



The **lwarp** package

L^AT_EX to HTML5

v0.32 — 2017/06/09

© 2016–2017 Brian Dunn
bd@BDTechConcepts.com

Abstract

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

Documents may be produced by L^AT_EX, LuaL^AT_EX, or XeL^AT_EX. A **texlua** script removes the need for system utilities such as **make** and **gawk**, and also supports **xindy** and **latexmk**. Configuration is automatic at the first manual compile.

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

A modular package-loading system uses the **lwarp** version of a package for HTML when available. Several dozen L^AT_EX 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 **lwarp, and some things may change in response to user feedback and further project development.**

License:

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

Contents

1	lwarp.sty	20
1	Updates	20
2	Introduction	26
2.1	Supported packages and features	27
3	Alternatives	32
3.1	Internet class	32
3.2	TeX4ht	32
3.3	Translators	32
3.4	AsciiDoc	33
3.5	Pandoc	33
3.6	Word processors	33
3.7	Commercial systems	34
3.8	Comparisons	34
4	Installation	35
4.1	Installing the lwarp package	35
4.2	Installing the lwarpmk utility	38
4.2.1	Using a local copy of lwarpmk	39
4.3	Installing additional utilities	40
5	Tutorial	43
5.1	Starting a new project	43
5.2	Compiling the print version with lwarpmk	47
5.3	Compiling the HTML version with lwarpmk	48
5.4	Generating the SVG images	49
5.5	Using MathJax for math	50
5.6	Changing the CSS style	51
5.7	Customizing the HTML output	52
5.8	Using latexmk	57
5.9	Using XeLaTeX or LuaLaTeX	58
5.10	Using a glossary	59
5.11	Cleaning auxiliary files	60
5.12	Cleaning auxiliary and output files	60
5.13	Processing multiple projects in the same directory	60
5.14	Using the make utility	60
6	Additional details	61
6.1	Font and UTF-8 support	61
6.1.1	Indexes and UTF-8	62
6.2	lwarp package loading and options	62
6.3	Selecting the operating system	64

6.4	Selecting actions for print or HTML output	64
6.5	Commands to be placed into the warpprint environment	65
6.6	Commands for a successful HTML conversion	65
6.7	Title page	67
6.8	HTML page meta descriptions	68
6.9	HTML page meta author	68
6.10	CSS	68
6.11	Modifying xindy index processing	69
6.12	Special cases and limitations	70
6.12.1	Text formatting	70
6.12.2	Cross-references	70
6.12.3	cleveref and varioref packages	70
6.12.4	Footnotes and page notes	70
6.12.5	Math	70
6.12.6	ntheorem package	72
6.12.7	Graphics	72
6.12.8	xcolor package	73
6.12.9	Tabular	73
6.12.10	longtable package	74
6.12.11	Save Boxes	75
6.12.12	Minipages	75
6.12.13	mdframed package	76
6.12.14	float , trivfloat , and/or algorithmicx together	76
6.12.15	caption and subcaption packages	76
6.12.16	subfig package	76
6.12.17	floatrow package	77
6.12.18	abstract package	77
6.12.19	verse and memoir	77
6.12.20	siunitx package	78
6.12.21	newclude package	78
6.12.22	newtxmath package	78
6.12.23	babel package	79
6.12.24	glossaries package	79
6.12.25	enumitem package	79
6.12.26	enumerate package	80
7	EPUB conversion	81
8	Word-processor conversion	83
9	Modifying lwarp	84
9.1	Creating an lwarp version of a package	84
9.2	Testing lwarp	85
9.3	Modifying lwarpmk	85
10	Troubleshooting	86

10.1	Using the <code>lwarp.sty</code> package	86
10.1.1	Debug tracing output	88
10.2	Compiling the <code>lwarp.dtx</code> file	88
11	Implementation	88
12	Stack depths	89
13	Source Code	91
14	Detecting the T_EX Engine — pdf_latex, lua_latex, xel_latex	92
15	Unicode Input Characters	92
16	Early package requirements	93
17	Operating-System portability	94
17.1	Common portability code	94
17.2	Unix, Linux, and Mac OS	94
17.3	MS-Windows	94
18	Package options	95
19	Misplaced packages	98
20	Required packages	101
21	Loading packages	106
22	Copying a file	109
23	Debugging messages	109
24	HTML-conversion output modifications	110
25	Remembering original formatting macros	111
26	Configuration Files	112
26.1	<code>project_html.tex</code>	112
26.2	<code>lwarpmk.conf</code>	112
26.3	<code>project.lwarpmkconf</code>	113
26.4	<code>lwarp.css</code>	114
26.5	<code>lwarp_sagebrush.css</code>	138
26.6	<code>lwarp_formal.css</code>	143
26.7	<code>sample_project.css</code>	147
26.8	<code>lwarp.xdy</code>	148
26.9	<code>lwarp_mathjax.txt</code>	148
26.10	<code>lwarpmk</code> option	150

27	Stacks	161
27.1	Assigning depths	161
27.2	Closing actions	162
27.3	Closing depths	162
27.4	Pushing and popping the stack	163
28	Data arrays	164
29	HTML entities	165
30	HTML filename generation	166
31	Homepage link	168
32	\PrintStack diagnostic tool	169
33	Closing stack levels	169
34	Forcing a new PDF page	170
35	HTML tags, spans, divs, elements	170
35.1	Mapping L ^A T _E X Sections to HTML Sections	171
35.2	HTML tags	171
35.3	Block tags and comments	172
35.4	Div class and element class	173
35.5	Single-line elements	174
35.6	HTML5 semantic elements	175
35.7	High-level block and inline classes	175
35.8	Closing HTML tags	176
36	Paragraph handling	177
37	Paragraph start/stop handling	181
38	Page headers and footers	183
39	CSS	184
40	HTML meta description and author	185
41	Footnotes	185
41.1	Regular page footnotes	185
41.2	Minipage footnotes	186
41.3	Titlepage thanks	186
41.4	Regular page footnote implementation	186
42	Marginpars	188

43	Splitting HTML files	188
44	Sectioning	193
44.1	Book class commands	194
44.2	Sectioning support macros	194
44.3	\section and friends	199
45	Starting a new file	201
46	Starting HTML output	204
47	Ending HTML output	207
48	Titles and the titling package	208
48.1	Setting the title, etc.	210
48.2	Changes to HTML titlepage and titlingpage	212
48.3	Printing the title, etc. in HTML	213
48.4	Printing the title, etc. in print form	216
48.5	\maketitle for print output	217
48.6	\maketitle for HTML output	219
49	Abstract	222
50	Quote and verse	223
50.1	Citations and attributions	223
50.2	Quotes, quotations	223
50.3	Verse	224
51	Verbatim	225
52	Fancyvrb	225
53	Theorems	231
54	Lists	232
54.1	Itemize	232
54.2	Enumerate	233
54.3	Description	234
55	Tabular	235
55.1	Token lookahead	237
55.2	Booleans	237
55.3	Handling & and !	238
55.4	Handling \\	240
55.5	Variables	241
55.6	Parsing @, >, <, ! columns	242
55.7	Parsing 'l', 'c', or 'r' columns	244
55.8	Parsing 'p', 'm', or 'b' columns	245

55.9	Parsing ‘D’ columns	245
55.10	Parsing the column specifications	245
55.11	Starting a new row	249
55.12	Data opening tag	250
55.13	Midrules	251
55.14	Multicolumns	252
55.14.1	Parsing multicolumns	252
55.14.2	High-level multicolumn interface	255
55.14.3	Longtable captions	256
55.14.4	<code>\tabledata multicolumntag</code>	258
55.15	Multirow	259
55.16	Utility macros inside a table	259
55.17	Checking for a new table cell	259
55.18	<code>\mrowcell</code>	261
55.19	New <code>\tabular</code> definition	261
55.20	Array	263
56	Cross-references	264
56.1	Setup	264
56.2	Zref setup	266
56.3	Labels	267
56.4	References	268
56.5	Hyper-references	271
57	Floats	274
57.1	Float captions	275
57.1.1	Caption inside a float environment	276
57.1.2	Caption and LOF linking and tracking	277
58	Table of Contents, LOF, LOT	280
58.1	Reading and printing the TOC	280
58.2	High-level TOC commands	283
58.3	Side TOC	283
58.4	Low-level TOC line formatting	284
59	Index and glossary	287
60	Math	289
60.1	Inline and display math	291
60.2	MathJax support	293
60.3	Equation environment	296
60.4	AMS Math environments	297
60.4.1	Support macros	297
60.4.2	Environment patches	299
61	Lateximages	303

62	center, flushleft, flushright	309
63	Siunitx	310
64	Graphics	310
64.1	<code>\graphicspath</code>	311
64.2	Length conversions and graphics options	311
64.3	<code>\includegraphics</code>	314
64.4	<code>\rotatebox</code> , <code>\scalebox</code> , <code>\reflectbox</code>	318
64.5	Null functions	321
65	Cleverref	321
66	Picture	322
67	Boxes and Minipages	323
67.1	Counters and lengths	324
67.2	Footnote handling	324
67.3	Minipage handling	324
67.4	Parbox, makebox, framebox, fbox, raisebox	328
68	Direct formatting	329
69	Skips, spaces, font sizes	332
70	<code>\phantomsection</code>	339
71	<code>\LaTeX</code> and other logos	339
71.1	HTML logos	339
71.2	Print logos	341
72	<code>\AtBeginDocument</code>, <code>\AtEndDocument</code>	341
73	Trademarks	342
2	lwarp-abstract.sty	343
74	Abstract	343
3	lwarp-afterpage.sty	345
75	Afterpage	345

4	lwarp-algorithmicx.sty	346
76	Algorithmicx	346
5	lwarp-alltt.sty	346
77	Alltt	347
6	lwarp-amsthm.sty	348
78	AMSthm	348
7	lwarp-bookmark.sty	350
79	Bookmark	350
8	lwarp-booktabs.sty	352
80	Booktabs	352
9	lwarp-ccaption.sty	352
81	Ccaption	352
10	lwarp-changepage.sty	353
82	Changepage	353
11	lwarp-cutwin.sty	354
83	Cutwin	354
12	lwarp-dcolumn.sty	354
84	Dcolumn	355
13	lwarp-draftwatermark.sty	356
85	Draftwatermark	356

14	lwarp-ellipsis.sty	356
86	Ellipsis	356
15	lwarp-emptypage.sty	357
87	Emtypage	357
16	lwarp-endnotes.sty	358
88	Endnotes	358
17	lwarp-enumerate.sty	359
89	Enumerate	359
18	lwarp-epigraph.sty	360
90	Epigraph	360
19	lwarp-eso-pic.sty	361
91	Eso-pic	361
20	lwarp-everypage.sty	362
92	Everypage	362
21	lwarp-extramarks.sty	362
93	Extramarks	362
22	lwarp-fancyhdr.sty	363
94	Fancyhdr	363
23	lwarp-float.sty	364
95	Float and \newfloat	364

24	lwarp-floatflt.sty	366
96	Floatflt	366
25	lwarp-floatrow.sty	367
97	Floatrow	367
26	lwarp-fontenc.sty	373
98	Fontenc	373
27	lwarp-fontspec.sty	373
99	Fontspec	373
28	lwarp-footmisc.sty	373
100	Footmisc	373
29	lwarp-footnote.sty	375
101	Footnote	375
30	lwarp-footnotehyper.sty	376
102	Footnotehyper	376
31	lwarp-framed.sty	377
103	Framed	377
32	lwarp-ftnright.sty	380
104	Ftnright	380
33	lwarp-geometry.sty	380
105	Geometry	380

34	lwarp-glossaries.sty	381
106	Glossaries	381
35	lwarp-graphics.sty	382
107	Graphics	382
36	lwarp-graphicx.sty	382
108	Graphicx	382
37	lwarp-hyperref.sty	383
109	Hyperref	383
38	lwarp-indentfirst.sty	386
110	Indentfirst	386
39	lwarp-inputenc.sty	386
111	Inputenc	386
40	lwarp-keyfloat.sty	387
112	Keyfloat	387
41	lwarp-layout.sty	389
113	Layout	389
42	lwarp-letterspace.sty	389
114	Letterspace	389
43	lwarp-lettrine.sty	390
115	Lettrine	390

44	lwarp-lips.sty	391
116	Lips	391
45	lwarp-listings.sty	392
117	Listings	392
46	lwarp-longtable.sty	396
118	Longtable	396
47	lwarp-lscape.sty	398
119	Lscape	398
48	lwarp-ltcaption.sty	398
120	Ltcaption	398
49	lwarp-marginfix.sty	399
121	Marginfix	399
50	lwarp-marginnote.sty	400
122	Marginnote	400
51	lwarp-mcaption.sty	400
123	Mcaption	400
52	lwarp-mdframed.sty	401
124	Mdframed	401
53	lwarp-microtype.sty	408
125	Microtype	408

54	lwarp-mparhack.sty	409
126	Mparhack	409
55	lwarp-multicol.sty	410
127	Multicol	410
56	lwarp-multirow.sty	412
128	Multirow	412
57	lwarp-nameref.sty	412
129	Nameref	412
58	lwarp-needspace.sty	413
130	Needspace	413
59	lwarp-newclude.sty	413
131	Newclude	413
60	lwarp-newunicodechar.sty	414
132	Newunicodechar	414
61	lwarp-nextpage.sty	414
133	Nextpage	414
62	lwarp-nowidow.sty	414
134	Nowidow	415
63	lwarp-ntheorem.sty	416
135	Ntheorem	416

64	lwarp-pagenote.sty	427
136	Pagenote	427
65	lwarp-parskip.sty	428
137	Parskip	428
66	lwarp-placeins.sty	428
138	Placeins	428
67	lwarp-ragged2e.sty	429
139	Ragged2e	429
68	lwarp-rotating.sty	430
140	Rotating	430
69	lwarp-setspace.sty	431
141	Setspace	431
70	lwarp-showidx.sty	432
142	Showidx	432
71	lwarp-showkeys.sty	432
143	Showkeys	432
72	lwarp-sidecap.sty	433
144	Sidecap	433
73	lwarp-sidenotes.sty	434
145	Sidenotes	434

74	lwarp-soul.sty	436
146	Soul	436
75	lwarp-subfig.sty	438
147	Subfig	438
76	lwarp-tabularx.sty	444
148	Tabularx	444
77	lwarp-tabulary.sty	445
149	Tabulary	445
78	lwarp-textpos.sty	446
150	Textpos	446
79	lwarp-theorem.sty	447
151	Theorem	447
80	lwarp-threeparttable.sty	451
152	Threeparttable	451
81	lwarp-tikz.sty	452
153	Tikz	452
82	lwarp-titleps.sty	453
154	Titleps	453
83	lwarp-titlesec.sty	455
155	Titlesec	455

84	lwarp-titletoc.sty	456
156	Titletoc	456
85	lwarp-titling.sty	457
157	Titling	457
86	lwarp-tocloft.sty	458
158	Tocloft	458
87	lwarp-trivfloat.sty	463
159	Trivfloat	463
159.1	Combining <code>\newfloat</code> , <code>\trivfloat</code> , and <code>algorithmicx</code>	463
88	lwarp-ulem.sty	465
160	Ulem	465
89	lwarp-verse.sty	467
161	Verse	467
90	lwarp-wallpaper.sty	469
162	Wallpaper	469
91	lwarp-wrapfig.sty	470
163	Wrapfig	470
92	lwarp-xcolor.sty	472
164	Xcolor	472

lwarp	18
-------	----

93	lwarp-xfrac.sty	475
-----------	------------------------	------------

165	Xfrac	475
------------	--------------	------------

	Change History and Index	478
--	---------------------------------	------------

List of Figures

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

List of Tables

1	L ^A T _E X–HTML generation — lwarp package — Supported functions	27
2	Additional supported packages	31
3	Required software programs	36
4	Files created along with the print version	46
5	Package options	63
6	Section depths and HTML headings	90
7	Tabular baseline	246
8	Tabular HTML column conversions	247
9	Cross-referencing data structures	265
10	Float data structures	274

Package 1

lwarp.sty

1 Updates

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

For a detailed list of changes, see the Change History on page 478.

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

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

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

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

v0.30:

 `lwarp-newproject`

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

To modify existing documents, remove from the document source:

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

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

 HTML setup changes.

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

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

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

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

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

```

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

```
- The file `lwarp_html.xdy` has been renamed `lwarp.xdy`. To update each document's project:
 1. Make the changes shown above.
 2. Recompile the document in print mode. This updates the project's configuration files, and also generates the new file `lwarp.xdy`.
 3. The old file `lwarp_html.xdy` may be deleted.
- The new `lwarp` package option `xdyFilename` may be used to tell `lwarpmk` to use a custom `.xdy` file instead of `lwarp.xdy`. See section 6.11.
- Improvements in index processing:
 - `xindy`'s language is now used for index processing as well as glossary.
 - Print mode without `latexmk` now uses `xindy` instead of `makeindex`.
 - `texindy`/`xindy` usage depends on `pdflatex` vs `xelatex`, `lualatex`.
 - For `pdflatex` and `texindy`, the `-C utf8` option is used. This is supported in modern distributions, but a customized `lwarpmk.lua` may need to be created for use with older distributions.

v0.29:

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

v0.28:

- `\HTMLAuthor {<name>}` assigns HTML meta author if non-empty. Defaults to `\theauthor`.
- Boolean `HTMLDebugComments` controls whether HTML comments are added for closing `<div>`s, opening and closing sections, etc.
- Boolean `FormatEPUB` changes HTML output for easy EPUB conversion via an external program. Removes per-file headers, footers, and nav. Adds footnotes per chapter/section.
- Boolean `FormatWordProcessor` changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments.
- Boolean `HTMLMarkFloats` adds text marks around floats only if `FormatWordProcessor`. These make it easier to identify float boundaries, which are to be manually converted to word-processor frames.
- Updated for the new MathJax CDN repository.
- Adds `tabulary`.
- Supports the options syntax for `graphics`.
- Improved index references, now pointing exactly to their target.
- Adds glossaries. `lwarpmk` is modified to add `printglossary` and `htmlglossary` actions.

v0.27:

- Improved documentation for MacOS install.
- Fix for microtype with `XYLaTeX` and `LuaLaTeX`.
- Fix for table footnote paragraph tags.
- Adds `lettrine`, `ulem`, and `soul`.

v0.26:

- Improved installation instructions for MiKTeX regarding generating the `lwarpmk` executable.
- Footnotes are now supported by `LaTeX` 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 `lateximages`. 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_YL^AT_EX and LuaL^AT_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 `varioref`.

v0.21:

- Documentation for installing on Windows has been updated and improved.
- For Windows compatibility, the `lateximages` shell script has been replaced with a `lateximages.txt` file, which is parsed by `lwarpmk` to generate `lateximages`. 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. (v0.30 changes this to `\HTMLDescription`.)
- `\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. (Version 0.30 changes these to package options `HomeHTMLFilename` and `HTMLFilename`.)
- `lwarp` now tries to auto-detect the operating system, and `\warpOSwindows` is only needed if the auto-detection fails to detect Windows. (As of v0.30, `\warpOSwindows` has been converted to the `OSWindows` option.)
- 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 6.12.5. 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 `lateximages`.
- `\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 65.
- `\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 $\text{X}\text{E}\text{L}\text{A}\text{T}\text{E}\text{X}$ and $\text{L}\text{u}\text{a}\text{L}\text{A}\text{T}\text{E}\text{X}$. See section 6.1.

2 Introduction

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

Several “modern” features of HTML5, CSS3, and SVG are employed to allow a fairly feature-rich document without relying on the use of Javascript. Limited testing on older browsers show that these new features degrade gracefully, although the SVG format for math may not be available on small cell phones.

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

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

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

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

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

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

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

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

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

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

Enough functionality is provided to convert a typical article containing technical content. Not every package has been tested, but many of the most useful ones are known to work, either as-is or through emulation with replacement code. (See table 1 on page 27.)

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

2.1 Supported packages and features

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

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

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

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

lwarp Supported Functions — continued

Category	Status
Sectioning:	Supported, with hyperlinks. Honors <code>tocdepth</code> and <code>secnumdepth</code> . Adds <code>filedepth</code> for splitting the HTML output. Files may be numbered sequentially or named according to section name. Common short words and punctuation is removed from the filenames.
Table of Contents, Figures, Tables:	Supported, with hyperlinks.
Title page:	<code>\maketitle</code> , <code>titlepage</code> , <code>titling</code> . Optional titling-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 \LaTeX source for the math expression. MathJax supported as an alternative. \mathcal{AMS} environments are supported. User-defined macros are available during conversion, due to native \LaTeX processing.
Theorems:	Support for native \LaTeX theorems, plus <code>theorem</code> , <code>amsthm</code> , <code>ntheorem</code> .
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> .

lwarp 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</code> , <code>tabulary</code> , <code>threeparttable</code> , <code>multirow:</code>	Emulated.
<code>longtable:</code>	Emulated. Converted to a <code>tabular</code> . Captions supported. Extra headings and <code>\kill</code> lines must be enclosed in <code>\warpprintonly{}</code>
<code>booktabs:</code>	Emulated. <code>\toprule</code> and <code>\bottomrule</code> form black rules, <code>\midrule</code> forms silver rules, as demonstrated on this table. <code>\cmidrule</code> , demonstrated at this line, does not use width or trim options.
<code>graphics</code> , <code>graphicx:</code>	Emulated. <code>\includegraphics</code> supports <code>width</code> , <code>height</code> , <code>origin</code> , <code>angle</code> , and <code>scale</code> tags, and adds <code>class</code> . References to PDF files are changed to SVG, other image types are accepted as well. <code>\rotatebox</code> and <code>\scalebox</code> are supported as well as HTML can handle.
<code>rotating:</code>	Emulated. All objects are displayed unrotated.
<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</code> and <code>tikz:</code>	Converted to an SVG image.
<code>minipage:</code>	Supported with some HTML5-imposed limitations. Nested minipages are supported. Footnotes appear at the bottom of the HTML page.
<code>fancyvrb:</code>	Supported except for verbatim footnotes.
<code>framed</code> , <code>mdframed:</code>	Supported

lwarp 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 <code>thin-unbreakable</code> , <code>unbreakable</code> , <code>\enskip</code> , <code>\quad</code> , <code>\qqquad</code> , <code>\hspace</code> .
Rules:	<code>\rule</code> with width, height, raise, text color.
HTML reserved characters:	<code>\&</code> , <code>\textless</code> , and <code>\textgreater</code> are converted to HTML entities.
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 \LaTeX package is used.
Emulated:	The \LaTeX package is not used, but some/all of its functions are emulated. Null functions, lengths, and counters are provided for source compatibility.

Supported packages include everything listed in the table of contents, plus 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, varioref, 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 \LaTeX or other document to HTML. Where an existing \LaTeX document is to be converted to HTML, **lwarp** may be a good choice. For new projects with a large number of documents, it may be worth investigating the alternatives before decided which path to take.

3.1 Internet class

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

3.2 TeX4ht

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

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

3.3 Translators

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

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

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

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

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

Finally, GladTeX may be used to directly insert L^AT_EX math into HTML:

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

3.4 AsciiDoc

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

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

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

The Asciidoctor-LaTeX project is adding additional L^AT_EX-related features.

Asciidoctor-LaTeX:

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

3.5 Pandoc

Prog Pandoc

A markup system which also reads and writes L^AT_EX and HTML.

Pandoc: <http://pandoc.org/>

(Watch for improvements in cross-references to figures and tables.)

3.6 Word processors

Prog Word It should be noted that the popular word processors have advanced through the years in their abilities to represent math with a L^AT_EX-ish input syntax, unicode math fonts, and high-quality output, and also generate HTML with varying success.

Prog LibreOffice

Prog OpenOffice See recent developments in Microsoft® Word® and LibreOffice™ Writer.

3.7 Commercial systems

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

3.8 Comparisons

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

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

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

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

4 Installation

Table 3 shows the tools which are used for the L^AT_EX to HTML conversion. In most cases, these will be available via the standard package-installation tools.

4.1 Installing the lwarp package

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

Pre-installed: Try entering into a command line:

```
Enter ⇒ kpsewhich lwarp.sty
```

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

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

```
Enter ⇒ tlmgr install lwarp
```

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

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

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

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

T_EX Live:

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

MiK_TE_X:

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

This should be something like:

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

4. Unpack the archive in the TDS local directory.

Table 3: Required software programs

Provided by your L^AT_EX distribution:

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

L^AT_EX: pdf_latex, xelatex, or lualatex.

The lwarp package: This package.

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

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

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

latexmk: Optionally used by lwarpmk to compile L^AT_EX code. On a MiK_TE_X system, Perl may need to be installed first.

pdftcrop: 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 \Rightarrow `mktextlsr`

— or —

Enter \Rightarrow `texhash`

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

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

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

Enter \Rightarrow `pdflatex lwarp.dtx`

(several times)

6. Create the .sty files:

Enter \Rightarrow `pdflatex lwarp.ins`

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

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

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

`<texlocal>/doc/local/lwarp`

9. Renew the cache:

Enter \Rightarrow `mktextlsr`

— or —

Enter \Rightarrow `texhash`

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

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

Just testing!

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

4.2 Installing the lwarpmk utility

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

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

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

Otherwise, continue with the following:

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

```
Enter ⇒ kpsewhich lwarpmk.lua
```

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

4. Create lwarpmk:

Unix: Create a symbolic link and make it executable:

- (a) Locate the T_EX Live binaries:

```
Enter ⇒ kpsewhich -var-value TEXMFROOT
```

This will be something like:

```
/usr/local/texlive/<year>
```

The binaries are then located in the bin/<arch> directory under the root:

```
/usr/local/texlive/<year>/bin/<architecture>/
```

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

- (b) In the binaries directory, create a new symbolic link from the binaries directory to lwarpmk.lua:

```
Enter ⇒ ln -s <pathtolwarpmk.lua> lwarpmk
```

- (c) Make the link executable:

Enter \Rightarrow `chmod 0755 lwarpmk`

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

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

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

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

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

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

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

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

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

4.2.1 Using a local copy of `lwarpmk`

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

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

`\usepackage[lwarpmk]{lwarp}`

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

Unix:

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

Enter \Rightarrow `chmod 0755 lwarpmk.lua`

- (b) Compile documents with

Enter \Rightarrow `./lwarpmk.lua html`

Enter \Rightarrow `./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 36.

Enter ⇒ `luatex -version`

Enter ⇒ `xindy -version`

Enter ⇒ `latexmk -version`

Enter ⇒ `perl -version`

Enter ⇒ `pdftocrop -version`

Enter ⇒ `pdftotext -v`

Enter ⇒ `pdfseparate -version`

Enter ⇒ `pdftocairo -v`

To install xindy, latexmk, and pdfcrop:

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

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

Prog pdftotext
 Prog pdfseparate
 Prog pdftocairo

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

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

To install the Poppler utilities to a MacOS machine:

1. Install Homebrew from <https://brew.sh/>:

Enter ⇒

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

2. Install the Poppler utilities:

Enter ⇒ `brew install poppler`

To install the Poppler utilities to a Windows machine:

1. See table 3 on page 36.
2. Download and extract the Poppler utilities pdftotext, pdfseparate, and pdftocairo to a directory, such as Poppler.
3. In the Start window, type "Path" to search for results related to Path. Or, open the control panel and search for "Path".
4. Choose "Edit the system environment variables" in the control panel.
5. Choose the "Environment Variables" button.
6. Choose the "Path" variable, then the "Edit" button.
7. Choose the "New" button to make an additional entry.
8. Enter the bin directory of the Poppler utilities, such as:

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

Be sure to include \bin.

9. Click "Ok" when done.

To install Perl to a Windows machine:

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

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

`C:\Strawberry\perl\bin`

Be sure to include `\bin`.

4. Click "Ok" when done.

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

5 Tutorial

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

5.1 Starting a new project

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

File `tutorial.tex`

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

Copy from the documentation PDF:

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

Copy from the `lwarp` documentation directory:

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

Enter \Rightarrow `texdoc -l lwarp_tutorial.txt`

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

File `lwarp_tutorial.txt`

 Note: `.txt` suffix!

 Bad formatting!

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

3. Compile the project:

Enter \Rightarrow `pdflatex tutorial.tex`

(several times)

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

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

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

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

Figure 1: tutorial.tex listing

Note: There are two pages!

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

\documentclass{book}

\usepackage{iftex}

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

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

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

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

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

```

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

\begin{document}

\maketitle                % Or titlepage/titlingpage environment.

% An article abstract would go here.

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

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

\section{Some math}

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

\printindex

\end{document}

```

Table 4: Files created along with the print version

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

When the lwarpmk option is given to the lwarp package:

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

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

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

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

5.2 Compiling the print version with `lwarpmk`

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

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

1. Re-compile the print version:

Enter \Rightarrow `lwarpmk print`

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

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

Enter \Rightarrow `lwarpmk print`

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

4. Force a recompile to occur.

Enter \Rightarrow `lwarpmk again`

Enter \Rightarrow `lwarpmk print`

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

5. Process the index.⁴

Enter \Rightarrow `lwarpmk printindex`

6. Recompile again to include the index.

Enter \Rightarrow `lwarpmk print`

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

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

⁴A `lwarpmk printglossary` command is also available to process a glossary produced with the `glossaries` package. See section [6.12.24](#).

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 \Rightarrow `lwarpmk html`

- (a) `lwarpmk` uses \LaTeX to process `tutorial_html.tex` to create `tutorial_html.pdf`.
- (b) `pdftotext` is then used to convert to the file `tutorial_html.html`. This file is a plain-text file containing HTML tags and content for the entire document.
- (c) `lwarpmk` manually splits `tutorial_html.html` into individual HTML files according to the HTML settings. For this tutorial, the result is `tutorial.html` (the home page), along with `First-chapter.html`⁵, `Some-math.html`, and the document's index in `_Index.html`.⁶

2. View the homepage in a web browser.

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

math

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

3. Force a recompile:

Enter \Rightarrow `lwarpmk again`

Enter \Rightarrow `lwarpmk html`

Enter \Rightarrow `lwarpmk print`

4. Process the HTML index and recompile:⁷

Enter \Rightarrow `lwarpmk htmlindex`

Enter \Rightarrow `lwarpmk html`

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

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

⁵`First-chapter.html` also contains the first section, even though the second section is its own HTML page. This behavior is controlled by the boolean `CombineHigherDepths`.

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

⁷A `lwarpmk htmlglossary` command is also available to process a glossary produced with the `glossaries` package. See section 6.12.24.

5.4 Generating the SVG images

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


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


1. Create the SVG images:

```
Enter  $\Rightarrow$  lwarpmk limages
```

```
Enter  $\Rightarrow$  lwarpmk html
```

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

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

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

5.5 Using MathJax for math

math with MathJax Math may also be represented using the MathJax Javascript project.

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

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

2. Recompile

```
Enter ⇒ lwarpmk html
```

3. Reload the math page.



MathJax requirements

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

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

5.6 Changing the CSS style

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

The styles provided by `lwarp` include:

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

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

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

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

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




Rename it!

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

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

5.7 Customizing the HTML output

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

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

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

Options for the `lwarp` package:

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

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

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

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

Ctrl **SideTOCDepth**

[sideTOC](#)

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

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

It is recommended to set:

`SideTOCDepth=FileDepth`

or

`SideTOCDepth=FileDepth+1`

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



Ctrl **FileDepth**

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

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

`\setcounter{FileDepth}{-5}`

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

`\setcounter{FileDepth}{1}`



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

`tocdepth >= FileDepth`

Bool **CombineHigherDepths**

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

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

Inaccessible pages!

Lost in an old page!

When making changes to the file structure, it is possible to end up with the web browser pointing to an old file which is no longer in use. When this occurs, changes to the web site will not appear in the browser, even

if reloading the page, because that page is no longer in use. It is best to return to the home page, clean the files (`lwarpmk cleanall`), change `FileDepth` and/or `CombineHigherDepths`, then finally recompile and renavigate to the desired page using the new file structure.

Bool `FileSectionNames`

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

HTML filenames

Example HTML filenames:

Numbered html nodes:

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

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

Named html sections, no prefix:

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

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

Named html sections, with prefix:

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

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

`\abstractname`

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

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

`\CSSFilename`


\CSSFilename: `{\filename.css}` Sets the CSS file to use for the following

files. May be changed before each each sectioning command which would cause a file split.

