logic.	150
\LWR@subhyperref:	Improved HTML output linebreaks.	214
\LWR@subhyperrefclass:	Improved HTML output linebreaks.	214
\LWR@subinlinemimage:	Surpress extra space.	215
General: 2016/05/19	1	
File: l warp.css:	Improved TOC outline display.	1
Files: l warp.css and l warp_formal.css:	Improved responsive design.	1
Microtype disabled during HTML generation	94	
PDF Unicode input characters.	87	
Test Suite: Verse package	1	
\lateximage:	pdfcrop: --hires added.	248
Reorganize \HomeHTMLfilename logic.	248	
Surpress extra space.	248	
\verse:	Supports verse, memoir packages.	460
\minipage:	Fix: \linewidth, \textwidth, \textheight inside a minipage.	269
v0.19		
\HTMLfiveName:	Docs: Escape filename underscores.	110
\HomeHTMLFileName:	Docs: Escape filename underscores.	110
\LWR@LwarpStart:	Enabled \\ equal to \newline.	147
\LWR@doubledollar:	MathJax support.	235
\LWR@filestart:	l warp_mathjax.txt loaded.	145
\LWR@hspace:	Fix: \hspace length computations.	278
\LWR@minipagestartpars:	Surpresses paragraph tags between minipages.	276
\LWR@singledollar:	MathJax support.	235
\LWR@tabledatamultirowtag:	Added optional args.	202
\LateximageFontSizeName:	Add: User-adjustable math/lateximage font size.	247
\minipagewidth:	Added: No width tag for the next minipage in HTML.	268

\rule:	Added	281
\warpHTMLonly:	Added	88
\warpprintonly:	Replaces \rowprintedonly	88
General:	2016/06/08	1
Added		468
Avoids MathJax		234
cleveref:	Loaded	
\AtEndPreamble		264
CSS for table note item		444
Docs: multirow browser bug workaround		68, 179
Docs: Math options		59
Docs: Table: Cross-referencing data structures, updated		207
File: l warp_mathjax.txt	added	1
File: l warp.css: tnoteitemheader	added	1
Introduction: MathJax support mentioned		24
MathJax support added		236, 240, 242
Options: mathsvg and mathjax		89
Supports colored \rule		465
titleps: null \pagestyle and \thispagestyle for HTML		97
tikzpicture:	Adapts to tikz version	445
equation:	MathJax support	239
v0.20		
\BlockClassSingle:	Renamed from "LWR@htmldivclassline"	119
\HTMLfiveName:	No longer escape underscores	110
\HomeHTMLFileName:	No longer escape underscores	110
\InlineClass:	Renamed from "inlineclass"	119
\LWR@LwarpStart:	Fix: math cross references	149
\LWR@closeparagraph:	\unskip extra spaces	122
No break tags in the start/end of a tabular		122
\LWR@endofline:	Fix: \\	276
\LWR@filestart:	Adds meta description	145
\LWR@hspace:	Add: Supports HTML thin breakable space	278
\LWR@htmldivclass:	Added optional style	117
\LWR@htmlelementclass:	Added optional style	117
\LWR@htmlfilename:	HTMLFileName: removed additional trailing '-' and may be empty	110
Sections called "Index" or "index": have an underscore prepended to their filenames if no prefix		110
\LWR@includegraphicsb:	Fix: \linewidth in a floatrow	258
Fix: Expands filename		258
\LWR@longtabledatacaptiontag:	Fix: Pars in captions	199
\LWR@section:	Combined higher-level sections together into files	139
\LWR@subhtmelementclass:	Factored code	117
\SetHTMLFileName:	Add: Control file numbers	110
\cpagerefFor:	User-redefinable word for page references	265
\dotfill:	Inserts an ellipsis	275
\hfill:	Inserts a \quad	275
\hrulefill:	Inserts a short rule	275
\hyperindexref:	Print mode provided in case hyperref not used	231
\pageref:	Added	213
\tracingl warp:	Added	101
\warpOSwindows:	Auto-detects operating system	104
General:	2017/02/09	1
afterpage:	Added	339
alltt:	Added	341
bookmark:	Added	344
caption and subcaption supported		1
cleveref and referencing patches: Applied \AfterEndPreamble		264
draftwatermark:	Added	350
eso-pic:	Added	355
everypage:	Added	356
extramarks:	Added	356
fancyhdr:	Added	357
hyperref:	Additional user macros	377

keyfloat:	Added.	381
letterspace:	User-interface emulated.	382
listings:	Added.	385
ltcaption:	Added.	391
l warp-newproject:	Added.	287
microtype:	User-interface emulated.	401
needspace:	Added.	406
nowidow:	Added.	408
placeins:	Added.	421
ragged2e:	Added.	422
setspace:	Improved support.	424
textpos:	Added.	439
titleps:	Added.	446
titlesec:	Added.	448
titletoc:	Added.	449
titling:	Improved compatibility.	450
tocloft:	Added.	451
wallpaper:	Added.	462
wrapfig:	Added.	463
Added @, <, > columns.	.	180
Added \NewHTMLdescription.	.	128
Added single-expansion data arrays.	.	108
Code factored into independent l warp_html files.	.	336
Docs:	Examples for generating HTML file names.	51
Docs:	Improved index.	1
Enhanced titling support.	.	156
File:	l warp.css: Minor fixes for validation.	1
File:	l warpmk used to compile print, HTML, indexes, and lateximages.	1
Fix:	\ linewidth in a floatrow.	364
Improved float caption type handling.	.	358
Moved sidebar and example code to test suite.	.	1
Page geometry set to 6in wide with large margins.	.	94
Parallel versions of aux files for print/HTML.	.	1
Removed reliance on make, grep, gawk.	.	1
Tabular:	\ unskip extra spaces.	180
Test Suite:	HTML meta descriptions.	1
Verbatim:	Added.	171
verbatim:	Added.	168
BlockClass:	Added optional style. Renamed from "blockclass".	119
LWR@nestspan:	Fix: Minipages inside a span.	115
v0.21		
\LWR@LwarpStart:	Changed late ximages to a .txt file.	147
\LWR@filestart:	Skip title if not given.	145
\LWR@newhtmlfile:	Skip title if not given.	134
\marginpar:	Fixed source listing.	131
General:	2017/02/23	1
fontenc:	Added.	367
fontspec:	Added.	367
inputenc:	Added.	380
newinclude:	Added.	406
newunicodechar:	Added.	407
Add:	Errors for misplaced packages.	90
Docs:	Added internet class.	30
Docs:	Added TeX2page, GladTeX.	30
Docs:	Installing on Windows.	36
File l warp_tutorial.txt added.	.	41
l warpmk:	Fix: l warpmk again for Windows.	326
l warpmk:	Fix: l warpmk limages for Windows.	326
l warpmk:	Fix: l warpmk uses late ximages text file instead of shell script.	326
v0.22		
\LWR@parseDcolumn:	Added tabular D column.	188
\LWR@parsebangcolumn:	Added tabular ! column.	185
\LWR@parsetablecols:	Unknown table column types become 1. Added tabular D, !, X columns.	189
\LWR@printmccoldata:	Added tabular D, !, and X columns.	196
General:	2017/03/02	1
abstract:	Added.	337
change page:	Added.	347
dcolumn:	Added.	349
enumerate:	Added.	353
ftnright:	Added.	374

geometry:	Nullified commands.	374
indentfirst:	Added.	380
layout:	Added.	382
lscape:	Added.	391
mcaption:	Added.	393
nameref:	Added.	405
nextpage:	Added.	407
parskip:	Added.	421
showkeys:	Added.	425
sidecap:	Added.	426
tabularx:	Added.	437
varioref:	Supported.	72
verse:	Added.	460
v0.23		
\LWR@parsetablecols:	Fix for vert bar column type.	189
\LWR@printmccoldata:	Fix for vert bar column type.	196
General:	2017/03/02	1
v0.24		
\LWR@hspace:	Add: \hspace \fill converts to 2em	278
\LWR@htmlfileref:	Fix: Index links while \tracingl warp.	209
\hypertocfloat:	List of floats responds to <i>lofdepth</i> , <i>lotdepth</i> .	229
General:	2017/03/15	1
floatrow:	Support for <i>subfig</i> .	361
<i>subfig</i> :	Added.	431
tikz:	For tikz v3.0.0 or later, auto-loads tikz babel library if necessary.	445
Docs:	Filename underscore.	50, 62
No longer preloads <i>subcaption</i> ; conflicted with <i>subfig</i> .		96
<i>picture</i> :	Fix for inline images.	266
<i>tikzpicture</i> :	Fix for inline images.	445
v0.25		
\LWR@loadnever:	Added the ability to prevent conflicting packages.	91
\addcontentsline:	Handles theorems.	222
General:	2016/03/22	1
<i>amsthm</i> :	Added.	342
<i>ccaption</i> :	Prevented.	346
<i>ellipsis</i> :	Added.	350
<i>emptypage</i> :	Added.	351
<i>framed</i> :	Added.	371
<i>lipsum</i> :	Added.	384
<i>mdframed</i> :	Help avoid hyphenation.	395
<i>ntheorem</i> :	Added.	409
<i>showidx</i> :	Added.	425
<i>theorem</i> :	Added.	440
Basic L ^A T _E X theorems:	improved CSS.	174
Docs:	Adds credits for patched code.	1
Docs:	Testing <i>l warp</i> .	80
Fix:	Allows XeL ^A T _E X and LuaL ^A T _E X to preload graphics and graphicx.	91
v0.26		
General:	2017/03/31	1
<i>cutwin</i> :	Added.	348
<i>endnotes</i> :	Added.	352
<i>floatflt</i> :	Added.	360
<i>footmisc</i> :	Added.	367
<i>footnotehyper</i> :	Added.	370
<i>footnote</i> :	Added.	369
<i>marginfix</i> :	Added.	392
<i>marginnote</i> :	Added.	393
<i>mparhack</i> :	Added.	402
<i>pagenote</i> :	Supported as-is.	420
<i>sidenotes</i> :	Added.	427
<i>l warp.css</i> :	Improved responsive <i>marginpar</i> and <i>marginblock</i> .	290
Docs:	Improved MiK _T E _X install instructions.	33, 36
Dollar span avoided in a <i>lateximage</i> .		234
Footnotes now are L ^A T _E X boxes instead of <i>pagenotes</i> .		129
<i>lateximage</i> :	Labels track page numbers of <i>lateximages</i> .	248
Print mode now uses a <i>minipage</i> of <i>\ linewidth</i> .		248
<i>picture</i> :	Fix for \makebox in <i>picture</i> .	266
v0.27		
General:	2017/04/04	1
<i>lettrine</i> :	Added.	383
<i>microtype</i> :	Fix with XeL ^A T _E X, LuaL ^A T _E X.	401
<i>soul</i> :	Added.	429
<i>ulem</i> :	Added.	458
Docs:	Installing utilities for MacOS.	38

Docs: Limitations of saveboxes.	69	tabularx: Fix for optional pos.	437
Fix for table footnote par tags.	130	tabulary: Added.	438
Page geometry modified to reduce line overflow.	94	Added \HTMLauthor.	128
v0.28		Added boolean FormatEPUB.	102
\@wrindex: Improved indexing.	231	Added boolean	
\LWR@LwarpEnd: If FormatEPUB or		FormatWordProcessor.	102
FormatWordProcessor, no		Added boolean	
bottom nav.	150	HTMLDebugComments.	102
\LWR@LwarpStart:		Added boolean HTMLMarkFloats	102
FormatWordProcessor forces		Docs: Modfyng lwarpmk and	
single-file output.	147	index processing.	80
\LWR@filestart: Adds HTML meta		File l warp_mathjax.txt:	
author.	145	Updated CDN repository.	324
\LWR@forcenewpage: Forces new		Forced oneside to maintain large	
PDF page before major		right margin.	94
environments.	114	lwarpmk: Add: printglossary	
\LWR@htmlcomment: Breaks		and htmlglossary commands.	326
ligatures in HTML comments.	116	v0.29	
\LWR@includegraphicsb: Adapts		\LWR@includegraphicsb: Fix:	
to graphics syntax.	258	Error when no optional	
\LWR@newhtmlfile: If FormatEPUB		arguments.	258
or FormatWordProcessor: skips		General: 2017/04/15	1
headers, footers, nav.	134	Add: l warpmklang option for	
\LWR@parsetablecols: Added L,		l warp-newproject.	288
C, R, J column types.	189	Add: l warpmklang option for	
\LWR@startref: Removed space.	211	l warp.	89
\chapter: If EPUB, prints footnotes		Docs: Using a glossary	56
before each section.	143	File *.l warpmkconf: Add:	
\hyperindexref: Improved		language option for config	
indexing.	231	files.	290
\textup: Fixed span class.	273	File l warpmk.conf: Add:	
General: 2017/04/14	1	language option for config	
glossaries: Added.	375	files.	285
graphics: Added.	376	lwarpmk: Add: language option	
		for config files.	326