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

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

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

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

Placed in the document wherever necessary:

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

won't have to be placed inside a **warpprint** environment, but unknown packages may cause problems which may be isolated from **lwarp** using this environment.

Env **warpHTML**

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

\warpprintonly

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

\warpHTMLonly

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

5.8 Using latexmk

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

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

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

2. Recompile the printed version of the document.

```
Enter ⇒ lwarpmk print
```

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

3. Recompile the document.

```
Enter ⇒ lwarpmk print
```

and/or

```
Enter ⇒ lwarpmk html
```

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

5.9 Using XeLaTeX or LuaLaTeX

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

1. Remove the auxiliary files for the project:

Enter \Rightarrow `lwarpmk cleanall`

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

Enter \Rightarrow `xelatex tutorial.tex`

-or-

Enter \Rightarrow `lualatex tutorial.tex`

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

3. To recompile the document:

Enter \Rightarrow `lwarpmk print`

-and-

Enter \Rightarrow `lwarpmk html`

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

5.10 Using a glossary

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

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

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

To process the glossary for the print version:

Enter \Rightarrow `lwarpmk printglossary`

To process the glossary for the HTML version:

Enter \Rightarrow `lwarpmk htmlglossary`

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

5.11 Cleaning auxiliary files

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

```
Enter ⇒ lwarpmk clean
```

5.12 Cleaning auxiliary and output files

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

```
Enter ⇒ lwarpmk cleanall
```

5.13 Processing multiple projects in the same directory

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

To create each project:

```
Enter ⇒ pdflatex project_a
```

```
Enter ⇒ pdflatex project_b
```

Each project is given its own configuration file:

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

To compile each project with `lwarkmk`:

```
Enter ⇒ lwarpmk print project_a
```

```
Enter ⇒ lwarpmk html project_b
```

5.14 Using the `make` utility

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

```
lwarpmk pdftohtml [project]
```

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

6 Additional details

6.1 Font and UTF-8 support

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

vector fonts
Computer Modern



`\usepackage{lmodern}`

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

`\usepackage{dejavu}`

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



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

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

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

- For X_YLaTeX or LuaLaTeX:

Pkg `fontspec`

- `fontspec` and font choices

ligatures

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

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

- For `pdflatex`:

Pkg `lmodern`

- `lmodern` or other font-related packages

Pkg `fontenc`

- `fontenc`

Pkg `inputenc`

- `inputenc`

Pkg `newunicodechar`

- `newunicodechar`

File `glyphtounicode`

- `\input glyphtounicode.tex`

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

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

3. `\usepackage{lwarp}` (section 6.2) goes after any of the above, followed by:

4. ... the rest of the preamble and the main document.

6.1.1 Indexes and UTF-8

lwarp uses the xindy program to processes indexes.

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

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

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

6.2 lwarp package loading and options

lwarp supports book, report, and article classes.

Pkg	lwarp	Load the lwarp package immediately after the font and UTF-8 setup commands.
Opt	warpprint	Select the warpprint option to generate print output (default), or the warpHTML option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual pdflatex, etc. When lwarp is loaded in print mode, it creates <code><project>_html.tex</code> , which sets the warpHTML option before calling the user's source code <code><project>.tex</code> . In this way, <code><project>.tex</code> can <code>\usepackage{lwarp}</code> without any options to create a printed version, while <code><project>_html.tex</code> will create an HTML version.
Opt	warpHTML	
Opt	mathsvg	For math display, select mathsvg (default), or mathjax. For more information about the math options, see section 6.12.5.
Opt	mathjax	

See table 5 for the full list of options.

Table 5: Package options

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

6.3 Selecting the operating system

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

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	warppHTML	Anything which is to be done only for HTML5 output is surrounded by a warppHTML environment:
-----	-----------	--

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

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	warppall	Anything which is to be done for any output may be surrounded by a warppall environment. Doing so is optional.
-----	----------	--

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

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

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

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

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

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

6.5 Commands to be placed into the `warpprint` environment

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

- Paragraph formatting: `\parindent` `\parskip`
- 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 \LaTeX 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

`\pagerefPageFor`, which defaults to “see ”, followed by a hyperlink to the appropriate object. Ex: “Sec. 1.23 on page (see sec. 1.23)”. `\pagerefPageFor` may be redefined to “page for ”, empty, etc.

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

\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`.

- | | |
|---|---|
| <code>\maketitle</code> | Use <code>\maketitle</code> just after the <code>\begin{document}</code> , as this will establish the title of the homepage. Optionally, use a <code>titlepage</code> environment instead. |
| Env <code>titlepage</code> | The <code>titlepage</code> environment may be used to hold a custom title page. The <code>titlepage</code> will be set in a <code><div></code> class <code>titlepage</code> , and <code>\printtitle</code> , etc. may be used inside this environment. |
| Env <code>titlingpage</code> | Another form of custom title page, where <code>\maketitle</code> is allowed, and additional information may be included as well. |
| <code>\title</code> | <code>{\langle title \rangle}</code> |
|  | Avoid newlines in the <code>\title</code> ; these will interfere with the file break and CSS detection. Use the <code>\subtitle</code> command instead. In HTML, the title will appear in a heading <code>h1</code> . |
| <code>\author</code> | <code>{\langle author \rangle}</code> |
|  | In <code>\author</code> , use <code>\protect</code> before formatting commands such as <code>\textsc</code> . In HTML, the author will appear in a <code><div></code> class <code>author</code> . <code>\affiliation</code> is a new addition to lwarp. |
| <code>\date</code> | <code>{\langle date \rangle}</code> |
| | <code>\date</code> works as expected. In HTML, this will appear in a <code><div></code> class <code>titledate</code> . |
| <code>\subtitle</code> | <code>{\langle subtitle \rangle}</code> |
| | A new command which sets a subtitle. Newlines are allowed. The default is empty. In HTML, this will appear in a <code><div></code> class <code>subtitle</code> . |
| <code>\published</code> | <code>{\langle published \rangle}</code> |
| | A new command which sets a publisher. The default is empty. In HTML, this will appear in a <code><div></code> class <code>published</code> . |

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

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

6.8 HTML page meta descriptions

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

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

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

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

6.9 HTML page meta author

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

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

```

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

```

6.11 Modifying xindy index processing

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

To customize index processing:

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


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

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

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

```

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

```

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

6.12 Special cases and limitations

6.12.1 Text formatting

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

6.12.2 Cross-references

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

6.12.3 `cleveref` and `varioref` packages

`cleveref` and `varioref` are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used instead. See section 6.6 to redefine the message which is printed for page number references.

6.12.4 Footnotes and page notes

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

6.12.5 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 \LaTeX into the initial PDF file using the current font⁸, then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by \LaTeX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML ALT tag carries the \LaTeX code which generated the math, allowing copy/paste of the \LaTeX math expression into other documents.

SVG image font size

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

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

⁸See section 165 regarding fonts and fractions.

SVG files As currently implemented, each instance of math creates a new SVG file. In text with many references to math variables, this can result in a large number of files with duplicate content. In the future, some method of content-based naming and checksumming may be used to remove the need for duplicate files.

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

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

MathML Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 7 regarding EPUB output with MathJax.

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

Prog **MathJax**

When MathJax is enabled, math is rendered twice:

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

MathJax limitations Limitations when using MathJax include:

Prog **MathJax**


chapter numbers

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



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

subequations




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

footnotes in math	<ul style="list-style-type: none"> Footnotes inside equations are not yet supported while using MathJax.
lateximage	<ul style="list-style-type: none"> Math appearing inside a <code>lateximage</code>, and therefore also inside a <code>Tikz</code> or <code>picture</code> environment, is rendered as SVG math even if MathJax is used in the rest of the document.
siunitx	<ul style="list-style-type: none"> Usage of <code>siunitx</code> inside a math equation is supported via a third-party MathJax extension. While inside a math expression, do not use <code>\SI</code> or <code>\si</code> inside <code>\text</code>, where it will be rendered as normal text.
 siunitx inside an equation	<p>https://github.com/mathjax/MathJax-third-party-extensions/tree/master/siunitx</p>
L ^A T _E X macros	<ul style="list-style-type: none"> MathJax does not automatically support custom L^AT_EX macros, but they may be set up by the user.
custom MathJax macros	For an example of using custom L ^A T _E X macros with MathJax, see page 291.


6.12.6 ntheorem package

 Font control	This conversion is not total. Font control is via CSS, and the custom L ^A T _E X font settings are ignored.
 Equation numbering	<code>ntheorem</code> has a bug with equation numbering in AMS environments when the option <code>thref</code> is used. <code>lwarp</code> does not share this bug, so equations with <code>\split</code> , etc, are numbered correctly with <code>lwarp</code> 's HTML output, but not with the print output. It is recommended to use <code>cleveref</code> instead of <code>ntheorem</code> 's <code>thref</code> option.


6.12.7 Graphics

 graphics vs. graphicx	If using the older <code>graphics</code> syntax, use both optional arguments for <code>\includegraphics</code> . A single optional parameter is interpreted as the newer <code>graphicx</code> syntax. Note that
 viewports	viewports are not supported by <code>warp</code> ; the entire image will be shown.
 \graphicspath	<code>\graphicspath</code> only works for a single directory; all graphics must be in this directory.
units	For <code>\includegraphics</code> , avoid <code>px</code> and <code>%</code> units for width and height, or enclose them inside <code>warpHTML</code> environments. For font-proportional image sizes, use <code>ex</code> or <code>em</code> . For fixed-sized images, use <code>cm</code> , <code>mm</code> , <code>in</code> , <code>pt</code> , or <code>pc</code> . Using the keys <code>width=.5\linewidth</code> , or similar for <code>\textwidth</code> or <code>\textheight</code> to give fixed-sized images proportional to a 6 by 9 inch text area.
options	<code>\includegraphics</code> accepts <code>width</code> and <code>height</code> , <code>origin</code> , <code>rotate</code> and <code>scale</code> , plus a new <code>class</code> 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.

6.12.8 xcolor package

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

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


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

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


6.12.9 Tabular

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

 **`\multirow`**

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

- table for HTML output.
- 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 `warppHTML` environment, if a “misplaced `\noalign`” error occurs, change


```
This & That \endhead
```

 to


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

 and likewise with the other `\end` headings. Keep the `\endfirsthead` row unchanged, as it is still relevant to HTML output.
 - 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 `lwrap`’s tabular handling algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:


```
\warpprintonly{1 & 2 & {\multirow{2}{2cm}{Text}} & 3 \\}
\warppHTMLonly{1 & 2 & \multirow{2}{2cm}{Text} & 3 \\}
```

6.12.10 longtable package

- Longtable `\endhead`, `\endfoot`, and `\endlastfoot` rows are not used for HTML, and these rows should be disabled. Use
- ```
\warpprintonly{row contents}
```
- instead of
- ```
\begin{warpprint} ... \end{warpprint}
```
- Doing so helps avoid “Misplaced `\noalign`.” when using `\begin{warpprint}`.
- Keep the `\endfirsthead` row, which is still relevant to HTML output.
- `\kill` is ignored, place a `\kill` line inside


`\begin{warpprint} ... \end{warpprint}`
 or place it inside `\warpingprintonly`.

6.12.11 Save Boxes

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

6.12.12 Minipages

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.

6.12.13 mdframed package

- support Most basic functionality is supported, including frame background colors and single-border colors and thickness, title and subtitle background colors and borders and thickness, border radius, and shadow. CSS classes are created for `mdframed` environments and frame titles.
- ⚠ loading When used, `lwarp` loads `mdframed` in HTML with `framemethod=none`.
- font For title font, use


```
frametitlefont=\textbf,
```

 instead of


```
frametitlefont=\bfseries,
```

 where `\textbf` must appear just before the comma and will receive the following text as its argument (since the text happens to be between braces in the `mdframed` source). Since `lwarp` does not support `\bfseries` and friends, only one font selection may be made at a time.
- theoremtitlefont `theoremtitlefont` is not supported, since the following text is not in braces in the `mdframed` source.
- footnotes Footnotes are currently placed at the bottom of the HTML page.
- ignored options `userdefinedwidth` and `align` are currently ignored.

6.12.14 float, trivfloat, and/or algorithmicx together

- ⚠ package conflicts If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section [159.1](#).

6.12.15 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`.


6.12.16 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.

6.12.17 floatrow package

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

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

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

6.12.18 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.

6.12.19 verse and memoir

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

```

\begin{warpHTML}

\let\attrib\attribution


\end{warpHTML}

```


Len <code>\leftskip</code> Len <code>\leftmargini</code> Len <code>\TMLvleftskip</code> Len <code>\TMLleftmargini</code>	<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.

6.12.20 siunitx package

Pkg <code>siunitx</code>  <code>per-mode</code>	<p>Do not use <code>per-mode=fraction</code>, which cannot be seen by the final <code>pdftotext</code> conversion.</p>
---	--

6.12.21 newclude package

Pkg <code>newclude</code>  <code>loading</code>	<p><code>newclude</code> modifies <code>\label</code> in a non-adaptive way, so <code>newclude</code> must be loaded before <code>lwarp</code> is loaded.</p>
---	---


Ex:

```

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


```

6.12.22 newtxmath package

Pkg <code>newtxmath</code>  <code>loading sequence</code>	<p>The proper load order is:</p>
---	----------------------------------

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

6.12.23 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}{:~}
```

6.12.24 glossaries package

Pkg `glossaries` `xindy` is required for `glossaries`.

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

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

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

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

6.12.25 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.

6.12.26 enumerate package

Pkg **enumerate** **enumerate** conflicts with **enumitem** if both are loaded at the same time, but **lwarp** does not actually load **enumerate**. While generating HTML, **lwarp** 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

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

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

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

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

To help convert lwarp HTML output to EPUB, add

```
\booltrue{FormatEPUB}
```

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

```
Enter ⇒ lwarpmk cleanall
```


then

```
Enter ⇒ lwarpmk html
```

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


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

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 lwarp projects, set the EPUB conversion software to do a breadth-first search of the files. For Calibre, this option is found in

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

Check the box Add linked files in breadth first order.

 **section breaks** The EPUB-conversion program must also know where the section breaks are located. For a list of lwarp's section headings, see table 6. For example, an `article` class document would break at `\section`, which is mapped to HTML heading level `<h4>`,

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

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

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

Math HTML tags with tag name:

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

`//h:h4` — or — `//h:h3`

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

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

MathJax support

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

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

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

8 Word-processor conversion

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

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

Bool **FormatWordProcessor**

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

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

```
\booltrue{FormatWordProcessor}
```

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

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

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

Bool **HTMLMarkFloats**

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

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

9 Modifying lwarp

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

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

An environment may be converted to a `lateximage` then displayed with an image of the resulting L^AT_EX output. See section 66 for an example of the `picture` environment.

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

9.1 Creating an lwarp version of a package

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

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

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

Each `lwarp-` package should first call either `\LWR@ProvidesPackageDrop` or `\LWR@ProvidesPackagePass`. If `Dropped`, the original print-version package is ignored, and only the `lwarp-` version is used. Use this where the original print version is useless for HTML. If `Passed`, the original package is loaded first, with the user-supplied options, then the `lwarp-` version continues loading as well. See section 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.

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

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

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

9.3 Modifying lwarpmk

Prog **lwarpmk**
File **lwarpmk.lua**

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

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

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

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

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

10 Troubleshooting

10.1 Using the `lwarp.sty` package

Also see:

Section 6.5: Commands to be placed into the `warpprint` environment

Section 6.6: Commands for a successful HTML conversion

Section 6.12: 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 \LaTeX code is correct, before attempting to generate an HTML version.

Missing sections:

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

Missing html files:

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

Missing / incorrect cross-references:

- Use `lwarpmk` again followed by `lwarpmk html` or `lwarpmk print` to compile the document one more time.
- `\nameref` refers to the most recently-used section where the `\label` was defined. If no section has been defined before the `\label`, the link will be empty. Index entries also use `\nameref` and have the same limitation.
- `cleveref` and `varioref` are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used instead. See section 6.6 to redefine the message which is printed for page number references.

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

Use \XeLaTeX or \LuaLaTeX .

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 `l@<floattype>` or `l@<sectiontype>` definition. See `lwarp`’s definitions for examples.

Plain-looking document:

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

Broken fragments of HTML:

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

Changes do not seem to be taking effect:

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

Un-matched conditional compiles:

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

10.1.1 Debug tracing output

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

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

10.2 Compiling the `lwarp.dtx` file

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

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

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

Un-nested environments:

Be sure to properly nest:

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

11 Implementation

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

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

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

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

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

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

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

12 Stack depths

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

Table 6: Section depths and HTML headings

Section	\LaTeX 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

13 Source Code


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

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

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

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

objects Black-colored tags in the left margin are used to identify programming objects such as files, packages, environments, booleans, and counters. Items without a tag are command macros. Each of these also appears in the index as individual entries, and are also listed together under “files”, “packages”, “environments”, “booleans”, and “counters”.

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

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

lwarp source code begins on the following page.

14 Detecting the T_EX Engine — pdf_lat_ex, lua_lat_ex, x_el_ate_x

```

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

```

15 Unicode Input Characters

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

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

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

```

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

```

In PDF_T_EX, preserve upright quotes in verbatim text:

```

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

```

16 Early package requirements

Pkg	etoolbox	Provides \ifbool and other functions.
		22 \RequirePackage{etoolbox}[2011/01/03]
		23 % requires v2.6 for \BeforeBeginEnvironment, etc.
Pkg	ifplatform	Provides \ifwindows to try to automatically detect Windows OS.
		24 \RequirePackage{ifplatform}% sense op-system platform
Pkg	comment	Provides conditional code blocks.
		25 \RequirePackage{comment}
		26 \excludecomment{testing}

17 Operating-System portability

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

17.1 Common portability code

Bool usingOSWindows Set if the OSWindows option is used.

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

17.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

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

17.3 MS-Windows

For MS-Windows:

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

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

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

```

35 \ifwindows
36 \LWR@setOSWindows
37 \fi

```

18 Package options

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

```

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

```

Bool **warpingprint**

Bool **warpingHTML**

Bool **mathjax**

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

```

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

```

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

Only process the contents if producing printed output.

```

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

```

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

Only process the contents if producing HTML output.

```

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

```

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

```

45 \includecomment{warpall}

```

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

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

```

46 \DeclareVoidOption{warpprint}{%
47 \PackageInfo{lwarp}{Using option 'warpprint'}
48 \includecomment{warpprint}%

```

```

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

```

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

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

```

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

```

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

```

60 \DeclareVoidOption{mathsvg}{%
61 \PackageInfo{lwarp}{Using option 'mathsvg'}
62 \boolfalse{mathjax}%
63 }

```

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

```

64 \DeclareVoidOption{mathjax}{%
65 \PackageInfo{lwarp}{Using option 'mathjax'}
66 \booltrue{mathjax}%
67 }

```

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

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

```

68 \DeclareStringOption[\jobname]{BaseJobname}

```

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

```

69 \DeclareStringOption[english]{IndexLanguage}

```


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

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

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

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

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

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

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

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

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

```
79 \DeclareStringOption[] {HomeHTMLFilename}
```

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

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

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

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

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

```

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

```

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

```

87 \excludecomment{LWR@createlwarpmk}

```

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

```

88 \ProcessKeyvalOptions*\relax

```

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

```

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

```

Defaults unless already over-ridden by the user:

```

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

```

19 Misplaced packages

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

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

`lwarp-<packagename>.sty`

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

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

```

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

```

```

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

```

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

```

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

```

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

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

```

113 \newcommand*{\LWR@loadnever}[2]{%
114 \PackageError{lwarp}
115 {Package #1 does not work with lwarp's HTML conversion.
116 Please use the #2 package instead}
117 {Package #1 conflicts with lwarp in some way, but package #2 probably will work instead.}
118 }

```

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

The following packages must be loaded after lwarp:

```

119 \LWR@loadafter{abstract}
120 \LWR@loadafter{afterpage}
121 \LWR@loadafter{algorithmicx}
122 \LWR@loadafter{alltt}
123 \LWR@loadafter{amsthm}
124 \LWR@loadafter{bookmark}
125 \LWR@loadafter{booktabs}
126 \LWR@loadafter{ccaption}
127 \LWR@loadafter{changepage}
128 \LWR@loadafter{cutwin}
129 \LWR@loadafter{dcolumn}
130 \LWR@loadafter{draftwatermark}
131 \LWR@loadafter{ellipsis}
132 \LWR@loadafter{emptypage}
133 \LWR@loadafter{enumerate}
134 \LWR@loadafter{epigraph}
135 \LWR@loadafter{eso-pic}

```

```
136 \LWR@loadafter{everypage}
137 \LWR@loadafter{extramarks}
138 \LWR@loadafter{fancyhdr}
139 \LWR@loadafter{floatrow}
140 \LWR@loadafter{float}
141 \LWR@loadafter{floatflt}
142 \LWR@loadafter{ftnright}
143 \LWR@loadafter{geometry}
144 \LWR@loadafter{glossaries}
145 % \LWR@loadafter{graphics}% pre-loaded by xunicode
146 % \LWR@loadafter{graphicx}% pre-loaded by xunicode
147 \LWR@loadafter{hyperref}
148 \LWR@loadafter{indentfirst}
149 \LWR@loadafter{keyfloat}
150 \LWR@loadafter{layout}
151 \LWR@loadafter{letterspace}
152 \LWR@loadafter{lettrine}
153 \LWR@loadafter{lips}
154 \LWR@loadafter{listings}
155 \LWR@loadafter{longtable}
156 \LWR@loadafter{lscape}
157 \LWR@loadafter{ltcaption}
158 \LWR@loadafter{marginfix}
159 \LWR@loadafter{marginnote}
160 \LWR@loadafter{mcaption}
161 \LWR@loadafter{mdframed}
162 \LWR@loadafter{microtype}
163 \LWR@loadafter{mparhack}
164 %\LWR@loadafter{multicol}% loaded by ltxdoc
165 \LWR@loadafter{multirow}
166 \LWR@loadafter{nameref}
167 \LWR@loadafter{needspace}
168 \LWR@loadafter{newtxmath}
169 \LWR@loadafter{nextpage}
170 \LWR@loadafter{nowidow}
171 \LWR@loadafter{ntheorem}
172 \LWR@loadafter{pagenote}
173 \LWR@loadafter{parskip}
174 \LWR@loadafter{placeins}
175 \LWR@loadafter{ragged2e}
176 \LWR@loadafter{rotating}
177 \LWR@loadafter{setspace}
178 \LWR@loadafter{showidx}
179 \LWR@loadafter{showkeys}
180 \LWR@loadafter{sidecap}
181 \LWR@loadafter{sidenotes}
182 \LWR@loadafter{soul}
183 \LWR@loadafter{subfig}
184 \LWR@loadafter{tabularx}
185 \LWR@loadafter{tabulary}
```

```

186 \LWR@loadafter{textpos}
187 \LWR@loadafter{theorem}
188 \LWR@loadafter{threeparttable}
189 \LWR@loadafter{tikz}
190 \LWR@loadafter{titleps}
191 \LWR@loadafter{titlesec}
192 \LWR@loadafter{titletoc}
193 \LWR@loadafter{tocloft}
194 \LWR@loadafter{trivfloat}
195 \LWR@loadafter{ulem}
196 \LWR@loadafter{varioref}
197 \LWR@loadafter{verse}
198 \LWR@loadafter{wallpaper}
199 \LWR@loadafter{wrapfig}
200 \LWR@loadafter{xcolor}
201 \LWR@loadafter{xfrac}

```

20 Required packages

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

In the document preamble, create a **warpprint** environment, and place inside it any of the following packages which are required and which are labeled as “Print: OK to Load in a **warpprint** environment”. Those packages which are labeled as “Print: Pre-Loaded” need not be placed into the document preamble.

for HTML & PRINT: 202 \begin{warpall}

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

Detects XeTeX and LuaTeX:

```

203 \RequirePackage{iftex}
204 \newif\ifxetexorluatex
205 \ifXeTeX
206     \xetexorluatextrue
207 \else
208     \ifLuaTeX
209         \xetexorluatextrue
210     \else
211         \xetexorluatexfalse
212     \fi
213 \fi

214 \end{warpall}

```

for HTML output: 215 \begin{warpHTML}

```
216 \ifxetexorluatex
217 % ^^A \usepackage[no-math]{fontspec}
```

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

```
218 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
219 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
220 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
221 \else
```

pdf_lat_ex only: Only pre-loaded if pdf_lat_ex 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.

```
222 \RequirePackage {microtype}
223
224 \microtypesetup{
225 protrusion=false,
226 expansion=false,
227 tracking=false,
228 kerning=false,
229 spacing=false}
230
231 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}
232
233 \fi
234
235 \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: 234 `\begin{warpHTML}`
235 `\RequirePackage[paperheight=190in,paperwidth=20in,%`
236 `left=2in,right=12in,%`
237 `top=1in,bottom=1in,%`
238 `]{geometry}`
239 `\@twosidefalse`
240 `\@mparswitchfalse`
241 `\end{warpHTML}`

for HTML & PRINT: 242 `\begin{warpall}`

Pkg `xparse`

L^AT_EX3 command argument parsing

243 `\RequirePackage{xparse}`

244 `\end{warpall}`

for HTML output: 245 `\begin{warpHTML}`

Pkg `expl3`

L^AT_EX3 programming

246 `\RequirePackage{expl3}`

Pkg `getttitlestring`

Used to emulate `\nameref`.

247 `\RequirePackage{getttitlestring}`

Pkg `everyhook`

`everyhook` is used to patch paragraph handling.

248 `\RequirePackage{everyhook}`

249 `\end{warpHTML}`

for HTML & PRINT: 250 `\begin{warpall}`

Pkg `fancyvrb`

Used for Verbatim, verse.

251 `\RequirePackage{fancyvrb}`

252 `\end{warpall}`

for HTML output: 253 \begin{warpHTML}

Pkg xifthen

254 \RequirePackage{xifthen}

Pkg xstring

255 \RequirePackage{xstring}

Pkg makeidx

256 \RequirePackage{makeidx}

257 \makeindex

Pkg calc

258 \RequirePackage{calc}

Pkg refcount

259 \RequirePackage{refcount}

Pkg newfloat

260 \RequirePackage{newfloat}

Pkg caption

261 \RequirePackage{caption}

Pkg enumitem

enumitem is patched to support \newlist with HTML.

262 \RequirePackage{enumitem}

263 \setlist[itemize]{leftmargin=0em}

264 \setlist[enumerate]{leftmargin=0em}

265 \setlist[description]{leftmargin=0em}

266 \end{warpHTML}

for HTML & PRINT: 267 \begin{warpall}

Pkg titling

Used for \maketitle and the title page. See section 48.

268 \RequirePackage{titling}


```
269 \end{warpall}
```

for HTML output:

```
270 \begin{warpHTML}
```

Pkg **zref**

Used for cross-references.

```
271 \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.

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

```
273 \RequirePackage{amsmath}
```

Pkg **environ**

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

```
274 \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.

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

```
\pagestyle {\langle style \rangle}
```

```
276 \let\LWR@origpagestyle\pagestyle
```

```
277 \renewcommand*\pagestyle}[1]{}%
```

```
\thispagestyle {\langle style \rangle}
```

```
278 \let\LWR@origthispagestyle\thispagestyle
```

```
279 \renewcommand*\thispagestyle}[1]{}%
```

`\pagenumbering` $\{\langle commands \rangle\}$

```
280 \let\LWR@origpagenumbering\pagenumbering
281 \renewcommand*\pagenumbering[1]{}
```

Pkg `xfrac`

Patched for HTML use. See section 165.

```
282 \RequirePackage{xfrac}
```

Used to convert lengths for image width/height options.

```
283 \RequirePackage{printlen}
```

```
284 \end{warpHTML}
```

21 Loading packages

for HTML output: 285 `\begin{warpHTML}`

Remember the original `\RequirePackage`:

```
286 \let\LWR@origRequirePackage\RequirePackage
```

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

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

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

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

```
288 \newcommand*\LWR@findword[3][,]{%
289   \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
290 }
```

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

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

Find the n 'th package name from the list:

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

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

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

See if the package name was found:

```
294 \IfStrEq{\LWR@strresulttwo}{}%
295 {}% no filename
296 {% yes filename
```

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

```
297 \IfFileExists{lwarp-\LWR@strresulttwo.sty}%
298 {% lwarp-* file found
299 \StrSubstitute%
300 {\LWR@requirepackagenames}%
301 {\LWR@strresult}%
302 {lwarp-\LWR@strresulttwo}[\LWR@requirepackagenames]%
303 }%
304 {}% no lwarp-* file
305 }% yes filename
306 }
```

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

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

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

Redirect up to nine names:

```
308 \renewcommand*\LWR@requirepackagenames}{#2}
309 \LWR@lookforpackagename{1}
310 \LWR@lookforpackagename{2}
311 \LWR@lookforpackagename{3}
312 \LWR@lookforpackagename{4}
313 \LWR@lookforpackagename{5}
314 \LWR@lookforpackagename{6}
315 \LWR@lookforpackagename{7}
316 \LWR@lookforpackagename{8}
317 \LWR@lookforpackagename{9}
```

\RequirePackage depending on the options and version:

```

318 \IfValueTF{#1}
319 {% options given
320 \IfValueTF{#3}% version given?
321 {\LWR@origRequirePackage{#1}{\LWR@requirepackagenames}[#3]}
322 {\LWR@origRequirePackage{#1}{\LWR@requirepackagenames}}
323 }
324 {% no options given
325 \IfValueTF{#3}% version given?
326 {\LWR@origRequirePackage{\LWR@requirepackagenames}[#3]}
327 {\LWR@origRequirePackage{\LWR@requirepackagenames}}
328 }
329 }
330 \let\usepackage\RequirePackage

```

\LWR@ProvidesPackagePass {<pkgname>} [<version>]

Uses the original package, including options.

```

331 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
332 \PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
333 \IfValueTF{#2}
334 {\ProvidesPackage{lwarp-#1}[#2]}
335 {\ProvidesPackage{lwarp-#1}}
336 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
337 \ProcessOptions\relax
338
339 \IfValueTF{#2}
340 {\LWR@origRequirePackage{#1}[#2]}
341 {\LWR@origRequirePackage{#1}}
342 }

```

\LWR@ProvidesPackageDrop {<pkgname>} [<version>]

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

```

343 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
344 \PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
345 \IfValueTF{#2}
346 {\ProvidesPackage{lwarp-#1}[#2]}
347 {\ProvidesPackage{lwarp-#1}}
348 \DeclareOption*{}
349 \ProcessOptions\relax
350 }

351 \end{warpHTML}

```

22 Copying a file

for HTML output: 352 \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.

```

353 \newcommand*{\LWR@copyfile}[2]{%
354 \newwrite\copyfile % open the file to write to
355 \immediate\openout\copyfile=#2
356 \newread\file % open the file to read from
357 \openin\file=#1
358 \begingroup\endlinechar=-1
359 \makeatletter
360 \loop\unless\ifeof\file
361 \read\file to\fileline % Read one line and store it into \fileline
362 % \fileline\par % print the content into the pdf
363 % print the content:
364 \immediate\write\copyfile{\unexpanded\expandafter{\fileline}}%
365 \repeat
366 \closeout\copyfile
367 \endgroup
368 }

369 \end{warpHTML}

```

23 Debugging messages

370 \begin{warpall}

Bool LWR@tracinglwarp True if tracing is turned on.

371 \newbool{LWR@tracinglwarp}

\tracinglwarp Turns on the debug tracing messages.

372 \newcommand{\tracinglwarp}{\booltrue{LWR@tracinglwarp}}

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

```

373 \newcommand{\LWR@traceinfo}[1]{%
374 \ifbool{LWR@tracinglwarp}%
375 {%

```

```

376 \typeout{*** lwarp: #1}%
377 % \PackageInfo{lwarp}{#1 : }%
378 }%
379 {}%
380 }

```

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

```

381 \newbool{HTMLDebugComments}
382 \boolfalse{HTMLDebugComments}

```

24 HTML-conversion output modifications

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

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

```

383 \newbool{FormatEPUB}
384 \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.