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>234</i>
\&	<i>109</i> , <u>2216</u>
\(.	<u>3254</u>
\)	<u>3254</u>
\@setcpageref	<u>3973</u>
\@setcref	<u>3969</u>
\@setcrerange	<u>3970</u>
@author	<i>153</i>
\@begintheorem	<u>2073</u>
\@capttype	<u>2949</u>
\@currentlabelname	<u>2732</u>
@date	<i>153</i>
\@dlnfloat	<u>2927</u>
\@endtheorem	<u>2083</u>
\@float	<u>2927</u>
\@fnssymbol	<u>1795</u>
\@makecaption	<u>2951</u>
\@maketitle	<u>1806</u>
\@nameedef	<u>70</u>
\@opargbegintheorem	<u>2078</u>
@published	<i>153</i>
@subtitle	<i>153</i>
@title	<i>153</i>
\@wrglossary	<u>3200</u>
\@wrindex	<u>3193</u>
\\"	<i>276</i>
\$	<i>234</i>
\$\$	<i>234</i>
A	
abstract (environment)	<u>1835</u>
abstract (package)	<i>337</i>
\abstractname	<i>52</i> , <u>1834</u>
accents	<i>137</i>
\addcontentsline	<u>3003</u>
Adobe (program)	<i>32</i>
\affiliation	<u>1501</u>
afterpage (package)	<i>339</i>
algorithmic with newfloat, trivfloat	<i>456</i>
algorithmicx (package)	<i>340</i>
align (environment)	<u>3470</u>
B	
align* (environment)	<u>3492</u>
alltt (package)	<i>341</i>
\AmS	<u>4322</u>
amsmath (package)	<i>97</i>
amsthm (package)	<i>342</i>
\and	<i>153</i>
array (package)	<i>206</i>
AsciiDoc (program)	<i>31</i>
AsciiDoctor (program)	<i>31</i>
Asciidoctor-LaTeX (program)	<i>31</i>
\attrib	<i>72</i> , <i>167</i> , <i>460</i>
\attribution	<u>1845</u>
author	
HTML meta tag	<i>64</i> , <i>128</i>
\author	<i>63</i> , <i>153</i> , <u>1508</u>
autosec	<i>138</i>
B	
babel (package)	<i>74</i>
\backmatter	<u>1168</u>
BaseJobName (option) [l warp- newproject]	<i>287</i>
\BaseJobname	<u>522</u>
BaseJobname (option) [l warp]	<i>89</i>
baseline	
tabular	<i>189</i>
\bfseries	<u>4133</u>
\BibTeX	<u>4317</u>
BlockClass (environment)	<i>739</i>
\BlockClassSingle	<i>751</i>
bookmark (package)	<i>344</i>
booktabs (package)	<i>346</i>
booleans:	
CombineHigherDepths	<i>51</i> , <i>132</i>
FileSectionNames	<i>51</i> , <i>110</i>
FormatEPUB	<i>76</i> , <i>102</i>
FormatWordProcessor	<i>78</i> , <i>102</i>
HTMLDebugComments	<i>102</i>
HTMLMarkFloats	<i>78</i> , <i>102</i>
LWR@amsmultiline	<i>240</i>
LWR@doingapar	<i>121</i>
LWR@doinghline	<i>180</i>
LWR@doingstartpars	<i>121</i>

LWR@doingtbrule	181	LWR@lateximagedepth	247
LWR@freezethisfloat	219	LWR@lateximagenumber	247
LWR@infloatrow	257	LWR@LIPage	247
LWR@intabularmetadata	181	LWR@midrulecounter	194
LWR@minipagefullwidth	268	LWR@minipagedepth	267
LWR@minipagethispar	268	LWR@nextautofloat	220
LWR@skippingmrowcell	181	LWR@nextautopage	220
LWR@starredlongtable	199	LWR@nextequation	236
LWR@startedrow	180	LWR@prevFileDepth	139
LWR@tableparcell	181	LWR@spandepth	121
LWR@tracingl warp	101	LWR@tablecolspos	184
LWR@validatablecol	189	LWR@tablecolswidth	184
LWR@verbtags	169	LWR@tabletotalcols	184
mathjax	88	LWR@tabletotalcolsnext	184
usingOSwindows	104	LWR@thisfloat	219
warpingHTML	88	SideTOCDepth	50, 226
warpingprint	88	tocdepth	50
bugs	82	\cpagerefFor	3972
BVerbatim (environment)	2058	cross-references	
		missing or incorrect	82
C			
calc (package)	96	CSS	
Calibre	76	file selection	65
caption (package)	96	l warp.css	64
\caption@begin	2982	per HTML page	65
\caption@end	2982	project-specific changes	64
\captionlistentry	2986	cutwin (package)	348
\captionof	3016		
\CaptionSeparator	2950	D	
ccaption (package)	346	danger icon	86
center (environment)	3692	\date	63
\centering	148	dcolumn (package)	349
changepage (package)	347	debugging	82
\chapter	1293	HTML debug comments	102
\citetitle	1852	\DeclareGraphicsExtensions	3722
classss:		\DeclareGraphicsRule	3722
internet	30	Deja Vu	58
cleveref (package)	264	description	
cmap (package)	59	HTML meta tag	63, 128
CombineHigherDepths (boolean)	51, 132	description (environment)	2145
comment (package)	88	displaymath (environment)	3263
Computer Modern	58	\dotfill	4149
\ConTeXt	4314	draftwatermark (package)	350
counters:			
FileDepth	51, 131	ellipsis (package)	350
lofdepth	228	\emph	4122
lotdepth	228	emptypage (package)	351
LWR@externalfilecnt	234	\end@dblfloat	2941
LWR@htmlfilename	110	\end@float	2941
LWR@latestautopage	220	endnotes (package)	352
E			

\enlargethispage	4252	equation (environment)	3330
\enskip	277, 4216	error messages	82
enumerate (environment)	2114	eso-pic (package)	355
enumerate (package)	74, 353	etoolbox (package)	87
enumitem (package)	74, 96	everyhook (package)	95
environ (package)	97	everypage (package)	356
environments:		expl3 (package)	95
abstract	1835	extramarks (package)	356
align	3470		
align*	3492	F	
BlockClass	739	fancyhdr (package)	357
BVerbatim	2058	fancyvrb (package)	95
center	3692	FAQ	82
description	2145	fbox	
displaymath	3263	around a minipage	62
enumerate	2114	\fbox	4112
equation	3330	file	
flalign	3514	inaccessible	51
flalign*	3536	underscore	50, 62
flushleft	3704	FileDepth (counter)	51, 131
flushright	3698	files:	
gather	3426	glyptounicode	58
gather*	3448	lwarf.css	64, 290
itemize	2096	lwarf_formal.css	319
lateximage	3606	lwarf_html.xdy	81, 324
longtable	2	lwarf_mathjax.txt	324
LVerbatim	2070	lwarf_sagebrush.css	314
LWR@nestspan	636	lwarf Tutorial.txt	41
LWR@tabular	2689	lwarfpmk.conf	285
math	3262	lwarfpmk.lua	80
minipage	4040	project.css	64
multiline	3379	project.lwarfpmkconf	289
multiline*	3402	project_html.tex	289
picture	265, 3990	sample_project.css	64, 323
quote	1860	tutorial.tex	41
theindex	3176	FileSectionNames (boolean)	51, 110
tikzpicture	8	\flagverse	461
titlepage	63, 1542	flalign (environment)	3514
titlingpage	63, 1548, 1561	flalign* (environment)	3536
Verbatim	1925	Flare (program)	32
verbatim	1881	float (package)	358
VerbatimClass	1917	floatflt (package)	360
verse	2	\floatname	11
warpall	34, 60	\floatplacement	14
warpHTML	42, 53, 60	floatrow (package)	361
warpprint	35, 53, 60	\floatstyle	17
epigraph (package)	354	flushleft (environment)	3704
EPUB		flushright (environment)	3698
conversion software	76	font	
HTML conversion settings	76, 102	bitmapped	58

Computer Modern	58	\HomeHTMLFilename	50
Deja Vu	58	horizontal space	
ligatures	58, 59	between minipages	276
packages	58	\href	2874
selection	58	\hrulefill	4148
size — lateximage	65, 232, 247	\hspace	277, 4245
size — math, SVG	65, 232, 247	HTML	
size — xfrac	468	commands for a successful conver-	
fontenc (package)	58, 367	sion	61
fontspec (package)	58, 367	conversion settings	50
footmisc (package)	368	debug comments	102
footnote (package)	369	EPUB	76, 102
footnotehyper (package)	370	word processor	78, 102
footnotes	129	filename generation	51
for		headings	85
		meta tag	
HTML & PRINT	86	author	64, 128
HTML output	86	description	63, 128
PRINT output	86	generator	146
FormatEPUB (boolean)	76, 102	viewport	146
FormatWordProcessor (boolean)	78, 102	tabular column conversion	190
\framebox	4109	\HTMLAuthor	64, 942
framed (package)	371	HTMLDebugComments (boolean)	102
FrameMaker (program)	32	\HTMLentity	512
Frequently Asked Questions	82	\HTMLfiveName	50, 523
\frontmatter	1165	HTMLleftmargini (length)	
ftnright (package)	374	72, 167, 168, 460	
\fup	4146	HTMLMarkFloats (boolean)	78, 102
		\HTMLunicode	513
		HTMLvleftskip (length)	72, 167, 168, 460
G		\hyperindexref	3207
gather (environment)	3426	hyperref (package)	214, 377
gather* (environment)	3448	\hypertoc	3118
GELLMU (program)	30	\hypertocfloat	3131
generator			
HTML meta tag	146	I	
geometry (package)	94, 374	icon	
gettitlestring (package)	95	warning	86
GladTeX (program)	31	ifplatform (package)	88
glossaries (package)	74, 375	images	
glossary		graphicx package	253
language	56	in strange places	83
processing	56	\includegraphics	3905
glyphounicode (file)	58	indentfirst (package)	380
graphics (package)	253, 376	InDesign (program)	32
\graphicspath	3720	\InlineClass	759
graphicx (package)	253, 376	inputenc (package)	58, 380
		internet (class)	30
H		item, list, empty	61
Hevea (program)	30	itemize (environment)	2096
\hfill	4147		
\HomeHTMLFileName	524		