```

385 \newbool{FormatWordProcessor}
386 \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.)

```

387 \newbool{HTMLMarkFloats}
388 \booltrue{HTMLMarkFloats}

```

```

389 \end{warpall}

```

25 Remembering original formatting macros

for HTML output: 390 \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.

```

391 \let\LWR@origtextit\textit
392 \let\LWR@origtextbf\textbf
393 \let\LWR@origtexttt\texttt
394 \let\LWR@origtextsc\textsc
395 \let\LWR@origtextsf\textsf
396 \let\LWR@origtextrm\textrm
397 \let\LWR@origbfseries\bfseries
398 \let\LWR@origrmfamily\rmfamily
399 \let\LWR@origttfamily\ttfamily
400 \let\LWR@orignormalfont\normalfont
401
402 \let\LWR@origraggedright\raggedright
403 \let\LWR@origonecolumn\onecolumn
404
405 \let\LWR@origtextsuperscript\textsuperscript
406 \let\LWR@origtextsubscript\textsubscript
407
408 \let\LWR@origscriptsize\scriptsize
409
410 \let\LWR@orignewpage\newpage
411
412 \let\LWR@origminipage\minipage
413 \let\LWR@origendminipage\endminipage
414
415 \let\LWR@orignewline\newline
416
417 \let\LWR@origitem\item
418
419 \let\LWR@origpar\par
420
421
422 \let\LWR@origfootnote\footnote
423 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
424
425 \let\LWR@origclearpage\clearpage
426 \let\clearpage\relax
427 \let\cleardoublepage\relax
428 \end{warpHTML}

```

26 Configuration Files

```
429 \begin{warpprint}
430 \typeout{lwarp: generating configuration files}
431 \end{warpprint}
```

26.1 project_html.tex

File `project_html.tex` Used to allow an HTML version of the document to exist alongside the print version.

Only write `\jobname_html.tex` if generating the print version.

```
432 \begin{warpprint}
433 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
434 \immediate\openout{LWR@file}=\jobname_html.tex
435 \immediate\write{LWR@file}{%
436 \detokenize{\PassOptionsToPackage}%
437 {warpHTML,BaseJobname=\jobname}{lwarp}%
438 }
439 \immediate\write{LWR@file}{%
440 \detokenize{\input}\string{\jobname.tex}\string }%
441 }
442 \immediate\closeout{LWR@file}
443 \end{warpprint}
```

26.2 lwarpmk.conf

File `lwarpmk.conf` `lwarpmk.conf` is automatically (re-)created by the `lwarp` package when executing `pdflatex <project.tex>`, or similar for `xelatex` or `lualatex`, in print-document generation mode, which is the default unless the `warpHTML` option is given. `lwarpmk.conf` is then used by the utility `lwarpmk`.

An example `lwarpmk.conf`:

```
opsystem = "Unix"    -- or "Windows"
latexname = "pdflatex" -- or "lualatex" or "xelatex"
sourcename = "projectname" -- your .tex source
homehtmlfilename = "index" -- or "projectname"
htmlfilename = ""    -- or "projectname" if numbered HTML files
```

for PRINT output:

```
444 \begin{warpprint}
445 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
446 \immediate\openout{LWR@file}=lwarpmk.conf
447 \ifbool{usingOSWindows}{
```



```

448 \immediate\write\LWR@file{opsystem = "Windows"}
449 }{
450 \immediate\write\LWR@file{opsystem = "Unix"}
451 }
452 \ifPDFTeX
453 \immediate\write\LWR@file{latexname = "pdflatex"}
454 \fi
455 \ifXeTeX
456 \immediate\write\LWR@file{latexname = "xelatex"}
457 \fi
458 \ifLuaTeX
459 \immediate\write\LWR@file{latexname = "lualatex"}
460 \fi
461 \immediate\write\LWR@file{sourcename = "\jobname"}
462 \immediate\write\LWR@file{%
463 homehtmlfilename = "\HomeHTMLFilename"%
464 }
465 \immediate\write\LWR@file{htmlfilename = "\HTMLFilename"}
466 \immediate\write\LWR@file{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
467 \immediate\write\LWR@file{language = "\LWR@IndexLanguage"}
468 \immediate\write\LWR@file{xdyfile = "\LWR@xdyFilename"}
469 \immediate\closeout\LWR@file
470 \end{warpprint}

```

26.3 project.lwarpmkconf

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

```

471 \begin{warpprint}
472 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
473 \immediate\openout\LWR@file=\jobname.lwarpmkconf
474 \ifbool{usingOSWindows}{
475 \immediate\write\LWR@file{opsystem = "Windows"}
476 }{
477 \immediate\write\LWR@file{opsystem = "Unix"}
478 }
479 \ifPDFTeX
480 \immediate\write\LWR@file{latexname = "pdflatex"}
481 \fi
482 \ifXeTeX
483 \immediate\write\LWR@file{latexname = "xelatex"}
484 \fi
485 \ifLuaTeX
486 \immediate\write\LWR@file{latexname = "lualatex"}
487 \fi
488 \immediate\write\LWR@file{sourcename = "\jobname"}
489 \immediate\write\LWR@file{%
490 homehtmlfilename = "\HomeHTMLFilename"%

```

```

491 }
492 \immediate\write\LWR@file{htmlfilename = "\HTMLFilename"}
493 \immediate\write\LWR@file{latexmk = "\ifbool{LWR@latexmk}{true}{false}}
494 \immediate\write\LWR@file{language = "\LWR@IndexLanguage"}
495 \immediate\write\LWR@file{xdyfile = "\LWR@xdyFilename"}
496 \immediate\closeout\LWR@file
497 \end{warpprint}

```

26.4 lwarp.css

File `lwarp.css` This is the base CSS layer used by lwarp.

This must be present both when compiling the project and also when distributing the HTML files.

```

498 \begin{warpprint}
499 \begin{VerbatimOut}{lwarp.css}
500 /*
501  CSS stylesheet for the LaTeX lwarp package
502  Copyright 2016-2017 Brian Dunn -- BD Tech Concepts LLC
503 */
504
505
506 /* a fix for older browsers: */
507 header, section, footer, aside, nav, main,
508     article, figure { display: block; }
509
510
511 A:link {color:#000080 ; text-decoration: none ; }
512 A:visited {color:#800000 ; }
513 A:hover {color:#000080 ; text-decoration: underline ;}
514 A:active {color:#800000 ; }
515
516 a.tocpart {display: inline-block ; margin-left: 0em ;
517     font-weight: bold ;}
518 a.tocchapter {display: inline-block ; margin-left: 0em ;
519     font-weight: bold ;}
520 a.tocsection {display: inline-block ; margin-left: 1em ;
521     text-indent: -.5em ; font-weight: bold ; }
522 a.tocsubsection {display: inline-block ; margin-left: 2em ;
523     text-indent: -.5em ; }
524 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
525     text-indent: -.5em ; }
526 a.tocparagraph {display: inline-block ; margin-left: 4em ;
527     text-indent: -.5em ; }
528 a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
529     text-indent: -.5em ; }

```

```
530 a.tocfigure {margin-left: 0em}
531 a.tocsubfigure {margin-left: 2em}
532 a.toctable {margin-left: 0em}
533 a.tocsubtable {margin-left: 2em}
534 a.toctheorem {margin-left: 0em}
535 a.toclstlisting {margin-left: 0em}
536
537
538 body {
539     font-family: "DejaVu Serif", "Bitstream Vera Serif",
540         "Lucida Bright", Georgia, serif;
541     background: #FAF7F4 ;
542     color: black ;
543     margin:0em ;
544     padding:0em ;
545     font-size: 100% ;
546     line-height: 1.2 ;
547 }
548
549 p {margin: 1.5ex 0em 1.5ex 0em ;}
550
551 /* Holds a section number to add space between it and the name */
552 span.sectionnumber { margin-right: .6em }
553
554 /* Inserted in front of index lines */
555 span.indexitem {margin-left: 0em}
556 span.indexsubitem {margin-left: 2em}
557 span.indexsubsubitem {margin-left: 4em}
558
559 div.hidden { display: none ; }
560
561 kbd {
562     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
563         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
564         "Courier New", monospace;
565     font-size: 100% ;
566 }
567
568 span.strong { font-weight: bold; }
569
570 span.textmd { font-weight: normal; }
571
572 span.textsc { font-variant: small-caps; }
573
574 span.textup { font-variant: normal; }
575
576 span.textrm {
577     font-family: "DejaVu Serif", "Bitstream Vera Serif",
578         "Lucida Bright", Georgia, serif;
579 }
```

```
580
581 span.textsf {
582     font-family: "DejaVu Sans", "Bitstream Vera Sans",
583         Geneva, Verdana, sans-serif ;
584 }
585
586 span.attribution {
587     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
588 }
589
590 span.citetitle {
591     margin-left: 1em ; font-size: 80% ; font-style: oblique;
592 }
593
594 span.poemtitle {
595     font-size: 120% ; font-weight: bold;
596 }
597
598 blockquote {
599     margin-left: 0px ;
600     margin-right: 0px ;
601 }
602
603 blockquote p {
604     line-height: 1.5;
605     text-align: left ;
606     font-size: .85em ;
607     margin-left: 3em ;
608     margin-right: 3em ;
609 }
610
611 blockquotation {
612     margin-left: 0px ;
613     margin-right: 0px ;
614 }
615
616 blockquotation p {
617     line-height: 1.5;
618     text-align: left ;
619     font-size: .85em ;
620     margin-left: 3em ;
621     margin-right: 3em ;
622 }
623
624 div.epigraph {
625     line-height: 1.2;
626     text-align: left ;
627     padding: 3ex 1em 0ex 1em ;
628     /* margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
629     margin: 3ex 1em 3ex auto ; /* Epigraph to the right */

```

```
630 /*      margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
631      font-size: .85em ;
632      max-width: 27em ;
633 }
634
635
636
637 div.epigraphsource{
638     text-align:right ;
639     margin-left:auto ;
640 /*      max-width: 50% ; */
641     border-top: 1px solid #A0A0A0 ;
642     padding-bottom: 3ex ;
643     line-height: 1.2;
644 }
645
646 div.epigraph p { padding: .5ex ; margin: 0ex ;}
647 div.epigraphsource p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
648
649
650 /* lettrine package: */
651 span.lettrine { font-size: 3ex ; float: left ; }
652 span.lettrinetext { font-variant: small-caps ; }
653
654 /* ulem and soul packages: */
655 span.uline {
656     text-decoration: underline ;
657     text-decoration-skip ;
658 }
659
660 span.uuline {
661     text-decoration: underline ;
662     text-decoration-skip ;
663     text-decoration-style: double ;
664 }
665
666 span.uwave {
667     text-decoration: underline ;
668     text-decoration-skip ;
669     text-decoration-style: wavy ;
670 }
671
672 span.sout {
673     text-decoration: line-through ;
674 }
675
676 span.xout {
677     text-decoration: line-through ;
678 }
679
```

```
680 span.dashuline {
681     text-decoration: underline ;
682     text-decoration-skip ;
683     text-decoration-style: dashed ;
684 }
685
686 span.dotuline {
687     text-decoration: underline ;
688     text-decoration-skip ;
689     text-decoration-style: dotted ;
690 }
691
692 span.letterspacing { letter-spacing: .2ex ; }
693
694 span.capsspacing {
695     font-variant: small-caps ;
696     letter-spacing: .1ex ;
697 }
698
699 span.highlight { background: #F8E800 ; }
700
701
702
703
704 html body {
705     margin: 0 ;
706     line-height: 1.2;
707 }
708
709
710 body div {
711     margin: 0ex;
712 }
713
714
715 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
716 {
717     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
718         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
719         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
720         "Times New Roman", serif;
721     font-style: normal ;
722     font-weight: bold ;
723     text-align: left ;
724 }
725
726 h1 { /* title of the entire website, used on each page */
727     text-align: center ;
728     font-size: 2.5em ;
729     padding: .4ex 0em 0ex 0em ;
```

```
730 }
731 h2 { font-size: 2.25em }
732 h3 { font-size: 2em }
733 h4 { font-size: 1.75em }
734 h5 { font-size: 1.5em }
735 h6 { font-size: 1.25em }
736 span.paragraph {font-size: 1em ; font-variant: normal ;
737     margin-right: 1em ; }
738 span.subparagraph {font-size: 1em ; font-variant: normal ;
739     margin-right: 1em ; }
740
741
742
743 /* Title of the file */
744 h1 {
745     margin: 0ex 0em 0ex 0em ;
746     line-height: 1.3;
747     text-align: center ;
748 }
749
750 /* Part */
751 h2 {
752     margin: 1ex 0em 1ex 0em ;
753     line-height: 1.3;
754     text-align: center ;
755 }
756
757 /* Chapter */
758 h3 {
759     margin: 3ex 0em 1ex 0em ;
760     line-height: 1.3;
761 }
762
763 /* Section */
764 h4 {
765     margin: 3ex 0em 1ex 0em ;
766     line-height: 1.3;
767 }
768
769 /* Sub-Section */
770 h5 {
771     margin: 3ex 0em 1ex 0em ;
772     line-height: 1.3;
773 }
774
775 /* Sub-Sub-Section */
776 h6 {
777     margin: 3ex 0em 1ex 0em ;
778     line-height: 1.3;
779 }
```

```
780
781
782 div.titlepage {
783   text-align: center ;
784 }
785
786 .footnotes {
787   font-size: .85em ;
788   margin: 3ex 1em 0ex 1em ;
789   padding-bottom: 1ex ;
790 border-top: 1px solid silver ;
791 }
792
793 .marginpar {
794   max-width:50%;
795   float:right;
796   text-align:left;
797   margin: 1ex 0.5em 1ex 1em ;
798   padding: 1ex 0.5em 1ex 0.5em ;
799   font-size: 85% ;
800   border-top: 1px solid silver ;
801   border-bottom: 1px solid silver ;
802   overflow-x: auto;
803 }
804
805 .marginpar br { margin-bottom: 2ex ; }
806
807 div.marginblock {
808   max-width:50%;
809   float:right;
810   text-align:left;
811   margin: 1ex 0.5em 1ex 1em ;
812   padding: 1ex 0.5em 1ex 0.5em ;
813   overflow-x: auto;
814 }
815
816 div.marginblock div.minipage {
817   display: block ;
818   margin: 0pt auto 0pt auto ;
819 }
820
821 div.marginblock div.minipage p { font-size: 85%}
822
823 div.marginblock br { margin-bottom: 2ex ; }
824
825
826 section.textbody div.footnotes{
827   margin: 3ex 0em 0ex 0em ;
828   border-bottom: 2px solid silver ;
829 }
```



```
830
831 .footnoteheader {
832     border-top: 2px solid silver ;
833     margin-top: 3ex ;
834     padding-top: 1ex ;
835     font-weight: bold ;
836 }
837
838 .mpfootnotes {
839     text-align: left ;
840     font-size: .85em ;
841     margin-left: 1em ;
842     border-top: 1px solid silver ;
843 }
844
845 /* Remove footnote top border in the title page. */
846 div.titlepage div.mpfootnotes {
847     border-top: none ;
848 }
849
850
851
852 ol {
853     margin: 1ex 1em 1ex 0em;
854     line-height: 1.2;
855 }
856
857 ul, body dir, body menu {
858     margin: 1ex 1em 1ex 0em;
859     line-height: 1.2;
860 }
861
862 li { margin: 0ex 0em 1ex 0em; }
863
864 html {
865     margin: 0;
866     padding: 0;
867 }
868
869 .programlisting {
870     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
871         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
872         "Courier New", monospace;
873     margin: 1ex 0ex 1ex 0ex ;
874     padding: .5ex 0pt .5ex 0pt ;
875     overflow-x: auto;
876 }
877
878 section.textbody>pre.programlisting {
879     border-top: 1px solid silver ;
```

```
880 border-bottom: 1px solid silver ;
881 }
882
883
884 .inlineprogramlisting {
885     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
886         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
887         "Courier New", monospace;
888     overflow-x: auto;
889 }
890
891
892 div.abstract {
893     margin: 2em 5% 2em 5% ;
894     padding: 1ex 1em 1ex 1em ;
895     /* font-weight: bold ; */
896     font-size: 90% ;
897 }
898
899 div.abstract dl {line-height:1.5;}
900 div.abstract dt {color:#304070;}
901
902 div.abstracttitle{
903     font-family: "URW Classico", Optima, "Linux Biolinum O",
904         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
905         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
906     font-weight:bold;
907     font-size:1.25em;
908     text-align: center ;
909 }
910
911 span.abstracrunintitle{
912     font-family: "URW Classico", Optima, "Linux Biolinum O",
913         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
914         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
915     font-weight:bold;
916 }
917
918
919 .verbatim {
920     overflow-x: auto ;
921 }
922
923 .alltt {
924     overflow-x: auto ;
925 }
926
927
928 .bverbatim {
929     margin: 1ex 0pt 1ex 0pt ;
```

```
930     padding: .5ex Opt .5ex Opt ;
931     overflow-x: auto ;
932 }
933
934 .lverbatim {
935     margin: 1ex Opt 1ex Opt ;
936     padding: .5ex Opt .5ex Opt ;
937     overflow-x: auto ;
938 }
939
940 .fancyvrb {
941     font-size:.85em ;
942     margin: 3ex Opt 3ex Opt
943 }
944
945 .fancyvrblabel {
946     font-weight:bold;
947     text-align: center ;
948 }
949
950
951 .verse {
952     font-family: "Linux Libertine Mono O", "Lucida Console",
953                 "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
954                 "Liberation Mono", "FreeMono", "Andale Mono",
955                 "Nimbus Mono L", "Courier New", monospace;
956     margin-left: 1em ;
957 }
958
959
960 div.singlespace { line-height: 1.2 ; }
961 div.onehalfspace { line-height: 1.5 ; }
962 div.doublespace { line-height: 2 ; }
963
964
965
966
967
968 /* Minipage environments, vertically aligned to top, center, bottom: */
969 .minipage {
970     /* display: inline-block ; */
971     /* Mini pages which follow each other will be tiled. */
972     margin: .25em .25em .25em .25em;
973     padding: .25em .25em .25em .25em;
974     display: inline-flex;
975     flex-direction: column ;
976     overflow: auto;
977 }
978
979 /* Paragraphs in the flexbox did not collapse their margins. */
```

```
980 /* Have not yet researched this. */
981 .minipage p {margin: .75ex 0em .75ex 0em ;}
982
983
984
985 .framebox {
986     margin: 0ex ;
987     padding: 0ex ;
988     border: 1px solid black;
989     border-radius: 0px ;
990     padding: .3ex .2em 0ex .2em ;
991     margin: .1ex ;
992     display: inline-block ;
993 }
994
995
996 .mdframed {
997 /*     padding: 0ex ; */
998 /*     border: 1px solid black; */
999 /*     border-radius: 0px ; */
1000     padding: 0ex ;
1001     margin: 3ex 5% 3ex 5% ;
1002 /*     display: inline-block ; */
1003 }
1004
1005 .mdframed p { padding: 0ex .5em 0ex .5em ; }
1006
1007 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
1008
1009 .mdframedtitle {
1010     padding: .5em ;
1011     display: block ;
1012     font-size: 130%
1013 }
1014
1015 .mdframedsubtitle {
1016     padding: 0ex .5em 0ex .5em ;
1017     display: block ;
1018     font-size: 115% ;
1019 }
1020
1021 .mdframedsubsubtitle {
1022     padding: 0ex .5em 0ex .5em ;
1023     display: block ;
1024 }
1025
1026 .mdtheorem {
1027     padding: 0ex .5em 0ex .5em ;
1028     margin: 3ex 5% 3ex 5% ;
1029 /*     display: inline-block ; */
```

```
1030 }
1031
1032
1033 /* framed package */
1034 .framed {
1035     margin: 3ex 0em 3ex 0em ;
1036     border: 1px solid black;
1037     border-radius: 0px ;
1038     padding: .3ex 1em 0ex 1em ;
1039     display: block ;
1040 }
1041
1042 .snugframed {
1043     margin: 3ex 0em 3ex 0em ;
1044     border: 1px solid black;
1045     border-radius: 0px ;
1046     display: block ;
1047 }
1048
1049 .framedleftbar {
1050     margin: 3ex 0em 3ex 0em ;
1051     border-left: 3pt solid black;
1052     border-radius: 0px ;
1053     padding: .3ex .2em .3ex 1em ;
1054     display: block ;
1055 }
1056
1057 .framedtitle {
1058     margin: 0em ;
1059     padding: 0em ;
1060     font-size: 130%
1061 }
1062
1063 .framedtitle p { padding: .3em }
1064
1065
1066
1067 dl {
1068     margin: 1ex 2em 1ex 0em;
1069     line-height: 1.3;
1070 }
1071
1072 dl dt {
1073     margin-top: 1ex;
1074     font-weight: bold;
1075 }
1076
1077 dl dd p { margin-top: 0em; }
1078
1079
```

```
1080 nav.toc, nav.lof, nav.lot, nav.lol, nav.lothm {
1081     font-family: "URW Classico", Optima, "Linux Biolinum O",
1082         "DejaVu Sans", "Bitstream Vera Sans",
1083         Geneva, Verdana, sans-serif ;
1084     margin-bottom: 4ex ;
1085 }
1086
1087 nav.toc p, nav.lof p, nav.lot p, nav.lol p, nav.lothm p {
1088     line-height: 1.2 ;
1089     margin-top:.5ex ;
1090     margin-bottom:.5ex;
1091     font-size: .9em ;
1092 }
1093
1094
1095
1096 img, img.hyperimage, img.borderimage {
1097     max-width: 600px;
1098     border: 1px solid silver;
1099     box-shadow: 3px 3px 3px #808080 ;
1100     padding: .5% ;
1101     margin: .5% ;
1102     background: none ;
1103 }
1104
1105 img.inlineimage{
1106     padding: 0px ;
1107     box-shadow: none ;
1108     border: none ;
1109     background: none ;
1110     margin: 0px ;
1111     display: inline-block ;
1112     border-radius: 0px ;
1113 }
1114
1115 img.logoimage{
1116     max-width: 300px ;
1117     box-shadow: 3px 3px 3px #808080 ;
1118     border: 1px solid black ;
1119     background:none ;
1120     padding:0 ;
1121     margin:.5ex ;
1122     border-radius: 10px ;
1123 }
1124
1125
1126 .section {
1127 /*
1128     To have each section float relative to each other:
1129 */
```

```
1130 /*
1131     display: block ;
1132     float: left ;
1133     position: relative ;
1134     background: white ;
1135     border: 1px solid silver ;
1136     padding: .5em ;
1137 */
1138     margin: 0ex .5em 0ex .5em ;
1139     padding: 0 ;
1140 }
1141
1142
1143 figure {
1144     margin: 3ex auto 3ex auto ;
1145     padding: 1ex 1em 1ex 1em ;
1146     overflow-x: auto ;
1147 }
1148
1149
1150 /* To automatically center images in figures: */
1151 /*
1152 figure img.inlineimage {
1153     margin: 0ex auto 0ex auto ;
1154     display: block ;
1155 }
1156 */
1157
1158 /* To automatically center minipages in figures: */
1159 /*
1160 figure div.minipage, figure div.minipage div.minipage {
1161     margin: 1ex auto 1ex auto ;
1162     display: block ;
1163 }
1164 */
1165
1166 figure div.minipage p { font-size: 85% ; }
1167
1168 figure.subfigure, figure.subtable {
1169     display: inline-block ; margin: 3ex 1em 3ex 1em ;
1170 }
1171
1172 figcaption .minipage { margin: 0 ; padding: 0 }
1173
1174 div.floatrow { text-align: center; }
1175
1176 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
1177
1178 div.floatfoot { font-size: .85em ;
1179     border-top: 1px solid silver ; line-height: 1.2 ; }
```

```

1180
1181 figcaption , .lstlistingtitle {
1182     font-size: .85em ;
1183     text-align: center ;
1184     font-weight: bold ;
1185 margin-top: 1ex ;
1186 margin-bottom: 1ex ;
1187 }
1188
1189 figure.subfigure figcaption, figure.subtable figcaption {
1190     border-bottom: none ; background: none ;
1191 }
1192
1193 div.nonfloatcaption {
1194     margin: 1ex auto 1ex auto ;
1195     font-size: .85em ;
1196     text-align: center ;
1197     font-weight: bold ;
1198 }
1199
1200 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
1201 figure div.floatrow div.minipage figcaption {
1202 border: none ;
1203 background: none ;
1204 }
1205
1206
1207 table {
1208     margin: 1ex auto 1ex auto ;
1209     border-collapse: collapse ;
1210     border-spacing: 0px ;
1211     line-height: 1.3 ;
1212 }
1213
1214 tr.hline {border-top: 1px solid silver ; margin-top: 0ex ;
1215     margin-bottom: 0ex ; } /* for \hline */
1216
1217 tr.tbrule {border-top: 1px solid black ; margin-top: 0ex ;
1218     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
1219
1220 td {padding: 1ex .5em 1ex .5em ;}
1221
1222 table td.tdl { text-align: left ; vertical-align: middle ; }
1223 table td.tdc { text-align: center ; vertical-align: middle ; }
1224 table td.tdr { text-align: right ; vertical-align: middle ; }
1225 table td.tdp { text-align: left ; vertical-align: bottom ; }
1226 table td.tdm { text-align: left ; vertical-align: middle ; }
1227 table td.tdb { text-align: left ; vertical-align: top ; }
1228 table td.tdP { text-align: center ; vertical-align: bottom ; }
1229 table td.tdM { text-align: center ; vertical-align: middle ; }

```



```

1230 table td.tdB { text-align: center ; vertical-align: top ; }
1231 table td.tdLrule { text-align: left ; border-top: 1px solid silver ;
1232     vertical-align: middle ; } /* for cmidrule */
1233 table td.tdCrule { text-align: center ; border-top: 1px solid silver ;
1234     vertical-align: middle ; }
1235 table td.tdRrule { text-align: right ; border-top: 1px solid silver ;
1236     vertical-align: middle ; }
1237 table td.tdPrule { text-align: left ; border-top: 1px solid silver ;
1238     vertical-align: bottom ; }
1239 table td.tdMrule { text-align: left ; border-top: 1px solid silver ;
1240     vertical-align: middle ; }
1241 table td.tdBrule { text-align: left ; border-top: 1px solid silver ;
1242     vertical-align: top ; }
1243 table td.tdPrule { text-align: center ; border-top: 1px solid silver ;
1244     vertical-align: bottom ; }
1245 table td.tdMrule { text-align: center ; border-top: 1px solid silver ;
1246     vertical-align: middle ; }
1247 table td.tdBrule { text-align: center ; border-top: 1px solid silver ;
1248     vertical-align: top ; }
1249
1250 /* Margins of paragraphs inside table cells: */
1251 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
1252     padding-bottom: 1ex ; margin: 0ex ; }
1253 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
1254     padding-bottom: 1ex ; margin: 0ex ; }
1255 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
1256     padding-bottom: 1ex ; margin: 0ex ; }
1257
1258 td.tdp , td.tdprule , td.tdP , td.tdPrule
1259     { padding: 0ex .5em 0ex .5em ; }
1260 td.tdm , td.tdmrule , td.tdM , td.tdMrule
1261     { padding: 0ex .5em 0ex .5em ; }
1262 td.tdb , td.tdbrule , td.tdB , td.tdBrule
1263     { padding: 0ex .5em 0ex .5em ; }
1264
1265
1266 /* table notes: */
1267 .tnotes {
1268     margin: 0ex 5% 1ex 5% ;
1269     padding: 0.5ex 1em 0.5ex 1em;
1270     font-size:.85em;
1271     text-align: left ;
1272 }
1273
1274 .tnotes dl dt p {margin-bottom:0px;}
1275
1276 .tnoteitemheader {margin-right: 1em;}
1277
1278
1279

```

```
1280 /* center, flushleft, flushright environments */
1281 div.center{text-align:center;}
1282 div.center table {margin-left:auto;margin-right:auto;}
1283 div.flushleft{text-align:left;}
1284 div.flushleft table {margin-left:0em ; margin-right:auto;}
1285 div.flushright{text-align:right;}
1286 div.flushright table {margin-left:auto ; margin-right: 0em ;}
1287
1288
1289
1290
1291 /* program listing callouts: */
1292 span.callout {
1293     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1294     Geneva, Verdana, sans-serif ;
1295     border-radius: .5em;
1296     background-color:black;
1297     color:white;
1298     padding:0px .25em 0px .25em;
1299     margin: 0 ;
1300     font-weight: bold;
1301     font-size:.72em ;
1302 }
1303
1304 div.programlisting pre.verbatim span.callout{
1305 font-size: .85em ;
1306 }
1307
1308
1309
1310
1311
1312 div.published
1313 {
1314     text-align: center ;
1315     font-variant: normal ;
1316     font-style: italic ;
1317     font-size: 1em ;
1318     margin: 3ex 0em 3ex 0em ;
1319 }
1320
1321 div.subtitle
1322 {
1323     text-align: center ;
1324     font-variant: normal ;
1325     font-style: italic ;
1326     font-size: 1.25em ;
1327     margin: 3ex 0em 3ex 0em ;
1328 }
1329
```

```
1330 div.subtitle p { margin: 1ex ; }
1331
1332 div.author
1333 {
1334     font-variant: normal ;
1335     font-style: normal ;
1336     font-size: 1em ;
1337     margin: 3ex 0em 3ex 0em ;
1338 }
1339
1340 div.author table {
1341     margin: 3ex auto 0ex auto ;
1342     background: none ;
1343 }
1344
1345 div.author table tbody tr td { padding: .25ex ; }
1346
1347 span.affiliation {font-size: .85em ; font-variant: small-caps; }
1348
1349 div.titledate {
1350     text-align: center ;
1351     font-size: .85em ;
1352     font-style: italic;
1353     margin: 6ex 0em 6ex 0em ;
1354 }
1355
1356
1357 nav.topnavigation{
1358     text-align: left ;
1359     padding: 0.5ex 1em 0.5ex 1em ;
1360 /*     margin: 2ex 0em 3ex 0em ; */
1361     margin: 0 ;
1362     border-bottom: 1px solid silver ;
1363     border-top: 1px solid silver ;
1364     clear:right ;
1365 }
1366
1367 nav.botnavigation{
1368     text-align: left ;
1369     padding: 0.5ex 1em 0.5ex 1em ;
1370 /*     margin: 3ex 0em 2ex 0em ; */
1371     margin: 0 ;
1372     border-top: 1px solid silver ;
1373     border-bottom: 1px solid silver ;
1374     clear:right ;
1375 }
1376
1377
1378 header{
1379     line-height: 1.2 ;
```

```
1380     font-size: 1em ;
1381 /*     border-bottom: 2px solid silver ; */
1382     margin: 0px ;
1383     padding: 0ex 1em 0ex 1em ;
1384     text-align:center ;
1385 }
1386
1387 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
1388
1389
1390 footer{
1391     font-size: .85em ;
1392     line-height: 1.2 ;
1393     margin-top: 1ex ;
1394     border-top: 2px solid silver ;
1395     padding: 2ex 1em 2ex 1em ;
1396     clear:right ;
1397     text-align:left ;
1398 }
1399
1400
1401 a.linkhome { font-weight:bold ; font-size: 1em ;}
1402
1403
1404 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
1405
1406 img.lateximage{
1407     padding: 0px 0px 0px 0px ;
1408     box-shadow: none ;
1409     border: none ;
1410     background: none ;
1411     margin: 0px 0px -.15ex 0px ;
1412     /* pdfcrop leaves a slight margin, adjust to baseline */
1413     max-width: 100% ;
1414     border-radius: 0ex ;
1415     border: none ;
1416 }
1417
1418
1419
1420 nav.sidetoc {
1421     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1422         "Lucida Bright", Georgia, serif;
1423     float:right ;
1424     width: 20%;
1425     border-left: 1px solid silver;
1426     border-top: 1px solid silver;
1427     border-bottom: 1px solid silver;
1428 /*     border-top: 2px solid #808080 ; */
1429     background: #FAF7F4 ;
```

```
1430     padding: 2ex 0em 2ex 1em ;
1431     margin: 0ex 0em 2ex 1em ;
1432     font-size:.9em ;
1433     border-radius: 20px 0px 0px 20px ;
1434 }
1435
1436 div.sidetoccontents {
1437 /*     border-top: 1px solid silver ; */
1438     overflow-y: auto ;
1439     width: 100% ;
1440     text-align: left ;
1441 }
1442
1443 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
1444     text-indent: 0 ; }
1445 nav.sidetoc p a {color:black ; font-size: .7em ;}
1446 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
1447     border-bottom: 1px solid silver ; }
1448 nav.sidetoc a:hover {text-decoration: underline ; }
1449
1450
1451
1452 section.textbody { margin: 0ex 1em 0ex 1em ;}
1453
1454
1455 div.multicolsheading { -webkit-column-span: all;
1456     -moz-column-span: all; column-span: all; }
1457 div.multicols { -webkit-columns: 3 380px ;
1458     -moz-columns: 3 380px ; columns: 3 380px ; }
1459 div.multicols p {margin-top: 0ex}
1460
1461
1462
1463 /* Used to support algorithmicx: */
1464 span.floatright { float: right ; }
1465
1466
1467
1468
1469 /* Native LaTeX theorems: */
1470
1471 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
1472 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
1473
1474
1475 /* theorem, amsthm, and ntheorem packages */
1476
1477 span.theoremheader,
1478 span.theoremheaderplain,
1479 span.theoremheaderdefinition,
```

```
1480 span.theoremheaderbreak,
1481 span.theoremheadermarginbreak,
1482 span.theoremheaderchangebreak,
1483 span.theoremheaderchange,
1484 span.theoremheadermargin
1485 {
1486 font-style:normal ; font-weight: bold ; margin-right: 1em ;
1487 }
1488
1489 span.amsthmnameplain,
1490 span.amsthmnamedefinition,
1491 span.amsthmnumberplain,
1492 span.amsthmnumberdefinition
1493 {
1494 font-style:normal ; font-weight: bold ;
1495 }
1496
1497
1498 span.amsthmnameremark,
1499 span.amsthmnumberremark
1500 {font-style:italic ; font-weight: normal ; }
1501
1502
1503 span.amsthmnoteplain,
1504 span.amsthmnotedefinition
1505 {font-style:normal ;}
1506
1507
1508 span.theoremheaderremark,
1509 span.theoremheaderproof,
1510 span.amsthmproofname
1511 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
1512
1513 span.theoremheadersc
1514 {
1515 font-style:normal ;
1516 font-variant: small-caps ;
1517 font-weight: normal ;
1518 margin-right: 1em ;
1519 }
1520
1521 .theoremdemark {float:right}
1522
1523 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
1524 div.theorembodybreak, div.theorembodynonumberbreak,
1525 div.theorembodymarginbreak,
1526 div.theorembodychangebreak,
1527 div.theorembodychange,
1528 div.theorembodymargin
1529 {
```

```
1530 font-style:italic;
1531 margin-top: 3ex ; margin-bottom: 3ex ;
1532 }
1533
1534 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
1535 div.theorembodyplainupright, nonumberplainuprightsc,
1536 div.amsthmbodydefinition, div.amsthmbodyremark,
1537 div.amsthmproof
1538 {
1539 font-style: normal ;
1540 margin-top: 3ex ; margin-bottom: 3ex ;
1541 }
1542
1543 span.amsthmnoteremark {}
1544
1545
1546
1547 /*
1548 For CSS LaTeX and related logos:
1549 Based on:
1550 http://edward.oconnor.cx/2007/08/tex-poshlet
1551 http://nitens.org/taraborelli/texlogo
1552 */
1553
1554 .latexlogofont {
1555     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1556                 "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1557     font-variant: normal ;
1558 }
1559
1560 .latexlogo {
1561     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1562                 "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1563     letter-spacing: .03em ;
1564     font-size: 1.1em;
1565 }
1566
1567 .latexlogo sup {
1568     text-transform: uppercase;
1569     letter-spacing: .03em ;
1570     font-size: 0.85em;
1571     vertical-align: 0.15em;
1572     margin-left: -0.36em;
1573     margin-right: -0.15em;
1574 }
1575
1576 .latexlogo sub {
1577     text-transform: uppercase;
1578     vertical-align: -0.5ex;
1579     margin-left: -0.1667em;
```

```
1580 margin-right: -0.125em;
1581 font-size: 1em;
1582 }
1583
1584 .xetexlogo {
1585     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1586         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1587     letter-spacing: .03em ;
1588     font-size: 1.1em;
1589 }
1590
1591 /* A smaller gap between Xe and Tex v.s. LaTeX: */
1592 .xetexlogo sub {
1593     text-transform: uppercase;
1594     vertical-align: -0.5ex;
1595     margin-left: -0.0667em;
1596     margin-right: -0.2em;
1597     font-size: 1em;
1598     letter-spacing: .03em ;
1599 }
1600
1601 /* A large gap between Xe and LaTeX v.s. TeX: */
1602 .xelatexlogo sub {
1603     text-transform: uppercase;
1604     vertical-align: -0.5ex;
1605     margin-left: -0.0667em;
1606     margin-right: -.05em;
1607     font-size: 1em;
1608     letter-spacing: .03em ;
1609 }
1610
1611 .amslogo {
1612     font-family: "TeXGyreChorus","URW Chancery L",
1613         "Apple Chancery","ITC Zapf Chancery","Monotype Corsiva",
1614         "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
1615         "Hoefler Text", Times, "Times New Roman", serif;
1616     font-style: italic;
1617 }
1618
1619 .lyxlogo {
1620     font-family: "URW Classico", Optima, "Linux Biolinum O",
1621         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
1622     Verdana, sans-serif ;
1623 }
1624
1625
1626
1627
1628 /* Only display top and bottom navigation if a small screen: */
1629 /* Hide the sidetoc if a small screen: */
```



```
1630 nav.topnavigation { display:none; }
1631 nav.botnavigation { display:none; }
1632
1633 @media screen and (max-width: 45em) {
1634 /*      nav.sidetoc {display:none;} */
1635     nav.sidetoc {
1636         float: none ;
1637         width: 100% ;
1638         margin: 5ex 0px 5ex 0px ;
1639         padding: 0 ;
1640         border-radius: 0 ;
1641         border-bottom: 1px solid black ;
1642         border-top: 1px solid black ;
1643         box-shadow: none ;
1644     }
1645 /*      nav.topnavigation { display:block } */
1646     nav.botnavigation { display:block }
1647     .marginpar {
1648         max-width: 100%;
1649         float: none;
1650         display:block ;
1651         margin: 1ex 1em 1ex 1em ;
1652     }
1653 }
1654
1655 @media print {
1656     body {
1657         font-family: "Linux Libertine O",
1658             "DejaVu Serif", "Bitstream Vera Serif",
1659             "Liberation Serif", "Nimbus Roman No 9 L",
1660             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1661     }
1662     nav.sidetoc { display:none; }
1663     nav.topnavigation { display: none; }
1664     nav.botnavigation { display: none; }
1665 }
1666
1667 @media handheld {
1668     nav.sidetoc { display:none; }
1669     nav.topnavigation { display:block }
1670     nav.botnavigation { display:block }
1671 }
1672
1673 @media projection {
1674     nav.sidetoc { display:none; }
1675     nav.topnavigation { display:block }
1676     nav.botnavigation { display:block }
1677 }
1678 \end{VerbatimOut}
1679 % \end{Verbatim}% for syntax highlighting
```

```
1680 \end{warpprint}
```

26.5 lwarp__sagebrush.css

File `lwarp_sagebrush.css` An optional CSS which may be used for a semi-modern appearance.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```
1681 \begin{warpprint}
1682 \begin{VerbatimOut}{lwarp_sagebrush.css}
1683 @import url("lwarp.css") ;
1684
1685
1686 A:link {color:#105030 ; text-decoration: none ; }
1687 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
1688 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
1689 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
1690
1691
1692
1693 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1694 {
1695     font-family: "URW Classico", Optima, "Linux Biolinum O",
1696         "Linux Libertine O", "Liberation Serif",
1697         "Nimbus Roman No 9 L", "FreeSerif",
1698         "Hoefler Text", Times, "Times New Roman", serif;
1699     font-variant: small-caps ;
1700 font-weight: normal ;
1701     color: #304070 ;
1702     text-shadow: 2px 2px 3px #808080;
1703 }
1704
1705 h1 { /* title of the entire website, used on each page */
1706     font-variant: small-caps ;
1707     color: #304070 ;
1708     text-shadow: 2px 2px 3px #808080;
1709     background-color: #F7F7F0 ;
1710     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
1711 }
1712
1713 h1 {
1714     border-bottom: 1px solid #304070;
1715     border-top: 2px solid #304070;
1716 }
1717
1718 h2 {
```

```
1719 border-bottom: 1px solid #304070;
1720 border-top: 2px solid #304070;
1721 background-color: #F7F7F0 ;
1722 background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
1723 }
1724
1725
1726
1727 div.abstract {
1728     background: #f5f5eb ;
1729     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1730
1731     border: 1px solid silver;
1732     border-radius: 1em ;
1733 }
1734
1735 div.abstract dl {line-height:1.5;}
1736 div.abstract dt {color:#304070;}
1737
1738 div.abstracttitle{
1739     font-family: "URW Classico", Optima, "Linux Biolinum O",
1740         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1741         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1742     font-weight:bold;
1743     font-variant: small-caps ;
1744     font-size:1.5em;
1745     border-bottom: 1px solid silver ;
1746     color: #304070 ;
1747     text-align: center ;
1748     text-shadow: 1px 1px 2px #808080;
1749 }
1750
1751 span.abstracrunintitle{
1752     font-family: "URW Classico", Optima, "Linux Biolinum O",
1753         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1754         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1755     font-weight:bold;
1756 }
1757
1758
1759 div.epigraph {
1760     background: #f5f5eb ;
1761     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1762
1763     border: 1px solid silver ;
1764     border-radius: 1ex ;
1765     box-shadow: 3px 3px 3px #808080 ;
1766 }
1767
1768
```

```
1769 .example {
1770     background-color: #f5f5eb ;
1771     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1772
1773 }
1774
1775 div.exampletitle{
1776     font-family: "URW Classico", Optima, "Linux Biolinum O",
1777         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1778         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1779     font-weight:bold;
1780     font-variant: small-caps ;
1781     border-bottom: 1px solid silver ;
1782     color: #304070 ;
1783     text-align: center ;
1784     text-shadow: 1px 1px 2px #808080;
1785 }
1786
1787
1788 .sidebar {
1789     background-color: #f5f5eb ;
1790     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1791
1792 }
1793
1794 div.sidebar{
1795     font-family: "URW Classico", Optima, "Linux Biolinum O",
1796         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1797         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1798     font-weight:bold;
1799     font-variant: small-caps ;
1800     border-bottom: 1px solid silver ;
1801     color: #304070 ;
1802     text-align: center ;
1803     text-shadow: 1px 1px 2px #808080;
1804 }
1805
1806
1807 .fancyvrblabel {
1808     font-family: "URW Classico", Optima, "Linux Biolinum O",
1809         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1810         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1811     font-weight:bold;
1812     font-variant: small-caps ;
1813     font-size: 1.5em ;
1814     color: #304070 ;
1815     text-align: center ;
1816     text-shadow: 1px 1px 2px #808080;
1817 }
1818
```

```
1819
1820
1821 .minipage {
1822     background-color: #eeeeee7 ;
1823     border: 1px solid silver ;
1824     border-radius: 1ex ;
1825 }
1826
1827 .framed .minipage , .framedleftbar .minipage {
1828     border: none ;
1829     background: none ;
1830     padding: 0ex ;
1831     margin: 0ex ;
1832 }
1833
1834 figure.figure .minipage, figcaption .minipage { border: none; }
1835
1836 div.marginblock div.minipage { border: none; }
1837
1838 figure , div.marginblock {
1839     background-color: #eeeeee7 ;
1840     border: 1px solid silver ;
1841     border-radius: 1ex ;
1842     box-shadow: 3px 3px 3px #808080 ;
1843 }
1844
1845 figure figure {
1846     border: 1px solid silver ;
1847     margin: 0em ;
1848     box-shadow: none ;
1849 }
1850
1851 /*
1852 figcaption {
1853     border-top: 1px solid silver ;
1854     border-bottom: 1px solid silver ;
1855     background-color: #e8e8e8 ;
1856 }
1857 */
1858
1859
1860 div.table {
1861     box-shadow: 3px 3px 3px #808080 ;
1862 }
1863
1864 /*
1865 .tnotes {
1866     background: #e8e8e8;
1867     border: 1px solid silver;
1868 }
```

```
1869 */
1870
1871
1872 nav.topnavigation{
1873     background-color: #b0b8b0 ;
1874     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
1875 }
1876
1877 nav.botnavigation{
1878     background-color: #b0b8b0 ;
1879     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
1880 }
1881
1882
1883
1884 header{
1885     background-color: #F7F7F0 ;
1886     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
1887 }
1888
1889 footer{
1890     background-color: #F7F7F0 ;
1891     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
1892 }
1893
1894
1895
1896 nav.sidetoc {
1897     background-color: #F7F7F0 ;
1898     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
1899     box-shadow: 3px 3px 3px #808080 ;
1900     border-radius: 0px 0px 0px 20px ;
1901 }
1902
1903 div.sidetoc title {color: #304070 ; }
1904
1905 nav.sidetoc a:hover {
1906     color:#006000 ;
1907     text-decoration: none ;
1908     text-shadow:0px 0px 2px #a0a0a0;
1909 }
1910
1911
1912 @media screen and (max-width: 45em) {
1913     nav.sidetoc { border-radius: 0 ; }
1914 }
1915
1916
1917 \end{VerbatimOut}
1918 % \end{Verbatim}% for syntax highlighting
```

```
1919 \end{warpprint}
```

26.6 lwarp_formal.css

File `lwarp_formal.css` An optional CSS which may be used for a more formal appearance.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```
1920 \begin{warpprint}
1921 \begin{VerbatimOut}{lwarp_formal.css}
1922 @import url("lwarp.css") ;
1923
1924
1925
1926 A:link {color:#802020 ; text-decoration:none; }
1927 A:visited {color:#802020 ; text-shadow:none ;}
1928 A:hover {color:#400000 ; text-shadow:none ;}
1929 A:active {color:#C00000 ; text-shadow:none ;}
1930
1931
1932 body {
1933     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1934         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1935         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1936         "Times New Roman", serif;
1937     background: #fffcf5;
1938 }
1939
1940 span.textrm {
1941     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1942         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1943         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1944         "Times New Roman", serif;
1945 }
1946
1947 span.textsf {
1948     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1949         Geneva, Verdana, sans-serif ;
1950 }
1951
1952
1953
1954 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1955 {
1956     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1957         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
```

```
1958         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1959         "Times New Roman", serif;
1960     color: #800000 ;
1961     text-shadow: none ;
1962 }
1963
1964 h1, h2 {
1965     background-color: #fffcf5 ;
1966     background-image: none ;
1967     border-bottom: 1px solid #808080;
1968     border-top: 2px solid #808080;
1969 }
1970
1971 div.abstracttitle {
1972     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1973     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1974     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1975     "Times New Roman", serif;
1976     color: black ;
1977     text-shadow: none ;
1978 }
1979
1980 span.abstracrunintitle {
1981     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1982     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1983     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1984     "Times New Roman", serif;
1985     color: black ;
1986     text-shadow: none ;
1987 }
1988
1989 div.abstract { font-size: 100% }
1990
1991 .sidebar {
1992     background: #fffcf5;
1993     background-image: none ;
1994     margin: 2em 5% 2em 5%;
1995     padding: 0.5em 1em;
1996     border: none ;
1997     border-top : 1px solid silver;
1998     border-bottom : 1px solid silver;
1999     font-size: 90% ;
2000 }
2001
2002 div.sidebartitle{
2003     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2004     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2005     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2006     "Times New Roman", serif;
2007     color: #800000 ;
```



```
2008     text-shadow: none ;
2009     border: none ;
2010 }
2011
2012 .example {
2013     background: #fffcf5;
2014     background-image: none ;
2015     margin: 2em 5% 2em 5%;
2016     padding: 0.5em 1em;
2017     border: none ;
2018     border-top : 1px solid silver;
2019     border-bottom : 1px solid silver;
2020 }
2021
2022 div.exampletitle{
2023     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2024         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2025         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2026         "Times New Roman", serif;
2027     color: #800000 ;
2028     text-shadow: none ;
2029     border: none ;
2030 }
2031
2032 div.fancyvrblabel{
2033     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2034         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2035         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2036         "Times New Roman", serif;
2037     color: #800000 ;
2038     text-shadow: none ;
2039     border: none ;
2040 }
2041
2042
2043
2044 .verse {
2045     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2046         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2047         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2048         "Times New Roman", serif;
2049 }
2050
2051
2052 figure {
2053     margin: 3ex 5% 3ex 5% ;
2054     padding: 1ex 1em 1ex 1em ;
2055     background-color: #fffcf5 ;
2056     overflow-x: auto ;
2057     border: none ;
```

```
2058 /*      border-top: 1px solid silver; */
2059 /*      border-bottom: 1px solid silver; */
2060 }
2061
2062
2063 figcaption , .lstlisting {
2064     border: none ;
2065 /*      border-top: 1px solid silver ; */
2066 /*      border-bottom: 1px solid silver ; */
2067     background-color: #ffffcf5 ;
2068 }
2069
2070 .tnotes {
2071     background: #ffffcf5 ;
2072 }
2073
2074 .theorem {
2075     background: none ;
2076 }
2077
2078 .minipage {
2079     background-color: #ffffcf5 ;
2080     border: none ;
2081 }
2082
2083 div.floatrow figure { border: none ; }
2084
2085 figure figure { border: none ; }
2086
2087
2088 nav.toc, nav.lof, nav.lot, nav.lol {
2089     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2090         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2091         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2092         "Times New Roman", serif;
2093 }
2094
2095 nav.sidetoc {
2096     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2097         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2098         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2099         "Times New Roman", serif;
2100     background-image: linear-gradient(to bottom, #ffffcf5, #C0C0C0);
2101     border-radius: 0px 0px 0px 20px ;
2102 }
2103
2104 div.sidetoctitle{
2105     color: #800000 ;
2106 }
2107
```

```

2108 header{
2109     background-color: #e0e0e0 ;
2110     background-image: linear-gradient(to top, #fff5f5, #b0b0b0);
2111     text-align:center ;
2112 }
2113
2114 footer{
2115     background-color: #e0e0e0 ;
2116     background-image: linear-gradient(to bottom, #fff5f5, #b0b0b0);
2117     padding: 2ex 1em 2ex 1em ;
2118     clear:right ;
2119     text-align:left ;
2120 }
2121
2122 nav.botnavigation {
2123     background: #dedcd5 ;
2124     border-top: 1px solid black ;
2125 }
2126 \end{VerbatimOut}
2127 % \end{Verbatim}% for syntax highlighting
2128 \end{warpprint}

```

26.7 sample_project.css

File `sample_project.css` The project-specific CSS file. Use with `\CSSFilename`.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```

2129 \begin{warpprint}
2130 \begin{VerbatimOut}{sample_project.css}
2131 /* ( --- Start of project.css --- ) */
2132 /* A sample project-specific CSS file for lwarp --- ) */
2133
2134 /* Load default lwarp settings: */
2135 @import url("lwarp.css") ;
2136 /* or lwarp_formal.css, lwarp_sagebrush.css */
2137
2138 /* Project-specific CSS setting follow here. */
2139 /* . . . */
2140
2141 /* ( --- End of project.css --- ) */
2142 \end{VerbatimOut}
2143 % \end{Verbatim}% for syntax highlighting
2144 \end{warpprint}

```

26.8 lwarp.xdy

File `lwarp.xdy` Used to modify the index for lwarp.

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

```

2145 \begin{warpprint}
2146 \begin{VerbatimOut}{lwarp.xdy}
2147 (require "tex/inputenc/latin.xdy")
2148 (merge-rule "\\PS *" "Postscript")
2149 (require "texindy.xdy")
2150 (require "page-ranges.xdy")
2151 (require "book-order.xdy")
2152 (markup-locref :open "\hyperindexref{" :close "}")
2153 \end{VerbatimOut}
2154 % \end{Verbatim}% for syntax highlighting
2155 \end{warpprint}

```

26.9 lwarp_mathjax.txt

File `lwarp_mathjax.txt` Used by lwarp when using MathJax.

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

```

2156 \begin{warpprint}
2157 \begin{VerbatimOut}{lwarp_mathjax.txt}
2158 <!-- https://groups.google.com/forum/#!topic/
2159                               mathjax-users/jUtewUcE2bY -->
2160 <script type="text/x-mathjax-config">
2161 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
2162     var seteqsectionDefault = {name: "", num: 0};
2163     var seteqsections = {}, seteqsection = seteqsectionDefault;
2164     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
2165     var AMS = MathJax.Extension["TeX/AMSmath"];
2166     TEX.Definitions.Add({
2167     macros: {
2168         seteqsection: "mySection",
2169         seteqnumber: "mySetEqNumber"
2170     }
2171     });
2172
2173     PARSE.Augment({
2174     mySection: function (name) {
2175         seteqsection.num = AMS.number;
2176         var n = this.GetArgument(name);

```

```

2177         if (n === "") {
2178             seteqsection = seteqsectionDefault;
2179         } else {
2180             if (!seteqsections["_"+n])
2181                 seteqsections["_"+n] = {name:n, num:0};
2182             seteqsection = seteqsections["_"+n];
2183         }
2184         AMS.number = seteqsection.num;
2185     },
2186     mySetEqNumber: function (name) {
2187         var n = this.GetArgument(name);
2188         if (!n || !n.match(/^ *[0-9]+ *$/))
2189             n = ""; else n = parseInt(n)-1;
2190         <!-- $ syntax highlighting -->
2191         if (n === "" || n < 1)
2192             TEX.Error
2193                 ("Argument to "+name+" should be a positive integer");
2194         AMS.number = n;
2195     }
2196 });
2197 MathJax.Hub.Config({
2198   TeX: {
2199     equationNumbers: {
2200       formatTag: function (n)
2201         {return "("+(seteqsection.name+"."+n).replace(/\^\.\/, "(")+")"},
2202       formatID: function (n) {
2203         n = (seteqsection.name+'.'+n).replace
2204           (/[:'>&]/g, "").replace(/\^\.\/, "");
2205         return 'mjx-eqn-' + n;
2206       }
2207     }
2208   }
2209 });
2210 }>
2211 </script>
2212
2213 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
2214 <script type="text/x-mathjax-config">
2215   MathJax.Ajax.config.path["Contrib"] =
2216     "https://cdn.mathjax.org/mathjax/contrib";
2217 </script>
2218
2219 <!-- https://github.com/mathjax/MathJax-third-party-extensions/
2220                                tree/master/siunitx -->
2221 <script type="text/x-mathjax-config">
2222   MathJax.Hub.Config({
2223     extensions: ["tex2jax.js", "[Contrib]/siunitx/siunitx.js"],
2224     jax: ["input/TeX", "output/HTML-CSS"],
2225     tex2jax: {inlineMath: [["$", "$"], ["\\(", "\\)"]]},
2226     TeX: {extensions: ["AMSmath.js", "AMSsymbols.js", "sinuitx.js"]}

```

```

2227 });
2228 </script>
2229
2230 <script type="text/x-mathjax-config">
2231 MathJax.Hub.Config({
2232     TeX: {
2233         equationNumbers: {
2234             autoNumber: "AMS"
2235         }
2236     }
2237 });
2238 </script>
2239
2240 <!-- Alternative CDN provider: -->
2241 <script type="text/javascript" async
2242 src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.0/MathJax.js?config=TeX-AMS_HTML-full">
2243 </script>
2244
2245 <!-- No longer supported after April 30, 2017: -->
2246 <!--
2247 <script
2248     src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
2249 </script>
2250 -->
2251
2252 \end{VerbatimOut}
2253 % \end{Verbatim}% for syntax highlighting
2254 \end{warpprint}

```

26.10 lwarpmk option

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

```

2255 \begin{LWR@createlwarpmk}

```

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

```

2256 \begin{VerbatimOut}{lwarpmk.lua}
2257 #!/usr/bin/env texlua
2258
2259 -- Copyright 2016-2017 Brian Dunn
2260
2261 -- Print the usage of the lwarpmk command:
2262
2263 printversion = "v0.32"
2264
2265 function printhelp ()

```

```

2266 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
2267 end
2268
2269 function printusage ()
2270 print ( [[
2271
2272 lwarpmk print [project]: Compile a print version.
2273 lwarpmk printindex [project]: Process the index for the print version.
2274 lwarpmk printglossary [project]: Process the glossary for the print version.
2275 lwarpmk html [project]: Compile an HTML version.
2276 lwarpmk htmlindex [project]: Process the index for the html version.
2277 lwarpmk htmlglossary [project]: Process the glossary for the html version.
2278 lwarpmk again [project]: Touch the source code to trigger recompiles.
2279 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
2280 lwarpmk pdftohtml [project]:
2281     For use with latexmk or a Makefile:
2282     Convert project_html.pdf to project_html.html and
2283     individual HTML files.
2284 lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, .gl*
2285 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
2286 lwarpmk -h: Print this help message.
2287 lwarpmk --help: Print this help message.
2288
2289 ]] )
2290 printconf ()
2291 end
2292
2293 -- Print the format of the configuration file lwarpmk.conf:
2294
2295 function printconf ()
2296 print ( [[
2297 An example lwarpmk.conf or <project>.lwarpmkconf project file:
2298 --
2299 opsystem = "Unix"      (or "Windows")
2300 latexname = "pdflatex" (or "lualatex", or "xelatex")
2301 sourcename = "projectname" (the source-code filename w/o .tex)
2302 homehtmlfilename = "index" (or perhaps the project name)
2303 htmlfilename = "" (or "projectname" - filename prefix)
2304 latexmk = "false" (or "true" to use latexmk to build PDFs)
2305 language = "english" (use a language supported by xindy)
2306 xdyfile = "lwarp.xdy" (or a custom file based on lwarp.xdy)
2307 --
2308 Filenames must contain only letters, numbers, underscore, or dash.
2309 Values must be in "quotes".
2310
2311 ]] ) ;
2312 end
2313
2314
2315 -- Split one large sourcefile into a number of files,

```

```
2316 -- starting with destfile.
2317 -- The file is split at each occurrence of <!--|Start file|newfilename|*
2318
2319 function splitfile (destfile,sourcefile)
2320 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
2321 local sfile = io.open(sourcefile)
2322 io.output(destfile)
2323 for line in sfile:lines() do
2324 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*|)" ) ;
2325 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
2326 io.output(newfilename) ;
2327 else -- not a splitpoint
2328 io.write (line .. "\n") ;
2329 end
2330 end -- do
2331 io.close(sfile)
2332 end -- function
2333
2334 -- Incorrect value, so print an error and exit.
2335
2336 function cvalueerror ( line, linenum , cvalue )
2337     print ( linenum .. " : " .. line ) ;
2338     print ("lwarpmk: incorrect variable value \"" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
2339     printconf () ;
2340     os.exit(1) ;
2341 end
2342
2343 -- Load settings from the project's "lwarpmk.conf" file:
2344
2345 function loadconf ()
2346 -- Default configuration filename:
2347 local conffile = "lwarpmk.conf"
2348 -- Optional configuration filename:
2349 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
2350 -- Default language:
2351 language = "english"
2352 -- Default xdyfile:
2353 xdyfile = "lwarp.xdy"
2354 -- Verify the file exists:
2355 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
2356 print("lwarpmk: " .. conffile .. " does not exist.")
2357 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
2358 printhelp () ;
2359 os.exit(1) -- exit the entire lwarpmk script
2360 else -- file exists
2361 -- Read the file:
2362 print ("lwarpmk: Reading " .. conffile .. ".")
2363 local cfile = io.open(conffile)
2364 -- Scan each line:
2365 local linenum = 0
```



```

2366 for line in cfile:lines() do -- scan lines
2367 linenum = linenum + 1
2368 i,j,cvarname,cvalue = string.find (line,"([%w-_]*)%s*=%s*\"([%w%-_.]*)\"") ;
2369 -- Error if incorrect enclosing characters:
2370 if ( i == nil ) then
2371 print ( linenum .. " : " .. line ) ;
2372 print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
2373 printconf () ;
2374 os.exit(1) ;
2375 end
2376 if ( cvarname == "opsystem" ) then
2377     -- Verify choice of opsystem:
2378     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
2379         opsystem = cvalue
2380     else
2381         cvalueerror ( line, linenum , cvalue )
2382     end
2383 elseif ( cvarname == "latexname" ) then
2384     -- Verify choice of LaTeX compiler:
2385     if (
2386         (cvalue == "pdflatex") or
2387         (cvalue == "xelatex") or
2388         (cvalue == "lualatex")
2389     ) then
2390         latexname = cvalue
2391     else
2392         cvalueerror ( line, linenum , cvalue )
2393     end
2394 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
2395 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
2396 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
2397 elseif ( cvarname == "latexmk" ) then latexmk = cvalue
2398 elseif ( cvarname == "language" ) then language = cvalue
2399 elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue
2400 else
2401 print ( linenum .. " : " .. line ) ;
2402 print ( "lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. conffile .. ".\n" ) ;
2403 printconf () ;
2404 os.exit(1) ;
2405 end
2406 end -- do scan lines
2407 io.close(cfile)
2408 end -- file exists
2409 -- Select some operating-system commands:
2410 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
2411 rmname = "rm"
2412 mvname = "mv"
2413 touchnamepre = "touch"
2414 touchnamepost = ""
2415 dirslash = "/"

```

```
2416 opquote= "\'"
2417 elseif opsystem=="Windows" then -- For Windows
2418 rmname = "DEL"
2419 mvname = "MOVE"
2420 touchnamepre = "COPY /b"
2421 touchnamepost = "+,,"
2422 dirsslash = "\\"
2423 opquote= "\""
2424 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
2425 end --- for Windows
2426
2427 -- set xindycmd according to pdflatex vs xelatex/lualatex:
2428 if ( latexname == "pdflatex" ) then
2429 xindycmd = "texindy -C utf8"
2430 glossarycmd = "xindy -C utf8"
2431 else
2432 xindycmd = "xindy -M texindy -C utf8"
2433 glossarycmd = "xindy -C utf8"
2434 end
2435
2436 end -- loadconf
2437
2438
2439 function refreshdate ()
2440 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
2441 end
2442
2443
2444 -- Scan the LaTeX log file for the phrase "Rerun to get",
2445 -- indicating that the file should be compiled again.
2446 -- Return true if found.
2447
2448 function reruntoget (filesorce)
2449 local fsource = io.open(filesorce)
2450 for line in fsource:lines() do
2451 if ( string.find(line,"Rerun to get") ~= nil ) then
2452 io.close(fsource)
2453 return true
2454 end
2455 end
2456 io.close(fsource)
2457 return false
2458 end
2459
2460
2461 -- Compile one time, return true if should compile again.
2462 -- fsuffix is "" for print, "_html" for HTML output.
2463
2464 function onetime (fsuffix)
2465 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
```

```
2466 err = os.execute(
2467 --      "echo " ..
2468      latexname .. " " .. sourcename..fsuffix )
2469 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2470 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
2471 end
2472
2473
2474 -- Compile up to five times.
2475 -- fsuffix is "" for print, "_html" for HTML output
2476
2477 function manytimes (fsuffix)
2478 if onetime(fsuffix) == true then
2479 if onetime(fsuffix) == true then
2480 if onetime(fsuffix) == true then
2481 if onetime(fsuffix) == true then
2482 if onetime(fsuffix) == true then
2483 end end end end end
2484 end
2485
2486 -- Exit if the given file does not exist.
2487
2488 function verifyfileexists (filename)
2489 if (lfs.attributes ( filename , "modification" ) == nil ) then
2490 print ( "lwarpmk: " .. filename .. " not found." ) ;
2491 os.exit (1) ;
2492 end
2493 end
2494
2495
2496 -- Convert <project>_html.pdf into HTML files:
2497
2498 function pdftohtml ()
2499 -- Convert to text:
2500 print ("lwarpmk: Converting " .. sourcename
2501      .. "_html.pdf to " .. sourcename .. "_html.html")
2502 os.execute("pdftotext -enc UTF-8 -nopgbrk -layout "
2503      .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
2504 -- Split the result into individual HTML files:
2505 splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
2506 end
2507
2508
2509 -- Remove auxiliary files:
2510
2511 function removeaux ()
2512 os.execute ( rmname .. " " ..
2513      sourcename .. ".aux " .. sourcename .. "_html.aux " ..
2514      sourcename .. ".toc " .. sourcename .. "_html.toc " ..
2515      sourcename .. ".lof " .. sourcename .. "_html.lof " ..
```