\itshape	4138	longtable (environment)	2
		longtable (package)	389
J		lotdepth (counter)	228
JavaScript		lscape (package)	391
MathJax	65, 232	ltcaption (package)	391
		LuaTeX	
		detection	93
		file & section names	137
K		\LuaTeX	4308
keyfloat (package)	381	LuaTeX (program) [requirement]	34
kvoptions (package)	88	\LuaTeX	4308
L		LVerbatim (environment)	2070
label		lwarp	
in HTML	149	loading	59
math environment	241	options	59
language		[lwarp]:	
glossary	56	BaseJobname (option)	89
language HTML metadata	145	lwarpmklang (option)	56, 74, 89, 375
\LaTeX	4300	mathjax (option)	59, 89
\TeX2HTML (program)	30	mathsvg (option)	59, 89
\TeXe	4300	warpHTML (option)	59, 89
lateximage (environment)	3606	warpprint (option)	59, 88
\LateximageFontSizeName	3570	lwarp (package)	59
lateximages		[lwarp-newproject]:	
font size	65, 232, 247	BaseJobName (option)	287
\TeXML (program)	30	lwarpmk (option)	287
layout (package)	382	lwarpmklang (option)	56, 74, 287, 375
lengths:		warpHTML (option)	287
HTMLleftmargini	72, 167, 168, 460	warpprint (option)	287
HTMLvleftskip	72, 167, 168, 460	lwarp-newproject (package)	287
LWR@minipageheight	267	lwarp.css (file)	64, 290
LWR@minipagewidth	267	lwarp_formal.css (file)	319
LWR@tempheight	278	lwarp_html.xdy	
LWR@tempraise	278	customizing	81
LWR@tempwidth	278	lwarp_html.xdy (file)	81, 324
\VerbatimHTMLWidth	169	lwarp_mathjax.txt (file)	324
vleftmargini	72, 167, 460	lwarp_sagebrush.css (file)	314
vleftskip	72, 167, 460, 461	lwarp_tutorial.txt (file)	41
letterspace (package)	382	lwarpmk	
lettrine (package)	383	customizing	80
LibreOffice (program)	31	lwarpmk (option) [lwarp-newproject]	287
ligatures	58, 59, 94	lwarpmk (program)	80, 326
line numbers	86	lwarpmk.conf (file)	285
\linebreak	4246	lwarpmk.lua (file)	80
\LinkHome	574	lwarpmklang (option) [lwarp-	
lips (package)	384	newproject]	56, 74, 287, 375
list item, empty	61	lwarpmklang (option) [lwarp]	
listings (package)	385	56, 74, 89, 375	
\listof	3084	\LWR@addmathjax	3319
lmodern (package)	58	\LWR@afterendverbatim	1909
lofdepth (counter)	228		

\LWR@amsmultiline (boolean)	<i>240</i>	\LWR@htmlelementclass	<i>703</i>
\LWR@atbeginverbatim	<i>1901</i>	\LWR@htmlelementclassend	<i>708</i>
\LWR@beginhideamsmath	<i>3364</i>	\LWR@htmlelementclassline	<i>722</i>
\LWR@botnavigation	<i>582</i>	\LWR@htmlelementend	<i>732</i>
\LWR@caption@begin	<i>2963</i>	\LWR@htmlfilenumber (counter)	<i>110</i>
\LWR@caption@end	<i>2973</i>	\LWR@htmlfileref	<i>2758</i>
\LWR@clearmidrules	<i>2446</i>	\LWR@htmlmathlabel	<i>3350</i>
\LWR@closeparagraph	<i>826</i>	\LWR@htmlopencomment	<i>667</i>
\LWR@closeprevious	<i>602</i>	\LWR@htmlrefsectionfilename	<i>567</i>
\LWR@closetabledatacell	<i>2194</i>	\LWR@HTMLSanitize	<i>3573</i>
LWR@colafterspec	<i>185</i>	\LWR@HTMLSanitizeexpand	<i>3591</i>
LWR@colatspec	<i>185</i>	\LWR@htmlsectionfilename	<i>534</i>
LWR@colbangspec	<i>185</i>	\LWR@htmllspan	<i>642</i>
LWR@colbeforespec	<i>185</i>	\LWR@htmllspanclass	<i>650</i>
\LWR@copyfile	<i>326</i>	\LWR@htmlltag	<i>658</i>
\LWR@createautosec	<i>1174</i>	\LWR@htmlltagc	<i>629</i>
\LWR@currentcss	<i>927</i>	\LWR@includegraphicsb	<i>3813</i>
\LWR@descitem	<i>2133</i>	\LWR@indexitem	<i>3181, 3185, 3189</i>
\LWR@docmidrule	<i>2465</i>	\LWR@indexsection	<i>3165</i>
LWR@doingapar (boolean)	<i>121</i>	LWR@infloatarrow (boolean)	<i>257</i>
LWR@doinghline (boolean)	<i>180</i>	LWR@intabularmetadata (boolean)	<i>181</i>
LWR@doingstartpars (boolean)	<i>121</i>	\LWR@itemizeitem	<i>2089</i>
LWR@doingtbrule (boolean)	<i>181</i>	LWR@lateautopage (counter)	<i>220</i>
\LWR@domulticolumn	<i>2538</i>	LWR@latexitimagedepth (counter)	<i>247</i>
\LWR@doubledollar	<i>3232</i>	\LWR@latexitimagedepthref	<i>2762</i>
\LWR@endhideamsmath	<i>3372</i>	LWR@latexitimage number (counter)	<i>247</i>
\LWR@endminipage	<i>4004</i>	\LWR@latexitimage numberref	<i>2765</i>
\LWR@endofline	<i>4193</i>	LWR@LIpage (counter)	<i>247</i>
\LWR@endsubminipage	<i>4025</i>	\LWR@loadafter	<i>71</i>
\LWR@ensuredoingapar	<i>800</i>	\LWR@loadbefore	<i>82</i>
LWR@externalfilecnt (counter)	<i>234</i>	\LWR@loadnever	<i>87</i>
\LWR@filenamenoblocks	<i>1019</i>	\LWR@longtabledatacaptiontag	<i>2559</i>
\LWR@filestart	<i>1351</i>	\LWR@lookforpackagename	<i>265</i>
\LWR@findword	<i>262</i>	\LWR@LwarpEnd	<i>1478, 4344</i>
\LWR@floatbegin	<i>2911</i>	\LWR@LwarpStart	<i>1408, 4344</i>
\LWR@floatend	<i>2929</i>	\LWR@maketitlesetup	<i>1786</i>
\LWR@forcenewpage	<i>608</i>	\LWR@maybeinthisfloat	<i>2946</i>
LWR@freezethisfloat (boolean)	<i>219</i>	\LWR@maybenewtablerow	<i>2395</i>
\LWR@futurenonspacelet	<i>2173</i>	LWR@midrulecounter (counter)	<i>194</i>
\LWR@getexparray	<i>508</i>	\LWR@midrules	<i>194</i>
\LWR@getmynexttoken	<i>2180</i>	LWR@minipagedepth (counter)	<i>267</i>
\LWR@hidelatexequation	<i>3305</i>	LWR@minipagefullwidth (boolean)	<i>268</i>
\LWR@hspace	<i>4224</i>	LWR@minipageheight (length)	<i>267</i>
\LWR@htmloblockcomment	<i>681</i>	\LWR@minipagestartpars	<i>4198</i>
\LWR@htmloblocktag	<i>689</i>	\LWR@minipagestoppars	<i>4204</i>
\LWR@htmclosecomment	<i>667</i>	LWR@minipagethispar (boolean)	<i>268</i>
\LWR@htmlcomment	<i>674</i>	LWR@minipagewidth (length)	<i>267</i>
\LWR@htmldivclass	<i>716</i>	\LWR@multicolother	<i>2490</i>
\LWR@htmldivclassend	<i>719</i>	\LWR@multicolpartext	<i>2485</i>
\LWR@htmlelement	<i>729</i>	\LWR@multicolskip	<i>2496</i>