```

2516         sourcename ..".lot " .. sourcename .. "_html.lot " ..
2517         sourcename ..".idx " .. sourcename .. "_html.idx " ..
2518         sourcename ..".ind " .. sourcename .. "_html.ind " ..
2519         sourcename ..".log " .. sourcename .. "_html.log " ..
2520         sourcename ..".gl* " .. sourcename .. "_html.gl* "
2521     )
2522 end
2523
2524
2525
2526 -- Create lateximages based on lateximages.txt:
2527 function createlateximages ()
2528 print ("lwarpmk: Creating lateximages.")
2529 local limagesfile = io.open("lateximages.txt")
2530 -- Create the lateximages directory, ignore error if already exists
2531 err = os.execute("mkdir lateximages")
2532 -- Scan lateximages.txt
2533 for line in limagesfile:lines() do
2534 -- lwimpage is the page number in the PDF which has the image
2535 -- lwimgnum is the sequential lateximage number to assign for the image
2536 i,j,lwimpage,lwimgnum = string.find (line,"|(.)|(.)|")
2537 -- For each entry:
2538 if ( i~=nil ) then
2539 -- Separate out the image into its own single-page pdf:
2540 err = os.execute(
2541 "pdfseparate -f " .. lwimpage .. " -l " ..
2542 "lwimpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
2543 -- Crop the image:
2544 err = os.execute(
2545 "pdfcrop -- hires lateximagetemp-" .. lwimpage .. ".pdf lateximage-" .. lwimgnum .. ".pdf")
2546 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2547 -- Convert the image to svg:
2548 err = os.execute(
2549 "pdftocairo -svg lateximage-" .. lwimgnum .. ".pdf lateximage-" .. lwimgnum .. ".svg")
2550 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2551 -- Move the result into lateximages/:
2552 err = os.execute(
2553 mvname .. " lateximage-" .. lwimgnum .. ".svg lateximages" .. dirslash )
2554 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2555 -- Remove the temporary files:
2556 err = os.execute(
2557 rmname .. " lateximage-" .. lwimgnum .. ".pdf lateximagetemp-" .. lwimpage .. ".pdf")
2558 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2559 end
2560 end -- do
2561 io.close(limagesfile)
2562 end -- function
2563
2564
2565 -- Use latexmk to compile source and index:

```

```

2566 -- fsuffix is "" for print, or "_html" for HTML
2567 function compilelatexmk ( fsuffix )
2568     -- The recorder option is required to detect changes in <project>.tex
2569     -- while we are loading <project>_html.tex.
2570     err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
2571         .. "-e "
2572         .. opquote
2573         .. "$makeindex = q/"
2574         .. xindycmd
2575         .. " -M " .. xdyfile
2576         .. " -L " .. language .. " /"
2577         .. opquote
2578         .. " -pdflatex=\"\" .. latexname .. " %O %S\" "
2579         .. sourcename..fsuffix ..".tex" ) ;
2580     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2581 end
2582
2583
2584
2585 -- lwarpmk --version :
2586
2587 if (arg[1] == "--version") then
2588     print ( "lwarpmk: " .. printversion )
2589
2590 else -- not -- version
2591
2592 -- print intro:
2593
2594 print ("lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package.")
2595
2596 -- lwarpmk print:
2597
2598 if arg[1] == "print" then
2599     loadconf ( )
2600     if ( latexmk == "true" ) then
2601         compilelatexmk ( "" )
2602         print ("lwarpmk: Done.")
2603     else -- not latexmk
2604         verifyfileexists (sourcename .. ".tex") ;
2605         -- See if up to date:
2606         if (
2607             ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
2608             (
2609                 lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2610                 lfs.attributes ( sourcename .. ".pdf" , "modification" )
2611             )
2612         ) then
2613             -- Recompile if not yet up to date:
2614             manytimes( "" )
2615             print ("lwarpmk: Done." ) ;

```

```
2616     else
2617         print ("lwarpmk: " .. sourcename .. ".pdf is up to date.");
2618     end
2619 end -- not latexmk
2620
2621 -- lwarp printindex:
2622 -- Compile the index then touch the source
2623 -- to trigger a recompile of the document:
2624
2625 elseif arg[1] == "printindex" then
2626     loadconf ()
2627     print ("lwarpmk: Processing the index.")
2628     os.execute(
2629         xindycmd
2630         .. " -M " .. xdyfile
2631         .. " -L " .. language
2632         .. " " .. sourcename .. ".idx")
2633     print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2634     refreshdate ()
2635     print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2636     print ("lwarpmk: Done.")
2637
2638 -- lwarp printglossary:
2639 -- Compile the glossary then touch the source
2640 -- to trigger a recompile of the document:
2641
2642 elseif arg[1] == "printglossary" then
2643     loadconf ()
2644     print ("lwarpmk: Processing the glossary.")
2645
2646     os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
2647         " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
2648         .. sourcename .. ".glo")
2649     print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2650     refreshdate ()
2651     print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2652     print ("lwarpmk: Done.")
2653
2654 -- lwarpmk html:
2655
2656 elseif arg[1] == "html" then
2657     loadconf ()
2658     if ( latexmk == "true" ) then
2659         compilelatexmk ("_html")
2660         pdftohtml ()
2661         print ("lwarpmk: Done.")
2662     else -- not latexmk
2663         verifyfileexists ( sourcename .. ".tex" );
2664         -- See if exists and is up to date:
2665         if (
```

```

2666      ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
2667      (
2668          lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2669          lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
2670      )
2671  ) then
2672      -- Recompile if not yet up to date:
2673      manytimes("_html")
2674      pdftohtml ()
2675      print ("lwarpmk: Done.")
2676  else
2677      print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
2678  end
2679 end -- not latexmk
2680
2681 elseif arg[1] == "pdftohtml" then
2682     loadconf ()
2683     pdftohtml ()
2684
2685 -- lwarpmk htmlindex:
2686 -- Compile the index then touch the source
2687 -- to trigger a recompile of the document:
2688
2689 elseif arg[1] == "htmlindex" then
2690 loadconf ()
2691 print ("lwarpmk: Processing the index.")
2692 os.execute(
2693     xindycmd
2694     .. " -M " .. xdyfile
2695     .. " -L " .. language
2696     .. " " .. sourcename .. "_html.idx"
2697 )
2698 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2699 refreshdate ()
2700 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2701 print ("lwarpmk: Done.")
2702
2703 -- lwarpmk htmlglossary:
2704 -- Compile the glossary then touch the source
2705 -- to trigger a recompile of the document:
2706
2707 elseif arg[1] == "htmlglossary" then
2708 loadconf ()
2709 print ("lwarpmk: Processing the glossary.")
2710
2711 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
2712     "_html -t " .. sourcename .. "_html.glg -o " .. sourcename ..
2713     "_html.gls " .. sourcename .. "_html.glo")
2714
2715 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")

```

```
2716 refreshdate ()
2717 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2718 print ("lwarpmk: Done.")
2719
2720 -- lwarpmk limages:
2721 -- Scan the lateximages.txt file to create lateximages,
2722 -- then touch the source to trigger a recompile.
2723
2724 elseif arg[1] == "limages" then
2725 loadconf ()
2726 print ("lwarpmk: Processing images.")
2727 createlateximages ()
2728 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2729 refreshdate ()
2730 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2731 print ("lwarpmk: Done.")
2732
2733 -- lwarpmk again:
2734 -- Touch the source to trigger a recompile.
2735
2736 elseif arg[1] == "again" then
2737 loadconf ()
2738 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2739 refreshdate ()
2740 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2741 print ("lwarpmk: Done.")
2742
2743 -- lwarpmk clean:
2744 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, .gl*
2745
2746 elseif arg[1] == "clean" then
2747 loadconf ()
2748 removeaux ()
2749 print ("lwarpmk: Done.")
2750
2751 -- lwarpmk cleanall
2752 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, .gl*
2753 -- and also project.pdf, *.html
2754
2755 elseif arg[1] == "cleanall" then
2756 loadconf ()
2757 removeaux ()
2758 os.execute ( rmname .. " " ..
2759     sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
2760     "*.html"
2761 )
2762 print ("lwarpmk: Done.")
2763
2764 -- lwarpmk with no argument :
2765
```



```

2766 elseif (arg[1] == nil) then
2767 printhelp ()
2768
2769 -- lwarpmk -h or lwarpmk --help :
2770
2771 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
2772 printusage ()
2773
2774 else
2775 print ("lwarpmk: Unknown command \""..arg[1].."\".\n")
2776 printhelp ()
2777 end
2778
2779 end -- not --version
2780 \end{VerbatimOut}
2781 % \end{Verbatim}% for syntax highlighting
2782 \end{LWR@createlwarpmk}

```

27 Stacks

for HTML output: 2783 \begin{warpHTML}



Stacks are used to remember how to close sections and list items. Before a new section is started, previously nested sections and items must be closed out (un-nested) in proper order. Note that starting a new section may close several levels of previously nested items at the same time. For example, starting a new `\section` would close any currently open subsection, subsubsection, and paragraph. General environments are not nested on the stack since they have their own close mechanism. List environments are nested, and items inside those environments are nested one level deeper still. List environments may be nested inside other list environments, and list items are nested inside list environments as well. Thus, the stack may have items which are not necessarily in order, since a description may contain an enumerate, for example. Depths to be recorded in `\LWR@closedepthone`, etc.

27.1 Assigning depths

initial depths for empty stack entries:

```
2784 \newcommand*{\LWR@depthnone}{-5}
```

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

```
2785 \newcommand*{\LWR@depthfinished}{-4}
```

```

2786 \newcommand*{\LWR@depthpart}{-1}
2787 \newcommand*{\LWR@depthchapter}{0}
2788 \newcommand*{\LWR@depthsection}{1}
2789 \newcommand*{\LWR@depthsubsection}{2}
2790 \newcommand*{\LWR@depthsubsubsection}{3}
2791 \newcommand*{\LWR@depthparagraph}{4}
2792 \newcommand*{\LWR@depthsubparagraph}{5}

```

used by \itemize, \enumerate, \description:

```

2793 \newcommand*{\LWR@depthlist}{6}

```

used by \item:

```

2794 \newcommand*{\LWR@depthlistitem}{7}

```

27.2 Closing actions

A stack to record the action to take to close each nesting level: Add more levels of stack if necessary for a very deeply nested document, adding to \pushclose and \popclose as well.

```

2795 \newcommand*{\LWR@closeone}{}% top of the stack
2796 \newcommand*{\LWR@closetwo}{}
2797 \newcommand*{\LWR@closethree}{}
2798 \newcommand*{\LWR@closefour}{}
2799 \newcommand*{\LWR@closefive}{}
2800 \newcommand*{\LWR@closesix}{}
2801 \newcommand*{\LWR@closeseven}{}
2802 \newcommand*{\LWR@closeeight}{}
2803 \newcommand*{\LWR@closenine}{}
2804 \newcommand*{\LWR@closeten}{}
2805 \newcommand*{\LWR@closeeleven}{}
2806 \newcommand*{\LWR@closetwelve}{}

```

27.3 Closing depths

A stack to record the depth of each level:



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

Ex:

```

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

```

```

2807 \newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
2808 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
2809 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
2810 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
2811 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
2812 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
2813 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
2814 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
2815 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
2816 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}
2817 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
2818 \newcommand*{\LWR@closedephtwelve}{\LWR@depthnone}

```

27.4 Pushing and popping the stack

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

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

```

2819 \NewDocumentCommand{\pushclose}{m m}
2820 {
2821 \let\LWR@closetwelve\LWR@closeeleven
2822 \let\LWR@closeeleven\LWR@closeten
2823 \let\LWR@closeten\LWR@closenine
2824 \let\LWR@closenine\LWR@closeeight
2825 \let\LWR@closeeight\LWR@closeseven
2826 \let\LWR@closeseven\LWR@closesix
2827 \let\LWR@closesix\LWR@closefive
2828 \let\LWR@closefive\LWR@closefour
2829 \let\LWR@closefour\LWR@closethree
2830 \let\LWR@closethree\LWR@closetwo
2831 \let\LWR@closetwo\LWR@closeone
2832 \let\LWR@closeone#1
2833 \let\LWR@closedephtwelve\LWR@closedeptheleven
2834 \let\LWR@closedeptheleven\LWR@closedepthten
2835 \let\LWR@closedepthten\LWR@closedepthnine

```

```

2836 \let\LWR@closedepthnine\LWR@closedeptheight
2837 \let\LWR@closedeptheight\LWR@closedepthseven
2838 \let\LWR@closedepthseven\LWR@closedepthsix
2839 \let\LWR@closedepthsix\LWR@closedepthfive
2840 \let\LWR@closedepthfive\LWR@closedepthfour
2841 \let\LWR@closedepthfour\LWR@closedepththree
2842 \let\LWR@closedepththree\LWR@closedepthtwo
2843 \let\LWR@closedepthtwo\LWR@closedepthone
2844 \let\LWR@closedepthone#2
2845 }

```

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

```

2846 \newcommand*{\popclose}
2847 {
2848 \let\LWR@closeone\LWR@closetwo
2849 \let\LWR@closetwo\LWR@closethree
2850 \let\LWR@closethree\LWR@closefour
2851 \let\LWR@closefour\LWR@closefive
2852 \let\LWR@closefive\LWR@closesix
2853 \let\LWR@closesix\LWR@closeseven
2854 \let\LWR@closeseven\LWR@closeeight
2855 \let\LWR@closeeight\LWR@closenine
2856 \let\LWR@closenine\LWR@closeten
2857 \let\LWR@closeten\LWR@closeeleven
2858 \let\LWR@closeeleven\LWR@closetwelve
2859 \let\LWR@closedepthone\LWR@closedepthtwo
2860 \let\LWR@closedepthtwo\LWR@closedepththree
2861 \let\LWR@closedepththree\LWR@closedepthfour
2862 \let\LWR@closedepthfour\LWR@closedepthfive
2863 \let\LWR@closedepthfive\LWR@closedepthsix
2864 \let\LWR@closedepthsix\LWR@closedepthseven
2865 \let\LWR@closedepthseven\LWR@closedeptheight
2866 \let\LWR@closedeptheight\LWR@closedepthnine
2867 \let\LWR@closedepthnine\LWR@closedepthten
2868 \let\LWR@closedepthten\LWR@closedeptheleven
2869 \let\LWR@closedeptheleven\LWR@closedepthtwelve
2870 }

2871 \end{warpHTML}

```

28 Data arrays

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

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

To assign an empty value:

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

for HTML output: 2872 \begin{warpHTML}

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

```
2873 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
2874 \expandafter\edef\csname #1#2\endcsname{\expandonce#3}%
2875 }
```

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

```
2876 \newcommand*{\LWR@getexparray}[2]{\csuse{#1#2}}
```

```
2877 \end{warpHTML}
```

29 HTML entities

for HTML output: 2878 \begin{warpHTML}

HTML entites and HTML Unicode entities:

```
2879 \let\LWR@origampersand\&
```

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

```
2880 \newcommand*{\HTMLentity}[1]{\LWR@origampersand#1;}
```

```
\HTMLunicode {<hex__unicode>}
```

```
2881 \newcommand*{\HTMLunicode}[1]{\HTMLentity{\#x#1}}
```

```
\&
```

```
2882 \renewcommand*{\&}{\HTMLentity{amp}}
```

```
\textless
\textgreater
```

```

2883 \let\LWR@origtextless\textless
2884 \renewcommand*{\textless}{\HTMLentity{lt}}
2885
2886 \let\LWR@origtextgreater\textgreater
2887 \renewcommand*{\textgreater}{\HTMLentity{gt}}

2888 \end{warpHTML}

```

30 HTML filename generation

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

for HTML & PRINT: 2889 \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.

```
2890 \providecommand*{\BaseJobname}{\jobname}
```

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

```
2891 \providecommand*{\HTMLFilename}{}
```

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

```
2892 \providecommand*{\HomeHTMLFilename}{\BaseJobname}
```

`\SetHTMLFileNumber` $\{\langle number \rangle\}$

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

```

2893 \newcommand*{\SetHTMLFileNumber}[1]{%
2894 \setcounter{LWR@htmlfilenumber}{#1}%
2895 }

```

Bool FileSectionNames Selects how to create HTML file names.

Defaults to use section names in the filenames.

```
2896 \newbool{FileSectionNames}
2897 \booltrue{FileSectionNames}

2898 \end{warpall}
```

for HTML output: 2899 \begin{warpHTML}

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

```
2900 \newcounter{LWR@htmlfilenumber}
2901 \setcounter{LWR@htmlfilenumber}{0}
```

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

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

```
2902 \newcommand*{\LWR@htmlsectionfilename}[1]{%
2903 \LWR@traceinfo{LWR@htmlsectionfilename A}%
```

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

```
2904 \LWR@traceinfo{about to assign temp}%
2905 \edef\LWR@tempone{#1}%
2906 \LWR@traceinfo{about to compare with ??}%
2907 \ifthenelse{\equal{\LWR@tempone}{??}}{%
2908 {%
2909 \LWR@traceinfo{found ??}%
2910 }{%
2911 \LWR@traceinfo{not found ??}%
2912 }%
2913 \LWR@traceinfo{about to compare with zero or empty}%
2914 \ifthenelse{%
2915 \equal{\LWR@tempone}{0}}{%
2916 \OR \equal{\LWR@tempone}{}}%
2917 \OR \equal{\LWR@tempone}{??}}%
2918 }%
2919 {%
2920 \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
2921 \HomeHTMLFilename.html%
2922 }%
```

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`:

```

2923 {%
2924 \LWR@traceinfo{\LWR@htmlsectionfilename C \LWR@tempone}%
2925 \ifthenelse{%
2926 \equal{\HTMLFilename}{ } \AND \equal{\LWR@tempone}{Index} \OR \equal{\LWR@tempone}{index}%
2927 }%
2928 {%
2929 \LWR@traceinfo{prefixing the index name with an underscore.}%
2930 \_#1.html}%

```

Otherwise, create a filename with the chosen prefix:

```

2931 {\HTMLFilename#1.html}%
2932 }%
2933 \LWR@traceinfo{\LWR@htmlsectionfilename Z}%
2934 }

```

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

Prints the filename for the given label

```

2935 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
2936 \LWR@traceinfo{\LWR@htmlrefsectionfilename A: '#1!}%
2937 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
2938 \LWR@traceinfo{\LWR@htmlrefsectionfilename B}%
2939 }

2940 \end{warpHTML}

```

31 Homepage link

for HTML output: 2941 `\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.

```

2942 \newcommand*{\LinkHome}{%
2943 \LWR@subhyperrefclass{%
2944 \HomeHTMLFilename.html}%
2945 {Home}{linkhome}%
2946 }

```


`\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.

```
2947 \newcommand*{\LWR@topnavigation}{
2948 \LWR@htmlclassline{nav}{topnavigation}{\LinkHome}
2949 }
```

`\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.

```
2950 \newcommand*{\LWR@botnavigation}{
2951 \LWR@htmlclassline{nav}{botnavigation}{\LinkHome}
2952 }
```

```
2953 \end{warpHTML}
```

32 \PrintStack diagnostic tool



Diagnostics tool: Prints the LaTeX nesting depth values for the stack levels. Must have `\LWR@startpars` active while printing the stack, so `\PrintStack` may be called from anywhere in the normal text flow.

for HTML output: 2954 `\begin{warpHTML}`

`\PrintStack` Prints the closedepth stack.

```
2955 \newcommand*{\PrintStack}{
2956 \LWR@startpars
2957 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
2958 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
2959 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
2960 \LWR@closedephten{} \LWR@closedeptheleven{} \LWR@closedephtwelve{}
2961 }
```

```
2962 \end{warpHTML}
```

33 Closing stack levels

for HTML output: 2963 `\begin{warpHTML}`

Close one nested level:

```

2964 \newcommand*{\LWR@closeoneprevious}{%
2965
2966 \LWR@closeone{
2967
2968 \popclose{
2969 }

```

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

```

2970 \newcommand*{\LWR@closeprevious}[1]{

```

Close any pending paragraph:

```

2971 \LWR@stoppars

```

Close anything nested deeper than the desired depth:

```

2972 \whileof{not\(\LWR@closedepthone<#1\)}{\LWR@closeoneprevious}
2973 }

```

```

2974 \end{warpHTML}

```

34 Forcing a new PDF page

for HTML output: 2975 `\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.

```

2976 \newcommand{\LWR@forcenewpage}{%
2977 \LWR@stoppars\LWR@orignewpage\LWR@startpars%
2978 }

```

```

2979 \end{warpHTML}

```

35 HTML tags, spans, divs, elements

for HTML output: 2980 `\begin{warpHTML}`

35.1 Mapping L^AT_EX Sections to HTML Sections

```

2981 \newcommand*{\LWR@tagpart}{h2}
2982 \newcommand*{\LWR@tagpartend}{/h2}
2983 \newcommand*{\LWR@tagchapter}{h3}
2984 \newcommand*{\LWR@tagchapterend}{/h3}
2985 \newcommand*{\LWR@tagsection}{h4}
2986 \newcommand*{\LWR@tagsectionend}{/h4}
2987 \newcommand*{\LWR@tagsubsection}{h5}
2988 \newcommand*{\LWR@tagsubsectionend}{/h5}
2989 \newcommand*{\LWR@tagsubsubsection}{h6}
2990 \newcommand*{\LWR@tagsubsubsectionend}{/h6}
2991 \newcommand*{\LWR@tagparagraph}{\span class="paragraph"}}
2992 \newcommand*{\LWR@tagparagraphend}{/span}
2993 \newcommand*{\LWR@tagsubparagraph}{\span class="subparagraph"}}
2994 \newcommand*{\LWR@tagsubparagraphend}{/span}
2995
2996 \newcommand*{\LWR@tagregularparagraph}{p}

```

35.2 HTML tags

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

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

```

2997
2998 \newcommand*{\LWR@htmltagc}[1]{%
2999 {%
3000 \protect\LWR@origttfamily%
3001 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
3002 }%
3003 }

```

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.

```

3004 \newenvironment*{LWR@nestspan}
3005 {%
3006 \addtocounter{LWR@spandepth}{1}%
3007 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{}}%
3008 }%
3009 {\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.

```

3010 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
3011 \LWR@ensuredoingapar%
3012 \LWR@htmltagc{#1}%
3013 \begin{LWR@nestspan}%
3014 #2%
3015 \LWR@htmltagc{/#1}%
3016 \end{LWR@nestspan}%
3017 }

```

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

```

3018 \NewDocumentCommand{\LWR@htmlspanclass}{m o +m}{%
3019 \LWR@ensuredoingapar%
3020 \LWR@subhtmlclass{span}{#1}[#2]%
3021 \begin{LWR@nestspan}%
3022 #3%
3023 \LWR@htmltagc{/span}%
3024 \end{LWR@nestspan}%
3025 }

```

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

Print an HTML tag: `<tag>`

```

3026 \newcommand*{\LWR@htmltagb}[1]{%
3027 \LWR@htmltagc{#1}%
3028 \endgroup%
3029 }
3030
3031 \newcommand*{\LWR@htmltag}{%
3032 \begingroup\catcode'\_ =12
3033 \LWR@htmltagb%
3034 }

```

35.3 Block tags and comments

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

```

\LWR@htmlopencomment
\LWR@htmlclosecomment

```

```

3035 \newcommand*{\LWR@htmlopencomment}{%
3036 {\LWR@origttfamily\LWR@origtextless{!}{-}{-}}%
3037 }
3038
3039 \newcommand*{\LWR@htmlclosecomment}{%
3040 {\LWR@origttfamily{-}{-}\LWR@origtextgreater{}}%
3041 }

```

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

```

3042 \newcommand{\LWR@htmlcomment}[1]{%
3043 \LWR@htmlopencomment}%
3044 {%
3045 \LWR@origttfamily% break ligatures
3046 #1%
3047 }%
3048 \LWR@htmlclosecomment{}}

```

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

```

3049 \newcommand{\LWR@htmlblockcommentb}[1]
3050 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
3051
3052 \newcommand{\LWR@htmlblockcomment}
3053 {%
3054 \begingroup\catcode'\_ =12%
3055 \LWR@htmlblockcommentb%
3056 }

```

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

```

3057 \newcommand*{\LWR@htmlblocktag}[1]{%
3058 \LWR@stoppars%
3059 \LWR@htmltag{#1}%
3060 \LWR@startpars%
3061 }

```

35.4 Div class and element class

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

Factored and reused in several places.

```

3062 \NewDocumentCommand{\LWR@subhtmlclass}{m m o}{%
3063 \IfValueTF{#3}%

```

```

3064 {% option
3065 \ifthenelse{\equal{#3}{}}{%
3066 {\LWR@htmltag{#1 class="#2"}}% empty option
3067 {\LWR@htmltag{#1 class="#2" style="#3"}}% non-empty option
3068 }% option
3069 {\LWR@htmltag{#1 class="#2"}}% no option
3070 }

```

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

```

3071 \NewDocumentCommand{\LWR@htmlelementclass}{m m o}{%
3072 \LWR@stoppars%
3073 \LWR@subhtmlelementclass{#1}{#2}[#3]%
3074 \LWR@startpars%
3075 }

```

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

```

3076 \newcommand*{\LWR@htmlelementclassend}[2]{%
3077 \LWR@stoppars%
3078 \LWR@htmltag{/#1}%
3079 \ifbool{HTMLDebugComments}{%
3080 \LWR@htmlcomment{End of #1 ‘#2’}%
3081 }{}%
3082 \LWR@startpars%
3083 }

```

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

```

3084 \NewDocumentCommand{\LWR@htmldivclass}{m o}{%
3085 \LWR@htmlelementclass{div}{#1}[#2]%
3086 }

```

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

```

3087 \newcommand*{\LWR@htmldivclassend}[1]{%
3088 \LWR@htmlelementclassend{div}{#1}%
3089 }

```

35.5 Single-line elements

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

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

```

3090 \NewDocumentCommand{\LWR@htmlelementclassline}{m m o +m}{%
3091 \LWR@stoppars
3092 \LWR@subhtmlelementclass{#1}{#2}[#3]%
3093 #4%
3094 \LWR@htmltag{/#1}
3095 \LWR@startpars
3096 }

```

35.6 HTML5 semantic elements

`\LWR@htmlelement` $\{\langle element \rangle\}$

```

3097 \newcommand*{\LWR@htmlelement}[1]{%
3098 \LWR@htmlblocktag{#1}
3099 }

```

`\LWR@htmlelementend` $\{\langle element \rangle\}$

```

3100 \newcommand*{\LWR@htmlelementend}[1]{%
3101 \LWR@stoppars
3102 \LWR@htmltag{/#1}
3103 \LWR@startpars
3104 }
3105
3106 \end{warpHTML}

```

35.7 High-level block and inline classes

These are high-level commands which allow the creation of arbitrary block or inline sections which may be formatted with CSS.

For other direct-formatting commands, see section 68.

Env `BlockClass` $\{\langle class \rangle\}$ $[\langle style \rangle]$ High-level interface for div classes.

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

for HTML output:

```

3107 \begin{warpHTML}
3108 \NewDocumentEnvironment{BlockClass}{m o}
3109 {
3110 \LWR@htmldivclass{#1}[#2]
3111 }
```

```

3112 {
3113 \LWR@htmldivclassend{#1}
3114 }
3115 \end{warpHTML}

```

for PRINT output:

```

3116 \begin{warpprint}
3117 \NewDocumentEnvironment{BlockClass}{m o}{ }{}
3118 \end{warpprint}

```

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

for HTML output:

```

3119 \begin{warpHTML}
3120 \newcommand{\BlockClassSingle}[2]{%
3121 \LWR@htmlclassline{div}{#1}{#2}%
3122 }
3123 \end{warpHTML}

```

for PRINT output:

```

3124 \begin{warpprint}
3125 \newcommand{\BlockClassSingle}[2]{#2}
3126 \end{warpprint}

```

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

for HTML output:

```

3127 \begin{warpHTML}
3128 \NewDocumentCommand{\InlineClass}{m o +m}{%
3129 \LWR@htmlspanclass{#1}{#2}{#3}%
3130 }
3131 \end{warpHTML}

```

for PRINT output:

```

3132 \begin{warpprint}
3133 \NewDocumentCommand{\InlineClass}{m o +m}{#3}
3134 \end{warpprint}

```

35.8 Closing HTML tags

for HTML output:

```

3135 \begin{warpHTML}

```

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

```

3136 \newcommand*{\LWR@printclosepart}
3137 {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
3138 \newcommand*{\LWR@printclosechapter}
3139 {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
3140 \newcommand*{\LWR@printclosesection}

```



```

3141     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
3142 \newcommand*{\LWR@printclosesubsection}
3143     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
3144 \newcommand*{\LWR@printclosesubsubsection}
3145     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
3146 \newcommand*{\LWR@printcloseparagraph}
3147     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
3148 \newcommand*{\LWR@printclosesubparagraph}
3149     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}

```

Lists require closing HTML tags:

```

3150 \newcommand*{\LWR@printcloselistitem}
3151     {\LWR@htmltag{/li}}
3152 \newcommand*{\LWR@printclosedescitem}
3153     {\LWR@htmltag{/dd}}
3154 \newcommand*{\LWR@printcloseitemize}
3155     {\LWR@htmltag{/ul}}
3156 \newcommand*{\LWR@printcloseenumerate}
3157     {\LWR@htmltag{/ol}}
3158 \newcommand*{\LWR@printclosedescription}
3159     {\LWR@htmltag{/dl}}

3160 \end{warpHTML}

```

36 Paragraph handling

These commands generate the HTML paragraph tags when allowed and required.

Paragraph tags are or are not allowed depending on many conditions. Section 37 has high-level commands which allow paragraph-tag generation to start/stop. Even when allowed (`\LWR@doingstartpars`), tags are not generated until a \LaTeX paragraph is being used (`\LWR@doingapar`). `LWR@lateximagedepth` is used to prevent nesting tags inside a `lateximage`. `LWR@spandepth` is used to prevent nesting paragraph tags inside a paragraph, which became important inside `\fbox` commands and other spans.

for HTML output: 3161 `\begin{warpHTML}`

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

```

3162 \newcounter{LWR@spandepth}
3163 \setcounter{LWR@spandepth}{0}

```

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

```

3164 \newbool{LWR@doingstartpars}
3165 \boolfalse{LWR@doingstartpars}

```

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

```

3166 \newbool{LWR@doingapar}
3167 \global\boolfalse{LWR@doingapar}

```

\LWR@ensuredoinapar 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:

```

3168 \newcommand*{\LWR@ensuredoinapar}{%
3169 \ifbool{LWR@doingstartpars}%
3170 {\global\booltrue{LWR@doingapar}}%
3171 {}%
3172 }

```

\LWR@openparagraph

```

3173 \newcommand*{\LWR@openparagraph}
3174 {%

```

See if paragraph handling is enabled:

```

3175 \ifbool{LWR@doingstartpars}%
3176 {% handling pars

```

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

```

3177 \ifthenelse{%
3178 \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3179 \cnttest{\value{LWR@spandepth}}{>}{0}%
3180 }{% 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 ``.

```

3181 {}% no nested par tags

```

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

```

3182 {% yes nest par tags
3183 \LWR@htmltagc{\LWR@tagregularparagraph}%

```

Manually indent item list labels to avoid left margin intrusion:

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

See if are nested inside an item list:

```
3184 \ifnumcomp{\@listdepth}{>}{0}%
3185 {%
```

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

```
3186 \LWR@orighspace{1in}%
3187 }{}%
```

Now have started a paragraph.

```
3188 \global\booltrue{LWR@doingapar}%
```

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

```
3189 \let\par\LWR@closeparagraph%
3190 }% end of yes nest par tags
3191 }% end of handling pars
3192 {}% not handling pars
3193 }
```

`\LWR@closeparagraph`

```
3194 \newcommand*{\LWR@closeparagraph}
3195 {%
```

See if paragraph handling is enabled:

```
3196 \ifbool{LWR@doingapar}%
```

If currently in paragraph mode:

```
3197 {% handling pars
```

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

```
3198 \ifthenelse{%
3199 \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3200 \cnttest{\value{LWR@spandepth}}{>}{0}%
3201 }%
```

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

```

3202 {% no nested par tags
3203 \ifthenelse{%
3204 \cnttest{\value{LWR@spandepth}}{>}{0}%
3205 \AND%
3206 \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3207 }%
3208 {%
3209 \ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}%
3210 }%
3211 {}%
3212 }% no nested par tags

```

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

```

3213 {% yes nest par tags

```

Print a closing tag:

```

3214 \unskip%
3215 \LWR@htmltagc{/LWR@tagregularparagraph}%

```

No longer doing a paragraph:

```

3216 \global\boolfalse{LWR@doingapar}%
3217 % Disable the special \env{minipage} & \cs{hspace} interaction
3218 % until a new minipage is found:
3219 % \begin{macrocode}
3220 \global\boolfalse{LWR@minipagethispar}%
3221 }% end of yes nest par tags
3222 }% end of handling pars

```

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

```

3223 {% not handling pars
3224 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
3225 {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
3226 {}%
3227 }% not handling pars

```

Finish with regular paragraph processing

```

3228 \LWR@origpar%
3229 }

```

```

3230 \end{warpHTML}

```

37 Paragraph start/stop handling

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

Section 36 has the commands which actually generate the tags.

The `everyhook` package is used to generate the opening paragraph tags. The closing tags are generated by `\par`.

for HTML output: 3231 `\begin{warpHTML}`

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

```
3232 \newcommand*{\LWR@startpars}%
```

```
3233 {%
```

See if currently handling HTML paragraphs:

```
3234 \ifbool{\LWR@doingstartpars}%
```

If already in paragraph mode, do nothing.