\LWR@mynexttoken	<u>2172</u>	\LWR@startref	<u>2805</u>
\LWR@myshorttoc	<u>3028</u>	\LWR@stopars	<u>876</u>
\LWR@nameref	<u>2755</u>	\LWR@stripperiod	<u>2733</u>
\LWR@nestspan (environment)	<u>636</u>	\LWR@strresult	<u>2246</u>
\LWR@newhtmlfile	<u>1097</u>	\LWR@subcmidrule	<u>2457</u>
\LWR@newlabel	<u>2792</u>	\LWR@subhtmelementclass	<u>694</u>
\LWR@newref	<u>2841</u>	\LWR@subhyperref	<u>2865</u>
\LWR@nextautofloat (counter)	<u>220</u>	\LWR@subhyperrefclass	<u>2869</u>
\LWR@nextautopage (counter)	<u>220</u>	\LWR@subinlineimage	<u>2904</u>
\LWR@nextequation (counter)	<u>236</u>	\LWR@subminipage	<u>4019</u>
\LWR@openparagraph	<u>805</u>	\LWR@subnewref	<u>2835</u>
\LWR@origcolspeс	<u>2247</u>	\LWR@subsublabel	<u>2771</u>
\LWR@originname	<u>3917</u>	\LWR@subtableofcontents	<u>3043</u>
\LWR@originnames	<u>3924</u>	\LWR@syncmathjax	<u>3266</u>
\LWR@parseaftercolumn	<u>2304</u>	\LWR@tablecolspeс	<u>2245</u>
\LWR@parseatcolumn	<u>2252</u>	\LWR@tablecolspos (counter)	<u>184</u>
\LWR@parsebangcolumn	<u>2274</u>	\LWR@tablecolswidth (counter)	<u>184</u>
\LWR@parsebeforecolumn	<u>2295</u>	\LWR@tabledatacolumntag	<u>2643</u>
\LWR@parseDcolumn	<u>2331</u>	\LWR@tabledatamulticolumntag	<u>2623</u>
\LWR@parseenormalcolumn	<u>2317</u>	\LWR@tabledatamultirowtag	<u>2629</u>
\LWR@parsepcolumn	<u>2327</u>	\LWR@tabledatasinglecolumntag	<u>2411</u>
\LWR@parseskipcolumn	<u>2314</u>	\LWR@tableparcell (boolean)	<u>181</u>
\LWR@parsetablecols	<u>2336</u>	\LWR@tabletotalcols (counter)	<u>184</u>
\LWR@prevFileDepth (counter)	<u>139</u>	\LWR@tabletotalcolsnext (counter)	<u>184</u>
\LWR@printmcoldata	<u>2499</u>	\LWR@tabular (environment)	<u>2689</u>
\LWR@printmcoltpe	<u>2469</u>	\LWR@tabularendofline	<u>2229</u>
\LWR@printthetitle	<u>1637</u>	\LWR@tempheight (length)	<u>278</u>
\LWR@ProvidesPackageDrop	<u>316</u>	\LWR@tempraise (length)	<u>278</u>
\LWR@ProvidesPackagePass	<u>304</u>	\LWR@tempwidth (length)	<u>278</u>
\LWR@pushoneclose	<u>1177</u>	\LWR@thisfilename	<u>1017</u>
\LWR@requesttoc	<u>1471</u>	\LWR@thisfloat (counter)	<u>219</u>
\LWR@requirepackagenames	<u>261</u>	\LWR@thisfilename	<u>1018</u>
\LWR@restoremathlatexformatting	<u>3287</u>	\LWR@titlingmaketitle	<u>1823</u>
\LWR@rotboxorigin	<u>3916</u>	\LWR@topnavigation	<u>579</u>
\LWR@rotstyle	<u>3803</u>	\LWR@traceinfo	<u>346</u>
\LWR@rule	<u>4253</u>	\LWR@tracingl warp (boolean)	<u>101</u>
\LWR@scalestyle	<u>3806</u>	\LWR@validtablecol (boolean)	<u>189</u>
\LWR@section	<u>1184</u>	\LWR@Verbatimclass	<u>1916</u>
\LWR@sectionnumber	<u>1171</u>	\LWR@verbtags (boolean)	<u>169</u>
\LWR@setexparray	<u>505</u>	\LWRsetnextfloat	<u>2954</u>
\LWR@setlatestname	<u>2734</u>	\LyX	<u>4325</u>
\LWR@sidetoc	<u>3097</u>		M
\LWR@singledollar	<u>3244</u>	Madcap (program)	<u>32</u>
\LWR@skippingmrowcell (boolean)	<u>181</u>	\mainmatter	<u>1161</u>
\LWR@spandepth (counter)	<u>121</u>	\makebox	<u>4104</u>
\LWR@splabel	<u>2768</u>	makeidx (package)	<u>96</u>
\LWR@starredlongtable (boolean)	<u>199</u>	\MakeIndex	<u>4317</u>
\LWR@startedrow (boolean)	<u>180</u>	\maketitle	<u>63, 160, 1798</u>
\LWR@startnewdepth	<u>1178</u>	\maketitlehookaa	<u>152</u>
\LWR@startpars	<u>864</u>		