```
3235 {}%
```

If not currently in paragraph mode:

```
3236 {%
```

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

```
3237 \PushPreHook{par}{\LWR@openparagraph}%
```

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

```
3238 \let\par\LWR@closeparagraph
```

```
3239
```

```
3240 }% an intentionally blank line
```

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

```
3241 \global\setbool{\LWR@doingstartpars}{true}%
```

No `<par>` tag yet to undo:

```
3242 \global\boolfalse{\LWR@doingapar}%
```

```
3243 }
```

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

```
3244 \newcommand*{\LWR@stoppars}%
```

```
3245 {%
```

See if currently handling HTML paragraphs:

```
3246 \ifbool{\LWR@doingapar}%
```

if currently in an HTML paragraph:

```
3247 {%
```

Print a closing tag:

```
3248 \unskip%
```

```
3249 \LWR@htmltagc{/ \LWR@tagregularparagraph}%
```

No longer have an open HTML paragraph:

```
3250 \global\boolfalse{\LWR@doingapar}%
```

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

```
3251 \global\boolfalse{\LWR@minipagethispar}
```

```
3252
```

```
3253 }% an intentionally blank line
```

If was not in an HTML paragraph:

```
3254 {}%
```

See if currently allowing HTML paragraphs:

```
3255 \ifbool{\LWR@doingstartpars}%
```

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

```
3256 {%
```

```
3257 \ClearPreHook{par}%
```

```
3258 }%
```

Else: do nothing

```
3259 {}%
```

no longer in paragraph mode

```

3260 \global\setbool{LWR@doingstartpars}{false}%

no <p> tag to undo:

3261 \global\boolfalse{LWR@doingapar}%
3262 }

3263 \end{warpHTML}

```

38 Page headers and footers

for HTML & PRINT: 3264 \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.

```

3265 \newcommand{\LWR@firstpagetop}{ } % for the home page alone
3266 \newcommand{\LWR@pagetop}{ } % for all other pages
3267 \newcommand{\LWR@pagebottom}{ }
3268
3269 \newcommand{\LWR@setfirstpagetopb}[1]{%
3270 \renewcommand{\LWR@firstpagetop}{#1}
3271 \catcode'\_ =8
3272 }

```

\HTMLFirstPageTop {*{text and logos}*}

```

3273 \newcommand{\HTMLFirstPageTop}{%
3274 \catcode'\_ =12
3275 \LWR@setfirstpagetopb
3276 }

```

```

3277 \newcommand{\LWR@setpagetopb}[1]{%
3278 \renewcommand{\LWR@pagetop}{#1}
3279 \catcode'\_ =8
3280 }

```

\HTMLPageTop {*{text and logos}*}

```

3281 \newcommand{\HTMLPageTop}{%
3282 \catcode'\_ =12
3283 \LWR@setpagetopb
3284 }

```

```

3285 \newcommand{\LWR@setpagebottom}[1]{%
3286 \renewcommand{\LWR@pagebottom}{#1}
3287 \catcode'\_ =8
3288 }

```

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

```

3289 \newcommand{\HTMLPageBottom}{%
3290 \catcode'\_ =12
3291 \LWR@setpagebottom
3292 }

3293 \end{warpall}

```

39 CSS

for HTML output: 3294 `\begin{warpHTML}`

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

```

3295 \newcommand*{\LWR@currentcss}{lwarp.css}

```

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

```

3296 \newcommand*{\LWR@newcssb}[1]{%
3297 \renewcommand*{\LWR@currentcss}{#1}
3298 \catcode'\_ =8
3299 }
3300
3301 \newcommand*{\CSSFilename}{
3302 \catcode'\_ =12
3303 \LWR@newcssb
3304 }
3305 \end{warpHTML}

```

for PRINT output: 3306 `\begin{warpprint}`
3307 `\newcommand*{\CSSFilename}[1]{}`
3308 `\end{warpprint}`

40 HTML meta description and author

for HTML & PRINT: 3309 \begin{warpall}

\HTMLAuthor {*\authorname*} The author to place into an HTML meta tag.

```
3310 \newcommand{\theHTMLAuthor}{\theauthor}
3311
3312 \newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}

3313 \end{warpall}
```

for HTML & PRINT: 3314 \begin{warpall}

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



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

\HTMLDescription {*\New HTML meta description.*} Assigns the HTML file's description meta tag.

```
3315 \newcommand{\LWR@currentHTMLDescription}{ }
3316
3317 \newcommand{\HTMLDescription}[1]{%
3318 \renewcommand{\LWR@currentHTMLDescription}{#1}
3319 }
3320
3321 \end{warpall}
```

41 Footnotes

lwarp 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 handle differently.

41.1 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 41.4 for the implementation.

41.2 Minipage footnotes

See section 67.2 for minipage footnotes.

41.3 Titlepage thanks

See section 48.6 for titlepage footnotes.

41.4 Regular page footnote implementation

for HTML output: 3322 \begin{warpHTML}

Patch L^AT_EX footnotes to use a new \box for lwarp footnotes.

```
3323 \newbox\LWR@footnotes
```

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

```
3324 \long\def\@makefntext#1{\textsuperscript{\@thefnmark} #1}
3325
3326 \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.

```
3327 \long\def\@footnotetext#1{%
3328 \global\setbox\LWR@footnotes=\vbox{\unvbox\LWR@footnotes%
3329 \protected@edef\@currentlabel{%
3330 \csname p@footnote\endcsname\@thefnmark%
3331 }% \@currentlabel
3332 \color@begingroup%
3333 \ifbool{LWR@doingstartpars}{\LWR@htmltagc{\LWR@tagregularparagraph}}%
3334 \@makefntext{#1}%
3335 \ifbool{LWR@doingstartpars}{\par}{\LWR@htmltagc{/LWR@tagregularparagraph}}%
3336 \color@endgroup%
3337 }% vbox
3338 }%
3339
3340 \long\def\@mpfootnotetext#1{%
```

```

3341 \global\setbox\@mpfootins\vbox{%
3342 \unvbox\@mpfootins
3343 \reset@font\footnotesize
3344 \hsize\columnwidth
3345 \@parboxrestore
3346 \protected@edef\@currentlabel
3347 {\csname p@mpfootnote\endcsname\@thefnmark}%
3348 \color@begingroup
3349 \@makefntext{%
3350 \ignorespaces#1%
3351 }%

```

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

```

3352 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
3353 {}%
3354 {\LWR@htmltagc{/\LWR@tagregularparagraph}}%

3355 \color@endgroup}%
3356 }

```

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

```

3357 \newcommand*\LWR@printpendingfootnotes{%
3358 \ifvoid\LWR@footnotes\else
3359 \LWR@forcenewpage
3360 \begin{BlockClass}{footnotes}
3361 \LWR@origmedskip
3362 \unvbox\LWR@footnotes
3363 \setbox\LWR@footnotes=\vbox{}
3364 \end{BlockClass}
3365 \fi
3366 }

```

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

```

3367 \newcommand*\LWR@epubprintpendingfootnotes{%
3368 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}%
3369 {\LWR@printpendingfootnotes}%
3370 {}%
3371 }

3372 \end{warpHTML}

```

42 Marginpars

for HTML output: 3373 \begin{warpHTML}

\marginpar [*left*] [*right*]

```
3374 \renewcommand{\marginpar}[2] [] {%
3375 \LWR@htmlspanclass{\marginpar}{#2}%
3376 }
```

```
3377 \end{warpHTML}
```

43 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: 3378 \begin{warpall}

Ctrl `FileDepth` [*section depth*] determines how deeply to break into new HTML files, similar to `tocdepth`. The default of -5 produces one large HTML file.

```
3379 \newcounter{FileDepth}
3380 \setcounter{FileDepth}{-5}
```

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

```
3381 \newbool{CombineHigherDepths}
3382 \booltrue{CombineHigherDepths}
```

```
3383 \end{warpall}
```

for HTML output: 3384 \begin{warpHTML}

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

```
3385 \newcommand*{\LWR@thisfilename}{}
```

`\LWR@thisnewfilename` The filename being sanitized.

```
3386 \newcommand*{\LWR@thisnewfilename}{}
```

`\LWR@filenameno blanks` $\{\langle filename \rangle\}$

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.

```
3387 \newcommand*{\LWR@filenameno blanks}[1]{%
3388 \begin{group}
```

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

```
3389 \renewcommand*{\HTMLUnicode}[1]{%
3390 \renewcommand*{\HTMLentity}[1]{##1}
3391 \renewcommand*{\LWR@htmltagc}[1]{%
3392 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}
```

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

```
3393 \renewcommand*{\&}{and}
3394 \renewcommand{\textit}[1]{##1}
3395 \renewcommand{\textsc}[1]{##1}
3396 \renewcommand{\textsl}[1]{##1}
3397 \renewcommand{\textbf}[1]{##1}
3398 \renewcommand{\texttt}[1]{##1}
3399 \renewcommand{\textsf}[1]{##1}
3400 \renewcommand{\textrm}[1]{##1}
3401 \renewcommand{\textsuperscript}[1]{##1}
3402 \renewcommand{\textsubscript}[1]{##1}
```

Replaces common symbols and short words with hyphens:

```
3403 \edef\LWR@thisnewfilename{#1}
3404 \fullexpandarg
```

Convert spaces into hyphens:

```
3405 \StrSubstitute{\LWR@thisnewfilename}{ }{-}\LWR@thisnewfilename]
```

Convert punctuation into hyphens:

```

3406 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
3407 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
3408 \StrSubstitute{\LWR@thisnewfilename}%
3409 {\LWR@origampersand}{-}[\LWR@thisnewfilename]
3410 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
3411 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
3412 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
3413 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
3414 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
3415 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
3416 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
3417 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
3418 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
3419 \StrSubstitute{\LWR@thisnewfilename}%
3420 {\textless}{-}[\LWR@thisnewfilename]
3421 \StrSubstitute{\LWR@thisnewfilename}%
3422 {\textgreater}{-}[\LWR@thisnewfilename]
3423 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
3424 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
3425 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
3426 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
3427 \StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
3428 \StrSubstitute{\LWR@thisnewfilename}%
3429 {\textbackslash}{-}[\LWR@thisnewfilename]
3430 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
3431 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
3432 %      "~{}" for babel
3433 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
3434 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
3435 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]

```

Convert short words:

```

3436 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
3437 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
3438 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
3439 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
3440 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
3441 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
3442 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
3443 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
3444 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
3445 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
3446 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
3447 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
3448 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
3449 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
3450 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
3451 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
3452 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]

```

```
3453 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]
```

Convert multiple hyphens:

```
3454 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
3455 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
3456 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
3457 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
3458 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
3459 %      emdash
3460 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
3461 %      endash
3462 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
3463 \endgroup
3464 }
```

`\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.

```
3465 \newcommand*{\LWR@newhtmlfile}[1]{
```

At the bottom of the ending file:

```
3466 \LWR@html elementclassend{section}{textbody}
3467
3468 \LWR@printpendingfootnotes
3469
```

No footer between files if EPUB:

```
3470 \ifbool{FormatEPUB}
3471 {}
3472 {
3473 \LWR@html element{footer}
3474
3475 \LWR@pagebottom
3476
3477 \LWR@html elementend{footer}
3478 }
```

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

```

3479 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3480 {}
3481 {\ifnumcomp{\value{LWR@htmlfilename}}{>}{0}{\LWR@botnavigation}{}}

```

End of this HTML file:

```

3482 \LWR@stoppars
3483 \LWR@htmltag{/body}\LWR@orignewline
3484 \LWR@htmltag{/html}\LWR@orignewline
3485 \LWR@orignewpage
3486
3487 \addtocounter{LWR@htmlfilename}{1}%

```

If using a filename, create a version without blanks. The filename without blanks will be placed into `\LWR@thisfilename`. If not using a filename, the file number will be used instead.

```

3488 \ifbool{FileSectionNames}%
3489 {\LWR@filenamenoblanks{#1}}
3490 {\renewcommand*{\LWR@thisfilename}{\theLWR@htmlfilename}}

```

Include an HTML comment to instruct lwarpmk where to split the files apart. Uses pipe-separated fields for `split_html.gawk`. Uses monospaced font with ligatures disabled for everything except the title.

```

3491 \LWR@htmlblockcomment{%
3492 |Start file|
3493 \LWR@htmlsectionfilename{\LWR@thisfilename}|
3494 }

```

At the top of the starting file:

```

3495 \LWR@stoppars
3496
3497 \LWR@filestart{ -- #1}% there is an Emdash in front of the #1
3498

```

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

```

3499 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3500 {}
3501 {\LWR@topnavigation}
3502

```

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

```

3503 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}

```



```

3504 {}
3505 {
3506 \LWR@html element{header}
3507
3508 \LWR@pagetop
3509
3510 \LWR@html elementend{header}
3511 }
3512

```

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

```

3513 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3514 {}
3515 {\ifcvoid{thetitle}{}\LWR@printthetitle}}
3516

```

No sideTOC if formatting for an EPUB or word processor:

```

3517 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3518 {}
3519 {\LWR@sidetoc}
3520

```

Start of the <tbody>:

```

3521 \LWR@html elementclass{section}{tbody}
3522

```

Keep paragraph tags disabled for now:

```

3523 \LWR@stoppars
3524

```

Track the page numbers:

```

3525 \setcounter{LWR@latestautopage}{\value{page}}
3526 }

3527 \end{warpHTML}

```

44 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing L^AT_EX code and packages. Formatting is handled by CSS, so the emulated code has much less work to do than the print versions.

Unicode Section names and the resulting filenames with accented characters are partially supported, depending on the ability of `pdflatex` to generate characters and `pdftotext` to read them. If extra symbols appear in the text, it may be that `pdflatex` is actually producing a symbol over or under a character, resulting in `pdftotext` picking up the accent symbol separately.



X_YLaTeX and LuaLaTeX directly support accented section and file names.

for HTML output: 3528 `\begin{warpHTML}`

44.1 Book class commands

`\mainmatter` Declare the main matter section of the document. Does not reset the page number, which must be consecutive arabic numbers for the HTML conversion.



```
3529 \newbool{LWR@mainmatter}
3530 \DeclareDocumentCommand{\mainmatter}{}{%
3531 \booltrue{LWR@mainmatter}%
3532 }
```

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

```
3533 \DeclareDocumentCommand{\frontmatter}{}{%
3534 \boolfalse{LWR@mainmatter}%
3535 }
```

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

```
3536 \DeclareDocumentCommand{\backmatter}{}{%
3537 \boolfalse{LWR@mainmatter}
3538 }
```

44.2 Sectioning support macros

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

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

```
3539 \newcommand*{\LWR@sectionnumber}[1]{%
3540 \InlineClass{sectionnumber}{#1} %
3541 }
```


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

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

Create an autosection tag.

```
3542 \newcommand*\LWR@createautosec}[1]{%
3543 \LWR@htmltag{#1 id="autosec-\thepage"}{}}%
3544 }
```


`\LWR@pushoneclose` $\{\langle depth\rangle\}$ $\{\langle printclose\rangle\}$ Stacks the new sectioning level's closing tag, to be used when this section is closed some time later.

 `\LWR@stoppars` must be executed first.

```
3545 \NewDocumentCommand{\LWR@pushoneclose}{m m}{\pushclose{#2}{#1}}
```

`\LWR@startnewdepth` $\{\langle depth\rangle\}$ $\{\langle printclose\rangle\}$

Closes currently stacked tags of a lesser level, then opens the new nesting level by saving this new sectioning level's closing tag for later use.

 `\LWR@stoppars` must be executed first.

```
3546 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
3547 \LWR@closeprevious{#1}%
```

Push a new section depth:

```
3548 \LWR@pushoneclose{#1}{#2}%
3549 }
```

`\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.

```
3550 \newcounter{LWR@prevFileDepth}
3551 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

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

The common actions for the high-level sectioning commands.

```
3552 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
3553 \LWR@traceinfo{LWR@section}%
3554 \LWR@stoppars%
```

Cancel special `minipage` horizontal space interaction:

```
3555 \global\boolfalse{LWR@minipagethispar}%
```

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

```
3556 \LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
3557 \IfBooleanTF{#1}{% starred
```

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

```
3558 \LWR@orignewpage%
3559
3560 }{% not starred
3561 \ifthenelse{%
3562 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}%
3563 \AND%
3564 \(%
3565 \NOT\boolean{CombineHigherDepths}\OR%
3566 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}%
3567 \)%
3568 }%
```

If so: start a new HTML file:

```
3569 {% new file
3570 \LWR@traceinfo{LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
3571 \IfNoValueTF{#2}%
```

If no optional entry

```
3572 {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
3573 {\LWR@newhtmlfile{#2}}%
3574 }% new file
```

Else: No new HTML file:

```
3575 {% not new file
```

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

```

3576 \LWR@orignewpage%
3577
3578 }% not new file
3579 }% not starred

```

Remember this section's name for \nameref:

```

3580 \LWR@traceinfo{\LWR@section: about to LWR@setlatestname}%
3581 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%

```

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

```

3582
3583 \ifbool{HTMLDebugComments}{%
3584 \LWR@htmlcomment{Opening #4 ‘‘#3’’}}%
3585 }{}
3586

```

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

```

3587 \ifthenelse{%
3588 \cnttest{\csuse{\LWR@depth#4}}{>=}{\LWR@depthparagraph}%
3589 }%
3590 {\LWR@startpars}
3591 {}

```

Create the opening tag with an autosec:

```

3592 \LWR@createautosec{\csuse{\LWR@tag#4}}%

```

If not starred, step counter and add to TOC:

```

3593 \IfBooleanTF{#1}%
3594 {}% starred
3595 {% not starred

```

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

```

3596 \ifthenelse{%
3597 \cnttest{\csuse{\LWR@depth#4}}{<=}{\value{secnumdepth}}%
3598 }%
3599 {% if secnumdepth

```

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

```

3600 \LWR@traceinfo{\LWR@section: about to test main matter}%
3601 \ifbool{\LWR@mainmatter}%
3602 {%

```

```
3603 \LWR@traceinfo{LWR@section: yes mainmatter}%
3604 \refstepcounter{#4}%
```

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

```
3605 \LWR@traceinfo{LWR@section: about to addcontentsline}%
3606 \addcontentsline{toc}{#4}%
3607 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
3608 \IfValueTF{#2}{#2}{#3}%
3609 \LWR@traceinfo{LWR@section: finished addcontentsline}%
3610 }% end of if main matter
```

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

```
3611 {% not main matter
3612 \LWR@traceinfo{LWR@section: no main matter}%
3613 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
3614 }% end of not main matter
3615 }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```
3616 {%
3617 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
3618 }%
```

For part, print the section type:

```
3619 \ifbool{LWR@mainmatter}%
3620 {%
3621 \ifthenelse{%
3622 \(\cnttest{\csuse{LWR@depth#4}}{<=}%
3623 {\value{secnumdepth}}\)\ \AND
3624 \(\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\)
3625 }%
3626 {\csuse{#4name}~{}}%
3627 }%
```

Print the section number:

```
3628 \LWR@traceinfo{LWR@section: about to print section number}%
3629 \ifthenelse{%
3630 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
3631 }%
3632 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
3633 }%
3634 \LWR@traceinfo{LWR@section: finished print section number}%
3635 }{}%
3636 }% end of not starred
```

Print the section name:

```
3637 #3
```

close the heading tag, such as /H2

```
3638 \LWR@htmltag{\csuse{LWR@tag#4end}}%
```

Generate a L^AT_EX label:

```
3639 \label{autopage-\thepage}%
```

Start paragraph handing unless is an inline paragraph or subparagraph:

```
3640 \ifthenelse{%
3641 \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}}%
3642 {\LWR@startpars}%
3643 {}%
```

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.

```
3644 \IfBooleanTF{#1}{}{% not starred
3645 \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
3646 }% not starred
3647 \LWR@traceinfo{LWR@section: done}%
3648 }
```

44.3 \section and friends

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

```
3649 \@ifundefined{chapter}
3650 {}
3651 {%
3652 \DeclareDocumentCommand{\part}{s o m}{%
3653 \LWR@epubprintpendingfootnotes%
3654 \LWR@stoppars%
3655
3656 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
3657
```

```

3658 \LWR@section{#1}{#2}{#3}{part}%
3659 }
3660 }

```

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

```

3661 \@ifundefined{chapter}
3662 {}
3663 {%
3664 \DeclareDocumentCommand{\chapter}{s o m}{%
3665 \LWR@traceinfo{chapter #3}%
3666 \LWR@epubprintpendingfootnotes%
3667 \LWR@stoppars%
3668 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
3669
3670 \LWR@traceinfo{chapter: about to LWR@section}%
3671 \LWR@section{#1}{#2}{#3}{chapter}%
3672 \LWR@traceinfo{chapter: done}%
3673 }
3674 }

```

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

```

3675 \DeclareDocumentCommand{\section}{s o m}{%
3676 \LWR@epubprintpendingfootnotes%
3677 \LWR@stoppars%
3678
3679 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
3680
3681 \LWR@section{#1}{#2}{#3}{section}%
3682 }

```

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

```

3683 \DeclareDocumentCommand{\subsection}{s o m}{%
3684 \LWR@epubprintpendingfootnotes%
3685 \LWR@stoppars%
3686
3687 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
3688
3689 \LWR@section{#1}{#2}{#3}{subsection}%
3690 }

```

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

```

3691 \DeclareDocumentCommand{\subsubsection}{s o m}{%
3692 \LWR@epubprintpendingfootnotes%

```



```

3693 \LWR@stoppars%
3694
3695 \LWR@startnewdepth{\LWR@depthsubsubsection}%
3696 {\LWR@printclosesubsubsection}%
3697
3698 \LWR@section{#1}{#2}{#3}{subsubsection}%
3699 }

```

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

```

3700 \DeclareDocumentCommand{\paragraph}{s o m}{%
3701 \LWR@epubprintpendingfootnotes%
3702 \LWR@stoppars%
3703
3704 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
3705
3706 \LWR@section{#1}{#2}{#3}{paragraph}%
3707 }

```

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

```

3708 \DeclareDocumentCommand{\subparagraph}{s o m}{%
3709 \LWR@epubprintpendingfootnotes%
3710 \LWR@stoppars%
3711
3712 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
3713
3714 \LWR@section{#1}{#2}{#3}{subparagraph}%
3715 }

```

```

3716 \end{warpHTML}

```

45 Starting a new file

for HTML & PRINT: 3717 \begin{warpall}

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

```

3718 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
3719
3720 \newcommand*{\HTMLLanguage}[1]{%
3721 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
3722 }

3723 \end{warpall}

```

for HTML output: 3724 \begin{warpHTML}

\LWR@filestart {\langle title_suffix \rangle}

Creates the opening HTML tags.

3725 \newcommand*{\LWR@filestart}[1]{

Locally temporarily disable direct-formatting commands:

3726 \begingroup

3727 \renewcommand{\textit}[1]{##1}% not used in filenames

3728 \renewcommand{\textsc}[1]{##1}

3729 \renewcommand{\textsl}[1]{##1}

3730 \renewcommand{\textbf}[1]{##1}

3731 \renewcommand{\texttt}[1]{##1}

3732 \renewcommand{\textsf}[1]{##1}

3733 \renewcommand{\textrm}[1]{##1}

3734 \renewcommand{\textsuperscript}[1]{##1}

3735 \renewcommand{\textsubscript}[1]{##1}

3736 \renewcommand*{\HTMLUnicode}[1]{}

3737 \renewcommand*{\HTMLentity}[1]{}

3738 \RenewDocumentCommand{\LWR@htmlspanclass}{m o +m}{##3}

3739 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}

Create the page's HTML header:

3740 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline

The language is user-adjustable:

3741 \LWR@htmltag{html lang="\LWR@currentHTMLLanguage"{}}\LWR@orignewline

Start of the meta data:

3742 \LWR@htmltag{head}\LWR@orignewline

Charset is fixed at UTF-8:

3743 \LWR@htmltag{meta charset="UTF-8" /}\LWR@orignewline

Author:

3744 \ifcsempy{theHTMLAuthor}{}{

3745 \LWR@htmltag{meta name="author" content="\theHTMLAuthor" /}\LWR@orignewline

3746 }

lwarp is the generator:

```
3747 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /}%
3748 \LWR@orignewline
```

If there is a description, add it now:

```
3749 \ifdefempty{\LWR@currentHTMLDescription}{}%
3750 \LWR@htmltag{%
3751 meta name="description" content="\LWR@currentHTMLDescription" /}%
3752 \LWR@orignewline
3753 }%
```

Mobile-friendly viewport:

```
3754 \LWR@htmltag{meta name="viewport" %
3755 content="width=device-width, initial-scale=1.0" /}%
3756 \LWR@orignewline
```

IE patch:

```
3757 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
3758 \LWR@htmltag{%
3759 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{}%
3760 \LWR@htmltag{/script}\LWR@orignewline
3761 \LWR@htmltag{![endif]{-}{-}}\LWR@orignewline
```

The page's title:

```
3762 \ifcsvoid{thetitle}{}%
3763 \LWR@htmltag{title}\thetitle#1\LWR@htmltag{/title}\LWR@orignewline%
3764 }%
```

The page's stylesheet:

```
3765 \LWR@htmltag{%
3766 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
3767 \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.

```
3768 \ifbool{mathjax}%
3769 {%
3770 \boolfalse{LWR@verbtags}
3771 \VerbatimInput{lwarp_mathjax.txt}%
3772 \booltrue{LWR@verbtags}
3773 \LWR@stoppars
```

```
3774 }% end of mathjax
3775 {}%
```

End of the header:

```
3776 \LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
3777 \LWR@htmltag{body}\LWR@orignewline
3778 \endgroup
3779 }
```

```
3780 \end{warpHTML}
```

46 Starting HTML output

for HTML output: 3781 \begin{warpHTML}

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

```
3782 \newcommand*{\LWR@LwarpStart}
3783 {%
```

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

```
3784 \ifbool{FormatWordProcessor}{%
3785 \setcounter{FileDepth}{-5}%
3786 \boolfalse{HTMLDebugComments}%
3787 }{}
```

Expand and detokenize \HomeHTMLFilename and \HTMLFilename:

```
3788 \edef\LWR@strresult{\HomeHTMLFilename}
3789 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
3790 \edef\LWR@strresult{\HTMLFilename}
3791 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn:

```
3792 \LWR@origonecolumn%
```

Reduce chance of line overflow in verbatim environments:

```
3793 \LWR@origscriptsize%
```

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

```
3794 \LWR@origraggedright%
3795 \let\\\LWR@endofline%
```

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

```
3796 \linespread{1.3}%
```

For `pdftotext` to reliably identify paragraph splits:

```
3797 \setlength{\parindent}{0pt}
3798 \setlength{\parskip}{2ex}
```

For the `lateximages` record file:

```
3799 \immediate\openout\LWR@file=lateximages.txt
```

Removes space after the caption in the HTML:

```
3800 \setlength{\belowcaptionskip}{-3ex}
```

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

```
3801 \renewcommand{\ps@plain}{}%
```

```
\centering Not used in the HTML environment:
\raggedleft
\raggedright 3802 \renewcommand*\centering{}{}
3803 \renewcommand*\raggedleft{}{}
3804 \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:

```
3805 \let\LWR@origtabular\tabular
3806 \let\LWR@origendtabular\endtabular
3807 \let\tabular\LWR@tabular
3808 \let\endtabular\endLWR@tabular%
```

Float captions:

```
3809 \let\LWR@origcaption\caption%
```

Labels: `\ltx@label` is used in `amsmath` environments and is also patched by `cleveref`.

Label in HTML

```

3810 \let\LWR@origltx@label\ltx@label
3811 \let\ltx@label\LWR@htmlmathlabel

```

Logos:

```

3812 \let\TeX\LWR@TeX
3813 \let\LaTeX\LWR@LaTeX
3814 \let\LuaTeX\LWR@LuaTeX
3815 \let\LuaLaTeX\LWR@LuaLaTeX
3816 \let\XeTeX\LWR@XeTeX
3817 \let\XeLaTeX\LWR@XeLaTeX
3818 \let\ConTeXt\LWR@ConTeXt

```

Graphics:

```

3819 \let\rotatebox\LWR@rotatebox
3820 \let\scalebox\LWR@scalebox
3821 \let\reflectbox\LWR@reflectbox

```

Not yet started any paragraph handling:

```

3822 \global\boolfalse\LWR@doingapar}
3823 \global\boolfalse\LWR@doingstartpars}

```

Start a new HTML file and a header:

```

3824 \LWR@filestart{}
3825 \LWR@htmltag{header}\LWR@orignewline
3826 \LWR@startpars
3827 \LWR@firstpagetop
3828 \LWR@stoppars
3829 \LWR@htmltag{/header}\LWR@orignewline
3830 \LWR@htmltag{section class="textbody"{} }
3831 \LWR@origpagestyle{empty}

```

Document and page settings:

```

3832 \mainmatter
3833 \LWR@origpagenumbering{arabic}

```

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

```

3834 \if@titlepage
3835   \thanksmarkseries{arabic}
3836 \else
3837   \thanksmarkseries{fnsymbol}
3838 \fi

```

Initial default patch for fancyvrb:

```
3839 \fvset{frame=none}%
```

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

```
3840 \catcode'\&=\active
```

Allow HTML paragraphs to begin:

```
3841 \LWR@startpars
3842 }
```

```
3843 \end{warpHTML}
```

47 Ending HTML output

for HTML output: 3844 \begin{warpHTML}

`\LWR@requesttoc` $\{\langle boolean \rangle\} \{\langle suffix \rangle\}$ Requests that a toc, lof, or lot be generated.

```
3845 \newcommand*{\LWR@requesttoc}[2]{%
3846 \ifbool{#1}
3847 {
3848 \expandafter\newwrite\csuse{tf@#2}
3849 \immediate\openout \csuse{tf@#2} \jobname.#2\relax
3850 }{}
3851 }
```

`\LWR@LwarpEnd` Final stop of all HTML output:

```
3852 \newcommand*{\LWR@LwarpEnd}
3853 {
3854 \LWR@stoppars
3855 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
3856 \LWR@html-elementclassend{section}{textbody}
```

Print any pending footnotes:

```
3857 \LWR@printpendingfootnotes
```

Create the footer:

```
3858 \LWR@htmlelement{footer}
3859
3860 \LWR@pagebottom
3861
3862 \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.