\maketitlehookaaa	152	multline (environment)	3379
margin		multiline* (environment)	3402
numbers	86		
tags	86		
marginfix (package)	392	N	
marginnote (package)	393	\Nameref	2862
\marginpar	1006	\nameref	2853
markup languages	32	nameref (package)	405
math		needspace (package)	406
font size — SVG	65, 232, 247	newclude (package)	73, 406
mathjax option	89	\NewCSS	49, 53, 65, 928
MathJax summary	66, 233	newfloat	
mathsvg option	89	with trivfloat, algorithmic	456
SVG summary	65, 232	\newfloat	2
math (environment)	3262	newfloat (package)	96
MathJax		\NewHTMLdescription	53, 63, 947
mathjax option	89	\newline	4191
subequations	66, 233	\newlist	2163
summary	66, 233	\newtheorem	174
tagged equations	66, 233	newtxmath (package)	73
MathJax (program)	66, 233	newunicodechar (package)	58, 407
MathJax (program) [requirement] . . .	34	nextpage (package)	407
mathjax (boolean)	88	\nolinebreak	4247
mathjax (option) [l warp]	59, 89	\nolinkurl	2886
mathsvg (option) [l warp]	59, 89	\nopagebreak	4251
mcaption (package)	393	nowidow (package)	408
mdframed (package)	394	ntheorem (package)	409
\mdseries	4132	\numberline	3115
memoir		numbers	
verse	461	left margin	86
memoir (package)	461	O	
meta tag, HTML		OpenOffice (program)	31
author	64, 128	options:	
description	63, 128	[l warp-newproject]:	
generator	146	BaseJobName	287
viewport	146	lwarpmpk	287
\MetaLanguage	52, 1350	lwarpmpklang	56, 74, 287, 375
microtype (package)	59, 94, 401	warpHTML	287
\MiKTeX	4324	warpprint	287
minipage		[l warp]:	
horizontal space between	276	BaseJobname	89
in an \fbox	62	lwarpmpklang	56, 74, 89, 375
\minipage (environment)	4040	mathjax	59, 89
\minipagewidth	4032	mathsvg	59, 89
missing sections	51	warpHTML	59, 89
\mparhack (package)	402	warpprint	59, 88
\mrowcell	2685	\OSPathSymbol	405
MS-Windows (program)	59, 104	P	
multicol (package)	403	packages	
multirow (package)	202, 405	required	93

packages:	
abstract	337
afterpage	339
algorithmicx	340
alltt	341
amsmath	97
amsthm	342
array	206
babel	74
bookmark	344
booktabs	346
calc	96
caption	96
ccaption	346
changegetPage	347
cleveref	264
cmap	59
comment	88
cutwin	348
dcolumn	349
draftwatermark	350
ellipsis	350
emptypage	351
endnotes	352
enumerate	74, 353
enumitem	74, 96
environ	97
epigraph	354
eso-pic	355
etoolbox	87
everyhook	95
everypage	356
expl3	95
extramarks	356
fancyhdr	357
fancyvrb	95
float	358
floatflt	360
floatrow	361
fontenc	58, 367
fontspec	58, 367
footmisc	368
footnote	369
footnotehyper	370
framed	371
ftnright	374
geometry	94, 374
gettitlestring	95
glossaries	74, 375
graphics	253, 376
graphicx	253, 376
hyperref	214, 377
ifplatform	88
indentfirst	380
inputenc	58, 380
keyfloat	381
kvoptions	88
layout	382
letterspace	382
lettrine	383
lips	384
listings	385
lmodern	58
longtable	389
lscape	391
ltcaption	391
lwarp	59
lwarp-newproject	287
makeidx	96
marginfix	392
marginnote	393
mcaption	393
mdframed	394
memoir	461
microtype	59, 94, 401
mparhack	402
multicol	403
multirow	202, 405
nameref	405
needspace	406
newclude	73, 406
newfloat	96
newtxmath	73
newunicodechar	58, 407
nextpage	407
nowidow	408
ntheorem	409
pagenote	420
parskip	421
placeins	421
ragged2e	422
refcount	96
rotating	423
setspace	424
showidx	425
showkeys	425
sidecap	426
sidenotes	427
siunitx	73, 253
soul	429

subfig	431	Plastex (program)	30
tabularx	437	\popclose	478
tabulary	438	Poppler	34, 39
textcomp	59	\postpublished	152
textpos	439	\postsubtitle	152
theorem	440	\prepublished	152
threeparttable	444	\presubtitle	152, 162
tikz	445	\printauthor	153, 1654, 1669
titleps	97, 446	\printdate	153, 1659, 1671
titlesec	448	\printindex	3168
titletoc	449	\printpublished	153, 1620, 1666
titling	96, 450	\PrintStack	587
tocloft	451	\printsubtitle	153, 1645, 1668
trivfloat	456	\printthanks	152, 156
ulem	458	\printtitle	153, 1629, 1667
verse	460, 461	problems	82
wallpaper	462	programs	
wrapfig	463	utility	33
xcolor	465	programs:	
xfrac	98, 468	[requirement]:	
xifthen	96	LuaLaTeX	34
xparse	95	MathJax	34
xstring	96	pdfcrop	34
zref	97	pdfLaTeX	34
page		pdfseparate	34, 39
inaccessible	51	pdftocairo	34, 39
\pagebreak	4248	pdftotext	34, 39
\pagenote (package)	420	perl	39
\pagenumbers	254	XeLaTeX	34
\pageref	2848	Adobe	32
\pagerefPageFor	2847	AsciiDoc	31
\pagestyle	250	AsciiDoctor	31
Pandoc (program)	31	Asciidoctor-LaTeX	31
\paragraph	1332	Flare	32
\parbox	4097	FrameMaker	32
\parsemulticolumnalignment	2526	GELLMU	30
\parskip (package)	421	GladTeX	31
\part	1281	Hevea	30
\pdfcrop (program) [requirement]	34	InDesign	32
\pdfLaTeX (program) [requirement]	34	LaTeX2HTML	30
\pdfseparate (program) [requirement]	34, 39	LaTeXML	30
\pdfseparate (program) [requirement]	34, 39	LibreOffice	31
\pdftocairo (program) [requirement]	34, 39	lwarpmk	80, 326
\pdftotext (program) [requirement]	34, 39	Madcap	32
Perl	39	MathJax	66, 233
\phantomsection	4294	MS-Windows	59, 104
\picture (environment)	265, 3990	OpenOffice	31
\placeins (package)	421	Pandoc	31
		Plastex	30
		TeX2page	30