```
3863 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}{
3864 {}
3865 {
3866 \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{
3867 }
```

```
3868 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
3869 \LWR@htmltag{/body}\LWR@orignewline
3870 \LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
3871 \LWR@orignewpage
```

For lateximage commands:

```
3872 \immediate\closeout\LWR@file
3873 }
```

```
3874 \end{warpHTML}
```

48 Titles and the titling package

Supports and extends the titling package.

Additional functions include `\published` and `\subtitle`, and the `\author` command has an additional `\affiliation` command to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using `\theauthor` in the main text.

The `titling` package maintains the definitions of `\thetitle`, `\theauthor`, etc., after the title has been typeset. These commands are to be used to refer to the document's title and author, etc., in the main text. These definitions have the `\thanks` and `\affiliation` removed, and for author the `\and` is replaced to generate a simple inline list of authors separated by commas.

`\printtitle`, `\printauthor`, etc., are provided for use inside the `titlepage` or `titlingpage` environments, and these retain the `\thanks` and `\affiliation`.

Several additional hooks are provided in addition to `titling`:

<code>\maketitlehookaa</code>	<code>\maketitlehookaa</code> : Between “published” and the title.
<code>\maketitlehookaaa</code>	<code>\maketitlehookaaa</code> : Between the title and the subtitle.
<code>\prepublished</code>	<code>\prepublished</code> : Before the “published” field.
<code>\postpublished</code>	<code>\postpublished</code> : After the “published” field.
<code>\presubtitle</code>	<code>\presubtitle</code> : Before the subtitle.
<code>\postsubtitle</code>	<code>\postsubtitle</code> : After the subtitle.
<code>\printthanks</code>	<code>\printthanks</code> has been added to force the printing of thanks inside a <code>titlingpage</code> environment when <code>\maketitle</code> is not used.

No footnotes!

Inside a `\titlepage` or `\titlingpage` environment, use `\thanks` for footnotes, do not use `\footnote`.

At the end of the `titlingpage` environment, footnote marks are forced to reset to zero.

Inside a `titlingpage` environment with the `article` document class, thanks marks will be `fnsymbol` instead of `arabic`. `arabic` is usually used when inside `titlepage` environments where the title page is on its own page, but is not automatically used inside a `titlingpage` environment.

To force the thanks marks to be `arabic`:

```
\begin{titlingpage}
\thanksmarkseries{arabic}
...
```

48.1 Setting the title, etc.

The following provide setting commands for both HTML and print outputs.

<code>\published</code>	<code>\@title</code> , <code>\@subtitle</code> , <code>\@author</code> , etc. store the values as originally assigned, including any <code>\thanks</code> , <code>\and</code> , or <code>\affiliation</code> . These are low-level macros intended to be used by other macros only inside a <code>titlepage</code> or <code>titlingpage</code> , and are used by <code>\maketitle</code> . The author is printed inside a single-column table, which becomes multiple single-column tables if multiples authors are included.
<code>\title</code>	
<code>\subtitle</code>	
<code>\author</code>	
<code>\date</code>	
<code>\printpublished</code>	<code>\printtitle</code> , <code>\printsubtitle</code> , etc. are user-level macros intended to be used in <code>titlepage</code> and <code>titlingpage</code> environments in cases where <code>\maketitle</code> is not desired. These commands preserve the <code>\thanks</code> , etc., and should not be used in the main text. The author is printed inside a single-column table, which becomes multiple single-column tables if multiples authors are included.
<code>\printtitle</code>	
<code>\printsubtitle</code>	
<code>\printauthor</code>	
<code>\printdate</code>	
<code>\thepublished</code>	<code>\thetitle</code> , <code>\thesubtitle</code> , <code>\theauthor</code> , etc. are user-level sanitized versions which have removed the <code>\thanks</code> and <code>\affiliation</code> , and <code>\and</code> is changed for inline text usage. The author is printed inline without <code>\affiliation</code> or <code>\thanks</code> , with <code>\and</code> placing commas between multiple authors. Thus, these commands are to be used in the main text whenever the user wishes to refer to the document's title and such. One practical use for this is to place the authors at the bottom of each HTML page, such as:
<code>\thetitle</code>	
<code>\thesubtitle</code>	
<code>\theauthor</code>	
<code>\thedata</code>	
<code>\HTMLPageBottom</code>	<code>{\text}</code>

```
\HTMLPageBottom{
\begin{center}\textcopyright~2016 \theauthor\end{center}
}
```

`\author` `{\author}` While using `\maketitle`, the author is treated as a single-column table and the `\and` feature finishes the current table then starts a new one for the next author. Each author thus is placed into its own table, and an affiliation may be placed on its own line such as

```
\author{Name \\ Affiliation \and Second Name \\ Second Affiliation}
```

After `\maketitle` has completed, `\theauthor` retains the definition of the author, but `\and` is changed to become a comma and a space, intending to print the authors names separated by spaces. This fails when affiliations are included on their own table rows.

A solution, provide here, is to define a macro `\affiliation` which during `\maketitle` starts a new table row and adds the affiliation, but after `\maketitle` is finished `\affiliation` is re-defined to throw away its argument, thus printing only the author names when `\author` is later used inline.

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

Adds the affiliation to the author for use in `\maketitle`. Nullified when later used for inline use of `\theauthor`.

for HTML output: 3875 `\begin{warpHTML}`
 3876 `\newrobustcmd{\affiliation}[1]{\ \InlineClass{affiliation}{#1}}`
 3877 `\end{warpHTML}`

for PRINT output: 3878 `\begin{warpprint}`
 3879 `\newrobustcmd{\affiliation}[1]{\ \textsc{small#1}}`
 3880 `\end{warpprint}`

The following are based on the original titling code:

for HTML & PRINT: 3881 `\begin{warpall}`

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

Redefined to nullify `\affiliation`, etc. before printing the authors inline.

`\@author` retains the entire author with its `\thanks`, while `\theauthor` will have `\thanks` removed and `\and` simplified.

```
3882 \renewcommand{\author}[1]{%
3883 \gdef\@author{#1}
3884 \begingroup
3885   \renewcommand{\thanks}[1]{ }
3886   \renewcommand{\and}{\unskip, }
3887   \renewcommand{\thanksmark}[1]{ }
3888   \renewcommand{\thanksgap}[1]{ }
3889   \renewcommand{\affiliation}[1]{ }
3890   \protected@xdef\theauthor{#1}
3891 \endgroup}
```

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

```
3892 \newcommand{\published}[1]{%
3893 \gdef\@published{#1}
3894 \begingroup
3895   \renewcommand{\thanks}[1]{ }
3896   \renewcommand{\thanksmark}[1]{ }
3897   \renewcommand{\thanksgap}[1]{ }
3898   \protected@xdef\thepublished{#1}
3899 \endgroup
3900 }
3901 \newcommand{\@published}{ }
3902 \newcommand{\thepublished}{ }
```

```

\subtitle {{text}}

3903 \newcommand{\subtitle}[1]{%
3904 \gdef\@subtitle{#1}
3905 \begingroup
3906   \renewcommand{\thanks}[1]{%
3907     \renewcommand{\thanksmark}[1]{%
3908       \renewcommand{\thanksgap}[1]{%
3909         \protected@xdef\thesubtitle{#1}
3910       \endgroup
3911     }
3912 \newcommand{\@subtitle}{%
3913 \newcommand{\thesubtitle}{%

3914 \end{warpall}

```

48.2 Changes to HTML titlepage and titlingpage

for HTML output: 3915 \begin{warpHTML}

Env titlepage Sets up a titlepage div with a L^AT_EX PDF minipage inside.

```

3916 \renewenvironment*{titlepage}
3917 {
3918 \LWR@forcenewpage
3919 \BlockClass{titlepage}\LWR@subminipage
3920 }
3921 {\LWR@endsubminipage\endBlockClass}

```

Env titlingpage

```

3922 \renewenvironment*{titlingpage}
3923 {%

```

Start an HTML titlepage div:

```

3924 \begin{titlepage}

```

Prepare for a custom version of \maketitle inside the titlingpage:

```

3925 \LWR@maketitlesetup
3926 \let\maketitle\LWR@titlingmaketitle
3927 }
3928 {

```

At the end of the environment, end the HTML titlepage div:

```
3929 \end{titlepage}
```

Reset the footnote counter:

```
3930 \@bscontmark
3931 }
```

```
3932 \end{warpHTML}
```

for HTML & PRINT: 3933 \begin{warpall}

`\printthanks` Forces the `\thanks` to be printed.

This is necessary in a `titlingpage` environment when `\maketitle` was not used.

```
3934 \newcommand*{\printthanks}{\@thanks}
```

Env `titlingpage` At the end of the titlingpage for both print and HTML, reset footnote markers to zero.

```
3935 \AtEndEnvironment{titlingpage}{\@bscontmark}
```

```
3936 \end{warpall}
```

48.3 Printing the title, etc. in HTML

The following are for printing the title, etc. in a `titlepage` or a `titlingpage` in HTML:

for HTML output: 3937 \begin{warpHTML}

Patch the pre/post title/author/date to add HTML tags, then initialize:

```
3938 \newcommand{\prepublished}[1]{%
3939 \def\@bspublished{\BlockClass{published}#1}%
3940 }
3941
3942 \newcommand{\postpublished}[1]{%
3943 \def\@bspublished{#1\endBlockClass}%
3944 }
3945
3946 \renewcommand{\pretitled}[1]{%
3947 \def\@bspublished{#1\LWR@stoppars\LWR@htmltag{h1}}}%
3948 }
```

```

3949
3950 \renewcommand{\posttitle}[1]{%
3951 \def\@bsposttitle{\LWR@htmltag{/h1}\LWR@startpars#1}%
3952 }
3953
3954 \newcommand{\presubtitle}[1]{%
3955 \def\@bspresubtitle{\BlockClass{subtitle}#1}%
3956 }
3957
3958 \newcommand{\postsubtitle}[1]{%
3959 \def\@bspostsubtitle{#1\endBlockClass}%
3960 }
3961
3962 \renewcommand{\preauthor}[1]{%
3963 \def\@bspreauthor{\BlockClass{author}#1}%
3964 }
3965
3966 \renewcommand{\postauthor}[1]{%
3967 \def\@bspostauthor{#1\endBlockClass}%
3968 }
3969
3970 \renewcommand{\predate}[1]{%
3971 \def\@bspredate{#1\BlockClass{titledate}}%
3972 }
3973
3974 \renewcommand{\postdate}[1]{%
3975 \def\@bspostdate{\endBlockClass#1}%
3976 }
3977
3978 \prepublished{\begin{center}}
3979 \postpublished{\par\end{center}}
3980
3981 \pretitle{\begin{center}}
3982 \posttitle{\par\end{center}}
3983
3984 \presubtitle{\begin{center}}
3985 \postsubtitle{\par\end{center}}
3986
3987 \preauthor{\begin{center}}%
3988 \begin{tabular}[t]{c}%
3989 }
3990 \postauthor{\end{tabular}\par\end{center}}
3991
3992 \predate{\begin{center}}
3993 \postdate{\par\end{center}}

```

\printpublished

```

3994 \newcommand*{\printpublished}{

```

```

3995 \ifthenelse{\equal{\thepublished}{}}
3996 {}
3997 {
3998 \begin{BlockClass}{published}
3999 \@published
4000 \end{BlockClass}
4001 }
4002 }

```

\printtitle

```

4003 \newcommand*{\printtitle}
4004 {
4005 \LWR@stoppars
4006 \LWR@htmltag{h1}%
4007 \@title%
4008 \LWR@htmltag{/h1}
4009 \LWR@startpars
4010 }

```

\LWR@printthetitle A private version which prints the title without footnotes, used to title each HTML page.

```

4011 \newcommand*{\LWR@printthetitle}
4012 {
4013 \LWR@stoppars
4014 \LWR@htmltag{h1}%
4015 \thetitle%
4016 \LWR@htmltag{/h1}
4017 \LWR@startpars
4018 }

```

\printssubtitle

```

4019 \newcommand*{\printssubtitle}{
4020 \ifthenelse{\equal{\thesubtitle}{}}
4021 {}
4022 {
4023 \begin{BlockClass}{subtitle}
4024 \@subtitle
4025 \end{BlockClass}
4026 }
4027 }

```

\printauthor

```

4028 \newcommand*{\printauthor}{

```

```

4029 \begin{BlockClass}{author}
4030 \begin{tabular}{c}\@author\end{tabular}
4031 \end{BlockClass}
4032 }

```

`\printdate`

```

4033 \newcommand*\printdate{%
4034 \begin{BlockClass}{titledate}
4035 \@date
4036 \end{BlockClass}
4037 }

```

```

4038 \end{warpHTML}

```

48.4 Printing the title, etc. in print form

The following are for printing the title, etc. in a `titlepage` or a `titlingpage` in print form:

for PRINT output: 4039 `\begin{warpprint}`

`\printpublished`

```

4040 \newcommand*\printpublished{{\Large\scshape\@published}}

```

`\printtitle`

```

4041 \newcommand*\printtitle{{\Huge\@title}}

```

`\printsubtitle`

```

4042 \newcommand*\printsubtitle{{\Large\itshape\@subtitle\bigskip}}

```

`\printauthor`

```

4043 \newcommand*\printauthor
4044     {{\large\begin{tabular}{t}{c}\@author\end{tabular}}}

```

`\printdate`

```

4045 \newcommand*\printdate{{\small\textit{\@date}}}

```


48.5 \maketitle for print output

`\maketitle` From the titling package, patched to add the publisher and subtitle.

```

4046 \providecommand{\maketitle}{}
4047 \if@titlepage
4048   \renewcommand{\maketitle}{\begin{titlepage}%
4049     \let\footnotesize\small
4050     \let\footnoterule\relax
4051     \let \footnote \thanks
4052     \@bsmarkseries
4053     \def\@makefnmark{\rlap{\@textsuperscript{%
4054       \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
4055     \long\def\@makefntext##1{\makethanksmark ##1}
4056     \null\vfil
4057     \vskip 60\p@
4058     \vspace*{\droptitle}
4059     \maketitlehooka
4060     \ifcsempy{\@published}
4061   {}
4062   {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
4063   {\@bspretitle \@title \@bsposttitle}
4064   \ifcsempy{\@subtitle}
4065 {}
4066 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
4067   \maketitlehookb
4068   {\@bspreauthor \@author \@bspostauthor}
4069   \maketitlehookc
4070   {\@bspredate \@date \@bspostdate}
4071   \maketitlehookd
4072   \par
4073   \@thanks
4074   \vfil\null
4075   \end{titlepage}%
4076   \@bscontmark % \setcounter{footnote}{0}%
4077   %%% \@bsmtitleempty
4078 } % end titlepage defs
4079 \else
4080   \renewcommand{\maketitle}{\par
4081     \begingroup
4082       \@bsmarkseries
4083       \def\@makefnmark{\rlap{\@textsuperscript{%
4084         \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
4085       \long\def\@makefntext##1{\makethanksmark ##1}
4086       \if@twocolumn
4087         \ifnum \col@number=\@ne
4088           \@maketitle
4089         \else
4090           \twocolumn[\@maketitle]%

```

```

4091         \fi
4092     \else
4093         \newpage
4094         \global\@topnum\z@
4095         \@maketitle
4096     \fi
4097     \thispagestyle{plain}\@thanks
4098 \endgroup
4099 \@bscontmark % \setcounter{footnote}{0}%
4100 %% \@bsmtitleempty
4101 } % end non-titlepage
4102
4103 \def\@maketitle{%
4104     \newpage
4105     \null
4106     \vskip 2em%
4107     \vspace*{\droptitle}
4108     \maketitlehooka
4109     \ifcseempty{@published}
4110 {}
4111 {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
4112 {\@bspretitle \@title \@bsposttitle}
4113 \ifcseempty{@subtitle}
4114 {}
4115 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
4116 \maketitlehookb
4117 {\@bspreadauthor \@author \@bspostauthor}
4118 \maketitlehookc
4119 {\@bspredate \@date \@bspostdate}
4120 \maketitlehookd
4121 \par
4122 \vskip 1.5em}
4123 \fi
4124
4125 \providecommand{\maketitlehookaa}{}
4126
4127 \providecommand{\maketitlehookaaa}{}
4128
4129 \newcommand{\prepublished}[1]{%
4130 \def\@bsprepublished{#1}%
4131 }
4132
4133 \newcommand{\postpublished}[1]{%
4134 \def\@bspostpublished{#1}%
4135 }
4136
4137 \newcommand{\presubtitle}[1]{%
4138 \def\@bspresubtitle{#1}%
4139 }

```

`\presubtitle` Hook after printing the subtitle.

```

4140 \newcommand{\postsubtitle}[1]{%
4141 \def\@bspostsubtitle{#1}%
4142 }

Initial settings:

4143 \if@titlepage
4144 \prepublished{
4145 \vspace*{-\baselineskip}\vspace*{-\medskipamount}\vspace*{-2em}
4146 \begin{center}}
4147 \postpublished{\par\end{center}\vskip 2em}
4148
4149 \presubtitle{\unskip\begin{center}\unskip}
4150 \postsubtitle{\par\end{center}\vskip 2em}
4151 \else
4152 \prepublished{\begin{center}}
4153 \postpublished{\par\end{center}\vskip 0.5em}
4154
4155 \presubtitle{\begin{center}\unskip}
4156 \postsubtitle{\par\end{center}\vskip 0.5em}
4157 \fi

4158 \end{warpprint}

```

48.6 `\maketitle` for HTML output

An HTML div of class `titlepage` is created, inside of which a L^AT_EX PDF minipage is generated (without HTML tags), allowing the `\thanks` footnotes to be generated immediately at the end of the title page during HTML output. This is desirable when a large table of contents immediately follows the title.

`\thanks` are a form of footnotes used in the title page. See section 41 for other kinds of footnotes.

See `\thanksmarkseries{series}`, below, to set the style of the footnote marks.

for HTML output: 4159 `\begin{warppHTML}`

`\LWR@maketitlesetup` Patches `\thanks` macros to use L^AT_EX minipage footnotes.

```

4160 \newcommand*\LWR@maketitlesetup{%

```

Select which kind of footnote marks to use:

```
4161 \@bsmarkseries
4162 \@mpbsmarkseries
```

Redefine the footnote mark:

```
4163 \def\@makefnmark{\textsuperscript{\thefootnote}}

\thefootnote ⇒ \nameuse{arabic}{footnote}, or
\thefootnote ⇒ \nameuse{fnsymbol}{footnote}
```

Redefine the footnote text:

```
4164 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
4165 \makethanksmark \LWR@origspace{1in}

\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒
\@thefnmark ⇒ \itshape a (or similar)
```

Print the text:

```
4166 ##1%
4167 }%
4168 }
```

`\@fnsymbol` $\{\langle counter \rangle\}$

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used `\|` which was not being found by `pdftotext`.

```
4169 \def\@fnsymbol#1{\ensuremath{\ifcase#1\or *\or \dagger\or \ddagger\or
4170 \mathsection\or \mathparagraph\or \text{\HTMLUnicode{2016}}\or
4171 **\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.

```
4172 \renewcommand*\maketitle{%
```

An HTML titlepage div is used for all classes.

```
4173 \begin{titlepage}
```

Set up special patches:

```
4174 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4175 \@maketitle
```

Immediately generate any \thanks footnotes:

```
4176 \@thanks
```

Close the HTML titlepage div:

```
4177 \end{titlepage}
```

Reset the footnote counter:

```
4178 \@bscontmark
4179 }
```

\@maketitle Typesets the title, etc. for HTML:

```
4180 \DeclareDocumentCommand{\@maketitle}{-}{%
4181 \maketitlehooka
4182 \ifcempty{@published}
4183 {}
4184 {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
4185 {\@bsprettitle \@title \@bsposttitle}
4186 \ifcempty{@subtitle}
4187 {}
4188 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
4189 \maketitlehookb
4190 {\@bspreauthor \@author \@bspostauthor}
4191 \maketitlehookc
4192 {\@bspredate \@date \@bspostdate}
4193 \maketitlehookd
4194 }

4195 \providecommand{\maketitlehookaa}{}
4196 \providecommand{\maketitlehookaaa}{}

```

\LWR@titlingmaketitle \maketitle for use inside an HTML titlingpage environment.

```
4197 \newcommand*{\LWR@titlingmaketitle}{%
```

Typeset the title, etc:

4198 `\@maketitle`

Immediately generate any `\thanks` footnotes:

4199 `\@thanks`
4200 `}`

`\thanksmarkseries` `{\series}`

Sets the type of footnote marks used by `\thanks`, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc. Modified to use the L^AT_EX PDF minipage which is included with the title page.

4201 `\renewcommand{\thanksmarkseries}[1]{%`
4202 `\def\@mpbsmarkseries{%`
4203 `\renewcommand*\{thempfootnote}\@nameuse{#1}{mpfootnote}}}%`
4204 `\def\@bsmarkseries{\renewcommand{thefootnote}\@nameuse{#1}{footnote}}}%`
4205 `}`

4206 `\end{warpHTML}`

49 Abstract

The following code replaces the L^AT_EX default, and will itself be replaced later if the `abstract` package is loaded.

for HTML output: 4207 `\begin{warpHTML}`

`\abstractname` User-redefinable title for the abstract.

Also over-written by the `babel` package.

4208 `\providecommand*\{abstractname}{Abstract}`

Env `abstract`

4209 `\DeclareDocumentEnvironment{abstract}{}%`
4210 `{`
4211 `\LWR@forcenewpage`
4212 `\BlockClass{abstract}`
4213 `\BlockClassSingle{abstracttitle}\{abstractname}`
4214 `}`
4215 `{`
4216 `\endBlockClass`
4217 `}`

```
4218 \end{warpHTML}
```

50 Quote and verse

50.1 Citations and attributions

`\attribution` for use inside quote, quotation, verse:

ex: `\attribution{author name} --- \citetitle{book name}`

for HTML output: 4219 `\begin{warpHTML}`
 4220 `\newcommand{\attribution}[1]{%`
 4221 `\InlineClass{attribution}{--\,#1}}% emdash`
 4222 `\end{warpHTML}`

for PRINT output: 4223 `\begin{warpprint}`
 4224 `\newcommand{\attribution}[1]{\textsc{---\,#1}}`
 4225 `\end{warpprint}`

`\citetitle` for use inside quote, quotation, verse:

for HTML output: 4226 `\begin{warpHTML}`
 4227 `\newcommand{\citetitle}[1]{%`
 4228 `\InlineClass{citetitle}{--\,#1}}% emdash`
 4229 `\end{warpHTML}`

for PRINT output: 4230 `\begin{warpprint}`
 4231 `\newcommand{\citetitle}[1]{\textsl{---\,#1}}`
 4232 `\end{warpprint}`

50.2 Quotes, quotations

for HTML output: 4233 `\begin{warpHTML}`

Env `quote`

```
4234 \renewenvironment*{quote}
4235 {
4236 \LWR@forcenewpage
4237 \LWR@htmlblocktag{blockquote}
4238 }
4239 {\LWR@htmlblocktag{/blockquote}}
```

```

4240
4241 \renewenvironment*{quotation}
4242 {
4243 \LWR@forcenewpage
4244 \LWR@htmlblocktag{blockquotation}
4245 }
4246 {\LWR@htmlblocktag{/blockquotation}}

4247 \end{warpHTML}

```

50.3 Verse

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

```

\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}

```

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

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

for HTML & PRINT: 4248 `\begin{warpall}`

The following lengths may be set in either print or HTML output, but are only used in HTML. This allows the user to set `\vleftskip` and `\leftmargini` for print output, and optionally select different values for HTML.

<p>Len <code>\TMLvleftskip</code></p>	<p>Sets <code>\vleftskip</code> inside a <code>verse</code> environment in HTML.</p>
---------------------------------------	--


```

4249 \newlength{\HTMLvleftskip}
4250 \setlength{\HTMLvleftskip}{1em}

```

Len `\TMLleftmargini` Sets `\leftmargini` inside a verse environment in HTML.

```

4251 \newlength{\HTMLleftmargini}
4252 \setlength{\HTMLleftmargini}{4.5em}

4253 \end{warpall}

```

51 Verbatim

for HTML output: 4254 `\begin{warpHTML}`

Env `verbatim`

```

4255 \AfterEndPreamble{
4256 \AtBeginEnvironment{verbatim}{%
4257 \LWR@forcenewpage
4258 \LWR@atbeginverbatim{verbatim}\unskip\vspace*{-\baselineskip}%
4259 }
4260 \AfterEndEnvironment{verbatim}{\unskip\vspace*{-\baselineskip}\LWR@afterendverbatim}
4261 }

4262 \end{warpHTML}

```

52 Fancyvrb

for HTML & PRINT: 4263 `\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.

```

4264 \newlength{\VerbatimHTMLWidth}
4265 \setlength{\VerbatimHTMLWidth}{4in}
4266 \end{warpall}

```

for HTML output: 4267 `\begin{warpHTML}`

Bool **LWR@verbtags** Used to temporarily turn off verbatim tags while doing **VerbatimInput** in the HTML head.

```
4268 \newbool{LWR@verbtags}
4269 \booltrue{LWR@verbtags}
```

For **\VerbatimFootnotes**:

```
4270 \renewcommand{\VerbatimFootnotes}{
4271 \PackageError{lwarp}
4272 {Verbatim footnotes are not yet supported by lwarp.}
4273 {This may be improved some day.}
4274 }
```

\LWR@atbeginverbatim $\{ \langle class \rangle \}$

Encloses a verbatim environment with the given CSS class.

```
4275 \newcommand*{\LWR@atbeginverbatim}[1]
4276 {%
```

Avoid excessive space between lines:

```
4277 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
4278 \LWR@stoppars%
```

Create a new **pre** of the given class:

```
4279 \ifbool{LWR@verbtags}{\LWR@htmltag{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.

```
4280 \LWR@origttfamily%
```

Do not produce HTML tags for **\hspace** inside a verse **par**. Restore plain **L^AT_EX** **\hspace** functionality:

```
4281 \let\hspace\LWR@orighspace%
4282 }
```

\LWR@afterendverbatim Finishes enclosing a verbatim environment.

```
4283 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the `pre`:

```
4284 \unskip%
```

At the end of the environment, close the `pre`:

```
4285 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}}
4286
4287 }{}%
```

Resume regular paragraph handling:

```
4288 \LWR@startpars%
4289 }
```

`\LWR@Verbatimclass` Holds the class of the following verbatim.

```
4290 \newcommand*{\LWR@Verbatimclass}{fancyvrb}
```

Env `VerbatimClass` `{\class}` [*Verbatim options*]

Creates a `Verbatim` enclosed in a `<div>` of the given class.

```
4291 \NewDocumentEnvironment{VerbatimClass}{m 0{}}
4292 {%
4293 \renewcommand*{\LWR@Verbatimclass}{#1}%
4294 \LWR@origVerbatim[#2]%
4295 }
4296 {\endVerbatim}
```

After the preamble is loaded, after any patches to `Verbatim`:

```
4297 \AfterEndPreamble{
```

Remember the original definition of `Verbatim`:

```
4298 \let\LWR@origVerbatim\Verbatim
```

Env `Verbatim` Patched to place the environment in a `fancyvrb` div, and the label in a `fancyvrblabel` div. Also corrects the left margin for line numbers. Also uses `VerbatimHTMLWidth` to control placement of line numbers on the right. Aligning the right margin requires knowing the width.

```
4299 \renewcommand*{\Verbatim}{%
4300 \LWR@forcenewpage
4301 \renewcommand*{\LWR@Verbatimclass}{fancyvrb}%
```

```

4302 \LWR@origVerbatim%
4303 }

```

The following patches to `Verbatim` are executed at the start and end of the environment, depending on the choice of `frame`. Original code is from the `fancyvrb` package.

```

4304 \newcommand*{\LWR@fvstartnone}{%
4305 \LWR@traceinfo{fvstartnone}%
4306 \ifbool{LWR@verbtags}{\hbox to\z@{\LWR@htmltagc{div class="\LWR@Verbatimclass"}}}{}%
4307 \hbox to\z@{\LWR@atbeginverbatim{verbatim}}}%
4308 }
4309
4310 \newcommand*{\LWR@fvendnone}{%
4311 \LWR@traceinfo{fvendnone}%
4312 \hbox to\z@{\LWR@afterendverbatim}%
4313 \ifbool{LWR@verbtags}{\hbox to\z@{\LWR@htmltagc{/div}}}{}%
4314 }
4315
4316 \newcommand*{\LWR@fvstartsingle}{%
4317 \LWR@traceinfo{fvstartsingle}%
4318 \LWR@fvstartnone%
4319 \FV@BeginListFrame@Single%
4320 }
4321
4322 \newcommand*{\LWR@fvendsingle}{%
4323 \LWR@traceinfo{fvendsingle}%
4324 \FV@endListFrame@Single%
4325 \LWR@fvendnone%
4326 }
4327
4328 \newcommand*{\LWR@fvstartline}{%
4329 \LWR@traceinfo{fvstartline}%
4330 \LWR@fvstartnone%
4331 \FV@BeginListFrame@Lines%
4332 }
4333
4334 \newcommand*{\LWR@fvendline}{%
4335 \LWR@traceinfo{fvendline}%
4336 \FV@endListFrame@Lines%
4337 \LWR@fvendnone%
4338 }

```

The following patches select the start/left/right/end behaviors depending on `frame`. Original code is from the `fancyvrb` package.

```

4339 \def\FV@Frame@none{%
4340 \let\FV@BeginListFrame\LWR@fvstartnone%
4341 \let\FV@LeftListFrame\relax%

```

```

4342 \let\FV@RightListFrame\relax%
4343 \let\FV@EndListFrame\LWR@fvendnone}
4344
4345 \def\FV@Frame@single{%
4346 \let\FV@BeginListFrame\LWR@fvstartsingle%
4347 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
4348 \let\FV@RightListFrame\FV@RightListFrame@Single%
4349 \let\FV@EndListFrame\LWR@fvendsingle}
4350
4351 \def\FV@Frame@lines{%
4352 \let\FV@BeginListFrame\LWR@fvstartline%
4353 \let\FV@LeftListFrame\relax%
4354 \let\FV@RightListFrame\relax%
4355 \let\FV@EndListFrame\LWR@fvendline}
4356
4357 \def\FV@Frame@topline{%
4358 \let\FV@BeginListFrame\LWR@fvstartline%
4359 \let\FV@LeftListFrame\relax%
4360 \let\FV@RightListFrame\relax%
4361 \let\FV@EndListFrame\LWR@fvendnone}
4362
4363 \def\FV@Frame@bottomline{%
4364 \let\FV@BeginListFrame\LWR@fvstartnone%
4365 \let\FV@LeftListFrame\relax%
4366 \let\FV@RightListFrame\relax%
4367 \let\FV@EndListFrame\LWR@fvendline}
4368
4369 \def\FV@Frame@leftline{%
4370 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
4371 \ifx\FancyVerbFillColor\relax%
4372 \let\FV@FrameFillLine\relax%
4373 \else%
4374 \@tempdima\FV@FrameRule\relax%
4375 \multiply\@tempdima-\tw@%
4376 \edef\FV@FrameFillLine{%
4377 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}}%
4378 \kern-\number\@tempdima sp}}%
4379 \fi%
4380 \let\FV@BeginListFrame\LWR@fvstartnone%
4381 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
4382 \let\FV@RightListFrame\relax%
4383 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the fancyvrb package.

```

4384 \def\FV@SingleFrameLine#1{%
4385 \hbox to\z@{%
4386 % \kern\leftmargin

```

```

4387 \ifnum#1=\z@\relax
4388 \let\FV@Label\FV@LabelBegin
4389 \else
4390 \let\FV@Label\FV@LabelEnd
4391 \fi
4392 \ifx\FV@Label\relax
4393 % \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
4394 \else
4395 \ifnum#1=\z@
4396 % \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
4397 \ifx\FV@LabelPositionTopLine\relax
4398 \else
4399 \LWR@htmltagc{div class="fancyvrblabel"}
4400 \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
4401 \LWR@htmltagc{/div}
4402 \fi
4403 \else
4404 % \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
4405 \ifx\FV@LabelPositionBottomLine\relax
4406 \else
4407 \LWR@htmltagc{div class="fancyvrblabel"}
4408 \LWR@origtextrm{\FV@LabelEnd}
4409 \LWR@htmltagc{/div}
4410 \fi
4411 \fi
4412
4413 \fi
4414 \hss
4415 }
4416 }

```

Processes each line, adding optional line numbers. Original code is from the `fancyvrb` package.

```

4417 \def\FV@ListProcessLine#1{%
4418 \hbox to \hsize{%
4419 % \kern\leftmargin
4420 \hbox to \VerbatimHTMLWidth {%
4421 \ifcvoid\FV@LeftListNumber}{-}{\kern 2.5em}%
4422 \FV@LeftListNumber%
4423 % \FV@LeftListFrame
4424 \FancyVerbFormatLine{#1}%
4425 \hss%
4426 % \FV@RightListFrame
4427 \FV@RightListNumber%
4428 }%
4429 \hss% required to avoid underfull hboxes
4430 }
4431 }

```

Env **BVerbatim**

```

4432 \AtBeginEnvironment{BVerbatim}
4433 {
4434 \LWR@forcenewpage
4435 \LWR@atbeginverbatim{bverbatim}
4436
4437 }
4438
4439 \AfterEndEnvironment{BVerbatim}
4440 {
4441 \leavevmode\par\vspace{-\baselineskip}
4442 \LWR@afterendverbatim
4443 }
```

Env **LVerbatim** No changes required.

End of the modifications to make at the end of the preamble:

```

4444 } % \AfterEndPreamble
```

\UseVerbatim $\{\langle text \rangle\}$

No changes required.