TeX4ht	30	missing	51
TtH	30	\section	1307
Windows	59, 104	\SetFirstPageTop	52, 905
Word	31	\SetHTMLFileName	525
xindy	81	\SetPageBottom	52, 153, 921
project.css (file)	64	\SetPageTop	52, 913
project.lwarpmkconf (file)	289	setspace (package)	424
project_html.tex (file)	289	settings	
\published	63, 1518	css project-specific	64
\pushclose	451	CSS selection	65
		HTML conversion	50
		lwarp package options	59
		selecting output	59
		title page	62
		\sffamily	4135
		\sfrac	468
		showidx (package)	425
		showkeys (package)	425
		sidecap (package)	426
		sidenotes (package)	427
		SideTOCDepth (counter)	50, 226
		\sidetocname	3094
		siunitx (package)	73, 253
		soul (package)	429
		\sp	4141
		space	
		horizontal between minipages	276
		stack depths	85
		subequations	
		MathJax	66, 233
		pdfseparate (program)	34, 39
		subfig (package)	431
		pdftocairo (program)	34, 39
		\subparagraph	1340
		pdftotext (program)	34, 39
		\subsection	1315
		perl (program)	39
		\subsubsection	1323
		XeLaTeX (program)	34
		\subtitle	63, 1529
		SVG	
		math summary	65, 232
		mathsvg option	89
			T
		\tableofcontents	52
		tabular	
		baseline	189
		HTML columnn conversion	190
		tabularx (package)	437
		tabulary (package)	438
		tagged equations	
		Mathjax	66, 233
		\TeX	4297
		\TeX2page (program)	30
		S	
		sample_project.css (file)	64, 323
		\sb	4142
		\scalebox	3951
		\scshape	4139, 4140
		section	
		depths	85

TeX4ht (program)	<i>30</i>	tutorial.tex (file)	<i>41</i>
\textbf{.}	<i>4124</i>		
textcomp (package)	<i>59</i>		U
\textgreater{.}	<i>515</i>	ulem (package)	<i>458</i>
\textit{.}	<i>4129</i>	underscore	
\textless{.}	<i>515</i>	filename	<i>50, 62</i>
\textmd{.}	<i>4123</i>	Unicode	
\textnormal{.}	<i>4131</i>	enhanced coverage	<i>58</i>
textpos (package)	<i>439</i>	file & section names	<i>137</i>
\textrm{.}	<i>4125</i>	input characters	<i>87</i>
\textsc{.}	<i>4130</i>	selection	<i>58</i>
\textsf{.}	<i>4126</i>	\up{.}	<i>4145</i>
\textsubscript{.}	<i>4144</i>	\upshape{.}	<i>4137</i>
.	<i>4143</i>	\url{.}	<i>2893</i>
\texttt{.}	<i>4127</i>	\UseLatexmk{.}	<i>50</i>
\textup{.}	<i>4128</i>	\UseVerbatim{.}	<i>2071</i>
\tfl@chapter@fix{.}	<i>456</i>	usingOSwindows (boolean)	<i>104</i>
\thanks{.}	<i>63</i>	UTF-8	
\thanksmarkseries{.}	<i>1827</i>	enhanced coverage	<i>58</i>
\theauthor{.}	<i>153</i>	file & section names	<i>137</i>
\thedate{.}	<i>153</i>	selection	<i>58</i>
theindex (environment)	<i>3176</i>	utility	
theorem (package)	<i>440</i>	programs	<i>33</i>
\thepublished{.}	<i>153</i>		
\thesubtitle{.}	<i>153</i>		V
\thetitle{.}	<i>153</i>	Verbatim (environment)	<i>1925</i>
\thispagestyle{.}	<i>252</i>	verbatim (environment)	<i>1881</i>
threeparttable (package)	<i>444</i>	VerbatimClass (environment)	<i>1917</i>
tikz		\VerbatimHTMLWidth (length)	<i>169</i>
catcodes	<i>445</i>	verse (environment)	<i>2</i>
dollar redefined	<i>445</i>	verse (package)	<i>460, 461</i>
tikz (package)	<i>445</i>	viewport	
tikzpicture (environment)	<i>8</i>	HTML meta tag	<i>146</i>
\title{.}	<i>63</i>	\leftmargini (length)	<i>72, 167, 460</i>
titlepage (environment)	<i>63, 1542</i>	\leftskip (length)	<i>72, 167, 460, 461</i>
titleps (package)	<i>97, 446</i>		
titlesec (package)	<i>448</i>		W
titletoc (package)	<i>449</i>	wallpaper (package)	<i>462</i>
titling (package)	<i>96, 450</i>	warning icon	<i>86</i>
titlingpage (environment)	<i>63, 1548, 1561</i>	warpall (environment)	<i>34, 60</i>
tocdepth (counter)	<i>50</i>	warpHTML (environment)	<i>42, 53, 60</i>
tocloft (package)	<i>451</i>	warpHTML (option) [lwarp-newproject]	<i>287</i>
\tracingl warp{.}	<i>83, 345</i>	warpHTML (option) [lwarp]	<i>59, 89</i>
trivfloat		\warpHTMLonly{.}	<i>33, 53, 60</i>
with newfloat, algorithmic	<i>456</i>	warpingHTML (boolean)	<i>88</i>
trivfloat (package)	<i>456</i>	warpingprint (boolean)	<i>88</i>
troubleshooting	<i>82</i>	\warpOSWindows{.}	<i>59, 104, 406</i>
HTML debug comments	<i>102</i>	warpprint (environment)	<i>35, 53, 60</i>
\ttfamily{.}	<i>4136</i>	warpprint (option) [lwarp-newproject]	<i>287</i>
TtH (program)	<i>30</i>	warpprint (option) [lwarp]	<i>59, 88</i>

\warpprintonly	32, 53, 60	XeLaTeX (program) [requirement]	34
Windows (program)	59, 104	\XeTeX	4310
Word (program)	31	xfrac (package)	98, 468
word processor		xifthen (package)	96
HTML conversion settings	78, 102	xindy	
wrapfig (package)	463	customizing	81
		xindy (program)	81
X		xparse (package)	95
xcolor (package)	465	xstring (package)	96
XeLaTeX			
detection	93		
file & section names	137		
\XeTeX	4310	Z	
		zref (package)	97