```

4445 \end{warpHTML}
```

53 Theorems

\newtheorem $\{\langle text \rangle\}$ [$\langle counter \rangle$] -or- [$\langle oldname \rangle$] $\{\langle text \rangle\}$

A few minor changes are made to supply HTML tags.

- The entire theorem is placed into a div of class **theoremcontents**.
- The label for each theorem is placed inside a span of class **theoremlabel**.
- The contents are placed inside a div of class **theoremcontents**.

for HTML output: 4446 \begin{warpHTML}

\@begintheorem $\{\langle name \rangle\}$ $\{\langle number \rangle\}$

```

4447 \renewcommand{\@begintheorem}[2]{%
4448 \LWR@forcenewpage
4449 \BlockClass{theoremcontents}
4450 \InlineClass{theoremlabel}{#1\ #2\ }
4451 }

```

`\@opargbegintheorem` $\{\langle name \rangle\} \{\langle number \rangle\} \{\langle oparg \rangle\}$

```

4452 \renewcommand{\@opargbegintheorem}[3]{%
4453 \LWR@forcenewpage
4454 \BlockClass{theoremcontents}
4455 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
4456 }

```

`\@endtheorem`

```

4457 \renewcommand*\@endtheorem{%
4458 \endBlockClass% theoremcontents
4459 }


```

```

4460 \end{warpHTML}

```

54 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.

54.1 Itemize

for HTML output: 4461 `\begin{warpHTML}`

```

4462 \let\LWR@origitem\item

```

`\LWR@itemizeitem` $[\langle label \rangle]$

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.

```

4463 \newcommand*{\LWR@itemizeitem}{%
4464 \LWR@stoppars%
4465 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloseitem{}}%
4466 \LWR@htmltag{li}%
4467 \LWR@startpars%
4468 \LWR@origitem%
4469 }
```

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)*]

```

4470 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
4471
4472 \newcommand*{\LWR@itemizestart}{%
4473 \LWR@stoppars%
4474 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4475 \LWR@htmltag{ul style="list-style-type:none"}}%
4476 \LWR@startpars%
4477 \let\item\LWR@itemizeitem%
4478 }
4479
4480 \AtEndEnvironment{itemize}{\LWR@itemizeend}
4481
4482 \newcommand*{\LWR@itemizeend}{%
4483 \LWR@stoppars%
4484 \LWR@closeprevious{\LWR@depthlistitem}%
4485 \LWR@closeoneprevious{}%
4486 \LWR@startpars%
4487 }
```

54.2 Enumerate

An HTML unordered list is used with customized L^AT_EX-generated labels.

Env **enumerate** [*enumitem options*]

```

4488 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
4489
4490 \newcommand*{\LWR@enumeratestart}{%
4491 \LWR@stoppars%
4492 \LWR@pushhonestclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4493 \LWR@htmltag{ul style="list-style-type:none">{}}%
4494 \LWR@startpars%
4495 \let\item\LWR@itemizeitem%
4496 }
4497
4498
4499 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
4500
4501 \newcommand*{\LWR@enumerateend}{%
4502 \LWR@stoppars%
4503 \LWR@closeprevious{\LWR@depthlistitem}%
4504 \LWR@closeoneprevious{}}%
4505 \LWR@startpars%
4506 }

```

54.3 Description

\LWR@descitem [*label*] Handles an `\item` inside a description.

```

4507 \newcommand*{\LWR@descitem}[1][]%
4508 {%
4509 \LWR@stoppars%
4510 \LWR@setlatestname{#1}%
4511 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem{}}%
4512 \LWR@origitem[]%

```

Be sure the label doesn't print to the left of the rest of the file:

```

4513 \LWR@origspace{1in}
4514 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
4515 \LWR@orignewline%
4516 \LWR@htmltag{dd}%
4517 \LWR@startpars%
4518 }

```

Env **description** [*enumitem options*]

```

4519 \AtBeginEnvironment{description}{\LWR@descriptionstart}
4520

```

```

4521 \newcommand*{\LWR@descriptionstart}{%
4522 \LWR@stoppars%
4523 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}%
4524 \LWR@htmltag{dl}%
4525 \LWR@startpars%
4526 \let\item\LWR@descitem%
4527 }
4528
4529 \AtEndEnvironment{description}{\LWR@descriptionend}
4530
4531 \newcommand*{\LWR@descriptionend}{%
4532 \LWR@stoppars%
4533 \LWR@closeprevious{\LWR@depthlistitem}%
4534 \LWR@closeoneprevious{}}%
4535 \LWR@startpars%
4536 }

```

`\newlist` $\{\langle name \rangle\} \{\langle type \rangle\} \{\langle maxdepth \rangle\}$

`\renewlist` $\{\langle name \rangle\} \{\langle type \rangle\} \{\langle maxdepth \rangle\}$

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.

```

4537 \let\LWR@orignewlist\newlist
4538
4539 \renewcommand*{\newlist}[3]{%
4540 \LWR@orignewlist{#1}{#2}{#3}%
4541 \AtBeginEnvironment{#1}{\csuse{\LWR@#2start}}%
4542 \AtEndEnvironment{#1}{\csuse{\LWR@#2end}}%
4543 }
4544 \end{warpHTML}

```

55 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.
- \newcolumnntype is ignored; unknown column types are set to 1.
- tabularx ignores the width, but X columns do produce paragraph columns or multicolumns.

- 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 lwarp’s tabular handling algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:

```
\warpprintonly{1 & 2 & {\multirow{2}{2cm}{Text}} & 3 \\\n\warpHTMLonly{1 & 2 & \multirow{2}{2cm}{Text} & 3 \\\n}
```

⚠ \multirow &
\multicolumn

⚠ \multirow

\multirow with rules

rule at last row

⚠ paragraphs

\cmidrule width, trim

longtable headings

⚠ \warpprintonly

⚠ S columns

55.1 Token lookahead

Used by `\LWR@futurenonSPACElet` to look at the next token.

for HTML output: 4545 `\begin{warphTML}`

`\LWR@mynexttoken`

4546 `\newcommand\LWR@mynexttoken\relax`

`\futurelet` copies the next token then executes a function to analyze

`\LWR@futurenonSPACElet` does the same, but ignores intervening white space

Based on the booktabs style:

`\LWR@futurenonSPACElet`

```
4547 \def\LWR@futurenonSPACElet#1{\def\LWR@cs{#1}%
4548 \afterassignment\LWR@fns lone\let\nexttoken= }
4549 \def\LWR@fns lone{\expandafter\futurelet\LWR@cs\LWR@fns ltwo}
4550 \def\LWR@fns ltwo{%
4551 \expandafter\ifx\LWR@cs\@sptoken\let\next=\@BTfns lthree%
4552 \else\let\next=\nexttoken\fi\next}
4553 \def\@BTfns lthree{\afterassignment\LWR@fns lone\let\next= }
```

`\LWR@getmynexttoken` Looks ahead and copies the next token into `\LWR@mynexttoken`.

```
4554 \newcommand*{\LWR@getmynexttoken}{%
4555 % nothing must follow this next line
4556 \LWR@futurenonSPACElet\LWR@mynexttoken\LWR@tabledatacolumn tag
4557 }
```

55.2 Booleans

Bool `LWR@startedrow` True if should print a row tag before this column.

```
4558 \newbool{LWR@startedrow}
4559 \boolfalse{LWR@startedrow}
```

Bool `LWR@doinghline` True if the next row will have an hline above it.

```
4560 \newbool{LWR@doinghline}
4561 \boolfalse{LWR@doinghline}
```

Bool LWR@doingtbrule True if the next row will have a top/bottom rule above it.

```
4562 \newbool{LWR@doingtbrule}
4563 \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.

```
4564 \newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty multi-row cell, and thus there is no data tag to close.

```
4565 \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.

```
4566 \newbool{LWR@intabularmetadata}
4567 \boolfalse{LWR@intabularmetadata}
```

55.3 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.

```
4568 \newcommand*{\LWR@closetabledatacell}{%
4569 \global\booltrue{LWR@intabularmetadata}%
4570 \ifbool{LWR@exitingtabular}{}%
4571 {% not exiting tabular
4572 \ifbool{LWR@skippingmrowcell}{}%
4573 {% not skippingmrowcell
```

Insert any < then any @ and ! column contents:

```
4574 \unskip%
4575 \LWR@getexparray{LWR@colafterspec}{\theLWR@tablecolspos}%
4576 \LWR@getexparray{LWR@colatspec}{\theLWR@tablecolspos}%
4577 \LWR@getexparray{LWR@colbangspec}{\theLWR@tablecolspos}%
```

Close paragraphs:

```
4578 \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
4579 \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell:

```
4580 \unskip\LWR@htmltag{/td}\LWR@orignewline%
4581 }% not skipping mrowcell
4582 }% not exiting tabular
4583 \global\boolfalse\LWR@skippingmrowcell}%
4584 }
```

`LWR@tabulardepth` tracks whether `&` is being used inside a `tabular`.

```
4585 \newcounter{LWR@tabulardepth}
4586 \setcounter{LWR@tabulardepth}{0}
4587
```

When not used inside a `tabular`, `&` performs its original function as recorded here (with catcode 4).

```
4588 \def\LWR@origampmacro{&}
```

See below for why the group is used.

```
4589 \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.

```
4590 \catcode'\&=\active
4591
4592 \gdef&{%
4593 \ifthenelse{\cnttest{\value{LWR@tabulardepth}}{>}{0}}%
4594 {%
```

If not skipping a multirow cell, close the current data cell.

```
4595 \unskip%
4596 \LWR@closetabledatacell%
```

Move to the next column.

```
4597 \addtocounter{LWR@tablecolspos}{1}%
```

Look at the next token to decide multi or single column data tag.

```
4598 \LWR@getmynexttoken%
4599 }%
```

If not inside a tabular, performs the original action:

```
4600 {\LWR@origampmacro}%
4601 }
4602 \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@lwarpStart finally makes & active at the beginning of the HTML conversion.

55.4 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendoffline

Throws away options \\[dim] or *

```
\LWR@tabularendoffline
```

```
4603 \NewDocumentCommand{\LWR@tabularendoffline}{s o}
4604 {%
4605 \LWR@closetabledatacell%
```

Finish the previous row:

```
4606 \LWR@htmltag{/tr}\LWR@orignewline
4607 \global\booltrue{\LWR@intabularmetadata}
```

Not yet started a table row:

```
4608 \global\boolfalse{\LWR@startedrow}
```

Additional setup:

```
4609 \global\boolfalse{\LWR@doinghline}%
4610 \global\boolfalse{\LWR@doingtbrule}%
4611 \LWR@clearmidrules%
```

Start at first column:

```
4612 \setcounter{\LWR@tablecolspos}{1}
```


Look at the next token to decide between single column data tag or a special case:

```
4613 \LWR@getmynexttoken%
4614 }
```

55.5 Variables

```
4615 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
4616 \newcommand*{\LWR@pposition}{}
4617 \newcommand*{\LWR@pleft}{}
4618 \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.

```
4619 \newcommand*{\LWR@tablecolspec}{}

```

`\LWR@strresult` Holds the result of `Str` functions.

```
4620 \newcommand*{\LWR@strresult}{}
4621 \newcommand*{\LWR@strresulttwo}{}

```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

```
4622 \newcommand*{\LWR@origcolspec}{}

```

`Ctr` `LWR@tablecolwidth` Holds the width of the table specification.

(This is not the total # columns.)

```
4623 \newcounter{LWR@tablecolwidth}

```

`Ctr` `LWR@tablecolspos` Where are currently looking into the table column specification.

```
4624 \newcounter{LWR@tablecolspos}

```

`Ctr` `LWR@tabletotalcols` Holds the final number of table columns.

```
4625 \newcounter{LWR@tabletotalcols}

```

`Ctr` `LWR@tabletotalcolsnext` Holds the next column while parsing. Is one more than `LWR@tabletotalcols`.

```
4626 \newcounter{LWR@tabletotalcolsnext}

```

- `LWR@colatspec` A data array of specifications for @ columns. The leftmost's index is `leftedge`, the others are counter values. See section 28.
- `LWR@colbangspec` A data array of specifications for ! columns. The leftmost's index is `leftedge`, the others are counter values. See section 28.
- `LWR@colbeforespec` A data array of specifications for > columns.
- `LWR@colafterspec` A data array of specifications for < columns.

55.6 Parsing @, >, <, ! columns

`\LWR@parseatcolumn` Handles @{text} columns.

```
4627 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
4628 \LWR@traceinfo{at column}%
4629 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into `\LWR@strresult`, expanding once:

```
4630 \LWR@traceinfo{about to read the next token:}%
4631 \expandarg%
4632 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]
4633 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
4634 \LWR@traceinfo{have now read the next token}%
4635 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}{
4636 {% left edge of the table:
4637 \LWR@traceinfo{at the left edge}%
4638 \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@strresult}%
4639 }%
4640 {% not at the left edge:
4641 \LWR@traceinfo{not at the left edge}%
4642 \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
4643 \LWR@traceinfo{at \theLWR@tabletotalcols: %
4644 \LWR@getexparray{LWR@colatspec}{\theLWR@tabletotalcols}}!}%
4645 \let\LWR@strresult\relax%
4646 \booltrue{LWR@validtablecol}%
4647 }%
4648 }
```

\LWR@parsebangcolumn

```
4649 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
4650 \LWR@traceinfo{bang column}%
```

```
4651 \addtocounter{\LWR@tablecolspos}{1}%
```

Read the next token into \LWR@strresult, expanding once:

```
4652 \LWR@traceinfo{about to read the next token:}%
```

```
4653 \expandarg%
```

```
4654 \StrChar{\LWR@origcolspec}{\the\LWR@tablecolspos}[\LWR@strresult]
```

```
4655 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```
4656 \LWR@traceinfo{have now read the next token}%
```

```
4657 \ifthenelse{\cnttest{\value{\LWR@tabletotalcols}}=0}
```

```
4658 {% left edge of the table:
```

```
4659 \LWR@traceinfo{at the left edge}%
```

```
4660 \LWR@setexparray{\LWR@colbangspec}{leftedge}{\LWR@strresult}%
```

```
4661 }%
```

```
4662 {% not at the left edge:
```

```
4663 \LWR@traceinfo{not at the left edge}%
```

```
4664 \LWR@setexparray{\LWR@colbangspec}{\the\LWR@tabletotalcols}{\LWR@strresult}%
```

```
4665 \LWR@traceinfo{bang \the\LWR@tabletotalcols: \LWR@colbangspec(\the\LWR@tabletotalcols)!}%
```

```
4666 }%
```

```
4667 \let\LWR@strresult\relax%
```

```
4668 \booltrue{\LWR@validtablecol}%
```

```
4669 }
```

\LWR@parsebeforecolumn Handles >{text} columns.

```
4670 \newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
4671 \addtocounter{\LWR@tablecolspos}{1}%
```

Read the next token, expanding once into \LWR@strresult:

```
4672 \expandarg%
```

```
4673 \StrChar{\LWR@origcolspec}{\the\LWR@tablecolspos}[\LWR@strresult]%
```

```
4674 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```

4675 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\LWR@strresult}%
4676 \let\LWR@strresult\relax%
4677 \booltrue{LWR@validtablecol}%
4678 }

```

`\LWR@parseaftercolumn` Handles <{text} columns.

```

4679 \newcommand*{\LWR@parseaftercolumn}{%

```

Move to the next token after the '<':

```

4680 \addtocounter{LWR@tablecolspos}{1}%

```

Read the next token, expanding once into `\LWR@strresult`:

```

4681 % \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]
4682 \expandarg%
4683 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
4684 \fullexpandarg%

```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```

4685 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
4686 \let\LWR@strresult\relax%
4687 \booltrue{LWR@validtablecol}%
4688 }

```

`\LWR@parseskipcolumn` Handles columns to skip, such as the vertical bar.

```

4689 \newcommand*{\LWR@parseskipcolumn}{%
4690 \booltrue{LWR@validtablecol}%
4691 }

```

55.7 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.

```

4692 \newcommand*{\LWR@parsenormalcolumn}[1]{%
4693 \appto\LWR@tablecolspec{#1}%
4694 \addtocounter{LWR@tabletotalcols}{1}%
4695 \addtocounter{LWR@tabletotalcolsnext}{1}%
4696 \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcolsnext}{\relax}%

```

```

4697 \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcolsnext}{\relax}%
4698 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\relax}%
4699 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcolsnext}{\relax}%
4700 \booltrue{LWR@validtablecol}%
4701 }

```

55.8 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn` $\{ \langle thiscolumn \rangle \}$ The width will be ignored.

```

4702 \newcommand*{\LWR@parsepcolumn}[1]{%

```

Converts to the given column type.

```

4703 \LWR@parsenormalcolumn{#1}%

```

skips the following width

```

4704 \addtocounter{LWR@tablecolspos}{1}%
4705 }

```

55.9 Parsing ‘D’ columns

From the dcolumn package.

`\LWR@parseDcolumn` $\{ \langle thiscolumn \rangle \}$ The three parameters will be ignored.

```

4706 \newcommand*{\LWR@parseDcolumn}[1]{%

```

Converts to the given column type.

```

4707 \LWR@parsenormalcolumn{#1}%

```

Skips the following three parameters.

```

4708 \addtocounter{LWR@tablecolspos}{3}%
4709 }

```

55.10 Parsing the column specifications



HTML CSS cannot exactly match the \LaTeX concept of a baseline for a table row. Table 7 shows the \LaTeX results for various vertical-alignment choices, with the

Table 7: Tabular baseline

l	p	m	b	r
			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

baseline of the first column drawn across all the columns for comparison. See the **p** column specification in table 8 for details.

Table 8 describes how each kind of column is converted to HTML.

Bool **LWR@validtablecol** True if found a valid table column type.

```
4710 \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**.

```
4711 \newcommand*{\LWR@parsetablecols}[1]{%
4712 \LWR@traceinfo{LWR@parsetablecols started}%
```

Remember the original supplied column spec:

```
4713 \renewcommand*{\LWR@origcolspec}{#1}%
```

Clear the parsed resulting column spec:

```
4714 \renewcommand*{\LWR@tablecolspec}{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
4715 \setcounter{LWR@tabletotalcols}{0}%
4716 \setcounter{LWR@tabletotalcolsnext}{1}%
4717 \LWR@setexparray{LWR@colatspec}{leftedge}{\relax}%
4718 \LWR@setexparray{LWR@colatspec}{1}{\relax}%
4719 \LWR@setexparray{LWR@colatspec}{2}{\relax}%
4720 \LWR@setexparray{LWR@colatspec}{3}{\relax}%
```

Table 8: Tabular HTML column conversions

l, r, c:	Converted to table cells without paragraph tags. Uses CSS <code>vertical-align:middle</code> so that top or bottom-aligned cells may go above or below this cell.
p:	Converted to table cells with paragraph tags. Ref: Table 7, \LaTeX places the top line of a parbox aligned with the rest of the text line, so CSS <code>vertical-align:bottom</code> is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the CSS may be desired, changing <code>td.tdp</code> , <code>td.tdP</code> , <code>td.tdprule</code> , and <code>td.tdPrule</code> to <code>vertical-align: middle</code> . Another possibility is to change L,R,C, and P to <code>vertical-align: top</code> and not worry about the alignment of B and M cells or trying to approximate \LaTeX baselines.
m:	With paragraph tags, CSS <code>vertical-align:middle</code> .
b:	With paragraph tags, CSS <code>vertical-align:top</code> so that the bottom of the text is closest to the middle of the text line.
P, M, B:	Horizontally-centered versions.
S:	Converted to 'r'. From the siunitx package.
D:	Converted to 'c'. From the dcolumn package.
@, !, >, <:	One each, in that order.
Unknown:	Converted to 'l'.
\newcolumn:	Currently treated as unknown.

```

4721 \LWR@setexparray{LWR@colbangspec}{\leftedge}{\relax}%
4722 \LWR@setexparray{LWR@colbangspec}{1}{\relax}%
4723 \LWR@setexparray{LWR@colbangspec}{2}{\relax}%
4724 \LWR@setexparray{LWR@colbangspec}{3}{\relax}%
4725 \LWR@setexparray{LWR@colbeforespec}{1}{\relax}%
4726 \LWR@setexparray{LWR@colbeforespec}{2}{\relax}%
4727 \LWR@setexparray{LWR@colbeforespec}{3}{\relax}%
4728 \LWR@setexparray{LWR@colafterspec}{1}{\relax}%
4729 \LWR@setexparray{LWR@colafterspec}{2}{\relax}%
4730 \LWR@setexparray{LWR@colafterspec}{3}{\relax}%

```

Starting at the first column specification:

```

4731 \setcounter{LWR@tablecolspos}{1}%

```

Place the colspecs string length into `\LWR@strresult`, and remember the number of characters in the column specification:

```

4732 \LWR@traceinfo{about to StrLen}%
4733 \noexpandarg%
4734 \StrLen{#1}[\LWR@strresult]%
4735 \fullexpandarg%
4736 \LWR@traceinfo{finished StrLen}%
4737 \setcounter{LWR@tablecolwidth}{\LWR@strresult}%

```

Scan through the column specifications:

```

4738 \whileof{\not\value{LWR@tablecolspos}>\value{LWR@tablecolwidth}}{%

```

Place the next single-character column type into `\LWR@strresult`:

```

4739 \noexpandarg%
4740 \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]%
4741 \fullexpandarg%

```

Not yet found a valid column type

```

4742 \boolfalse{LWR@validtablecol}%

```

Note that the parameter for a `p{spec}` column is a token list which will NOT match `l,c,r,p`.



```

4743 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{%
4744 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{%
4745 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{%
4746 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{%
4747 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{%
4748 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{%
4749 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{%

```



```

4750 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{}%
4751 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
4752 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
4753 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
4754 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
4755 \IfStrEq{\LWR@strresult}{|}{\LWR@parseskipcolumn}{}%
4756 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
4757 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
4758 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%

```

From the dcolumn package:

```

4759 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{}%

```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```

4760 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%

```

Many people define centered versions “P”, “M”, and “B”:

```

\newcolumntype{P}[1]{>\centering\arraybackslashp{#1}}

```

```

4761 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
4762 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
4763 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%

```

If this column was an invalid column type, convert it to a p column:

```

4764 \ifbool{LWR@validtablecol}{}{}%
4765 \LWR@parsenormalcolumn{l}%
4766 }%
4767 \addtocounter{LWR@tablecolspos}{1}%
4768 }%
4769 }%

```

55.11 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.

```

4770 \newcommand*{\LWR@maybenewtablerow}
4771 {%
4772 \ifbool{LWR@startedrow}%
4773 {}% started the row
4774 {}% not started the row

```

Remember that now have started the row:

```
4775 \global\booltrue{LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```
4776 \global\booltrue{LWR@intabularmetadata}%
4777 \ifbool{LWR@doinghline}%
4778 {\LWR@htmltag{tr class="hline"{}}\LWR@orignewline}%
4779 {% not doing hline
4780 \ifbool{LWR@doingtbrule}%
4781 {\LWR@htmltag{tr class="tbrule"{}}\LWR@orignewline}%
4782 {\LWR@htmltag{tr}\LWR@orignewline}%
4783 }% end of not doing hline
4784 }% end of not started the row
4785 }
```

55.12 Data opening tag

`\LWR@tabledatasinglecolumnstag` Print a table data opening tag with style for alignment

```
4786 \newcommand*{\LWR@tabledatasinglecolumnstag}%
4787 {%
4788 \LWR@maybenewtablerow%
```

If have found the end of tabular command, do not create the next data cell:

```
4789 \ifbool{LWR@exitingtabular}{}%
4790 {% not exiting tabular
```

Fetch the current column's alignment character into `\LWR@strresult`:

```
4791 \StrChar{\LWR@tablecolspec}{\the\LWR@tablecolspos}[\LWR@strresult]%
```

print the start of a new table data cell:

```
4792 \LWR@htmltag{td class="td%
```

append this column's spec:

```
4793 \LWR@strresult%
```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag:

```
4794 \ifthenelse{\equal{\LWR@getexparray{LWR@midrules}{\the\LWR@tablecolspos}}{Y}}{rule}{}%
4795 "{}}%
```

If this is a p, m, b, or X column, allow paragraphs:

```

4796 \ifthenelse{%
4797 \equal{\LWR@strresult}{p}\OR%
4798 \equal{\LWR@strresult}{m}\OR%
4799 \equal{\LWR@strresult}{b}\OR%
4800 \equal{\LWR@strresult}{P}\OR%
4801 \equal{\LWR@strresult}{M}\OR%
4802 \equal{\LWR@strresult}{B}\OR%
4803 \equal{\LWR@strresult}{X}%
4804 }%
4805 {% allow pars
4806 \LWR@startpars%
4807 \global\booltrue{\LWR@tableparcell}%
4808 }% allow pars
4809 {}% no pars

```

Print the @ and ! contents before first column, and then the > contents:

```

4810 \ifthenelse{\cnttest{\value{\LWR@tablecolspos}}=1}%
4811 {%
4812 \LWR@getexparray{\LWR@colatspec}{leftedge}%
4813 \LWR@getexparray{\LWR@colbangspec}{leftedge}%
4814 }% left edge
4815 {}% not left edge
4816 \LWR@getexparray{\LWR@colbeforespec}{\the\LWR@tablecolspos}%
4817 \global\boolfalse{\LWR@intabularmetadata}%
4818 }% not exiting tabular
4819 }%

```

55.13 Midrules

LWR@midrules LWR@midrules is a data array (section 28) of columns containing Y if a midrule should be created for each column.

Ctr LWR@midrulecounter Indexes across the LWR@midrules data array.

```

4820 \newcounter{\LWR@midrulecounter}

```

\LWR@clearmidrules Start new midrules. Called at beginning of tabular and also at \.

Clears all LWR@midrules markers for this line.

```

4821 \newcommand*{\LWR@clearmidrules}
4822 {%
4823 \setcounter{\LWR@midrulecounter}{1}%

```

```

4824 \whiledo{%
4825 \cnttest{\value{LWR@midrulecounter}}{<=}{\value{LWR@tablecolwidth}}%
4826 }%
4827 {%
4828 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{\relax}%
4829 \addtocounter{LWR@midrulecounter}{1}%
4830 }%
4831 }

```

`\LWR@subcmidrule` [*width*] [*trim*] [*leftcolumn*] [*rightcolumn*]

Marks `LWR@midrules` data array elements to be “Y” from left to right columns.

```

4832 \newcommand*{\LWR@subcmidrule}[4]{%
4833 \setcounter{LWR@midrulecounter}{#3}%
4834 \whiledo{\cnttest{\value{LWR@midrulecounter}}{<=}{#4}}%
4835 {%
4836 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{Y}%
4837 \addtocounter{LWR@midrulecounter}{1}%
4838 }% end of the whiledo
4839 }

```

`\LWR@docmidrule` [*width*] [*trim*] [*leftcolumn-rightcolumn*]

Marks `LWR@midrules` array elements to be “Y” from left to right columns.

```

4840 \NewDocumentCommand{\LWR@docmidrule}{o d() >{\SplitArgument{1}{-}}m}%
4841 {\LWR@subcmidrule{#1}{#2}{#3}}

```

55.14 Multicolumns

55.14.1 Parsing multicolumns

```

4842 \newcounter{LWR@tablemulticolwidth}
4843 \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`.

```

4844 \newcommand*{\LWR@printmccoltype}[1]{%
4845 \LWR@traceinfo{lw@printmccoltype -#1-}%

```

Get one token of the column spec:

```

4846 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%

```

Add to the HTML tag depending on which column type is found:

```

4847 \IfStrEq{\LWR@strresult}{l}{l}{}%
4848 \IfStrEq{\LWR@strresult}{c}{c}{}%
4849 \IfStrEq{\LWR@strresult}{r}{r}{}%
4850 \IfStrEq{\LWR@strresult}{p}{p}{}%
4851 \IfStrEq{\LWR@strresult}{m}{m}{}%
4852 \IfStrEq{\LWR@strresult}{b}{b}{}%
4853 \IfStrEq{\LWR@strresult}{P}{P}{}%
4854 \IfStrEq{\LWR@strresult}{M}{M}{}%
4855 \IfStrEq{\LWR@strresult}{B}{B}{}%
4856 \IfStrEq{\LWR@strresult}{S}{r}{}%
4857 \IfStrEq{\LWR@strresult}{X}{p}{}%
4858 \LWR@traceinfo{lwarp@printmccoltype done}%
4859 }

```

`\LWR@multicolpartext` Print the data with paragraph tags:

```

4860 \newcommand*{\LWR@multicolpartext}{%
4861 \LWR@startpars%
4862 \LWR@multicoltext%
4863 \LWR@stoppars%
4864 }

```

`\LWR@multicolother` `{\colspec}` For @, >, <, print the next token without paragraph tags:

```

4865 \newcommand*{\LWR@multicolother}[1]{%
4866 \addtocounter{LWR@tablemulticolspos}{1}%
4867 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
4868 \LWR@strresult%

```

A valid column data type was found:

```

4869 \booltrue{LWR@validtablecol}%
4870 }

```

`\LWR@multicolskip` Nothing to print for this column type.

```

4871 \newcommand*{\LWR@multicolskip}{%

```

A valid column data type was found:

```

4872 \booltrue{LWR@validtablecol}%
4873 }

```

`\LWR@printmccoldata` `{\colspec}` Print the data for any valid column type found.

```
4874 \newcommand*{\LWR@printmccoldata}[1]{%
4875 \LWR@traceinfo{lw@printmccoldata -#1}%
```

Not yet found a valid column type:

```
4876 \boolfalse{\LWR@validtablecol}%
```

Get one token of the column spec:

```
4877 \StrChar{#1}{\the\LWR@tablemulticolspos}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
4878 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
4879 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
4880 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
4881 \IfStrEq{\LWR@strresult}{D}{}%
4882 \addtocounter{\LWR@tablemulticolspos}{3}% skip parameters
4883 \LWR@multicoltext%
4884 }{}%
4885 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolparttext}{}%
4886 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolparttext}{}%
4887 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolparttext}{}%
4888 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolparttext}{}%
4889 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolparttext}{}%
4890 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolparttext}{}%
4891 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolparttext}{}%
4892 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolparttext}{}%
4893 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
4894 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@multicolother{#1}}{%
4895 \IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}{%
4896 \IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}{%
4897 \IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}{%
```

If an invalid column type:

```
4898 \ifbool{\LWR@validtablecol}{\LWR@multicoltext}%
```

Tracing:

```
4899 \LWR@traceinfo{lw@printmccoldata done}%
4900 }
```

`\parsemulticolumnalignment` $\{\langle 1: \text{colspec} \rangle\} \{\langle 2: \text{printresults} \rangle\}$

Scan the multicolumn specification and execute the printfunction for each entry.

Note that the spec for a `p{spec}` column, or `@`, `>`, `<`, is a token list which will NOT match `l`, `c`, `r`, or `p`.

```
4901 \newcommand*\LWR@parsemulticolumnalignment}[2]{%
4902 \setcounter{LWR@tablemulticolspos}{1}%
4903 \StrLen{#1}[\LWR@strresult]%
4904 \setcounter{LWR@tablemulticolwidth}{\LWR@strresult}%
```

Scan across the tokens in the column spec:

```
4905 \whiledo{%
4906 \not\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolwidth}%
4907 }%
4908 {%
```

Execute the assigned print function for each token in the column spec:

```
4909 #2{#1}%
```

Move to the next token in the column spec:

```
4910 \addtocounter{LWR@tablemulticolspos}{1}%
4911 }%
4912 }
```

55.14.2 High-level multicolumn interface

`\LWR@domulticolumn` $\{ \langle 1: numcols \rangle \} \{ \langle 2: colspec \rangle \} \{ \langle 3: text \rangle \}$

```
4913 \newcommand{\LWR@multicoltext}{}
4914
4915 \NewDocumentCommand{\LWR@domulticolumn}{m m +m}{%
4916 \LWR@traceinfo{lw@domulticolumn -#1- -#2-}%
```

Remember the text to be inserted, and remember that a valid column type was found:

```
4917 \renewcommand{\LWR@multicoltext}{%
4918 #3%
4919 \booltrue{LWR@validtablecol}%
4920 }%
```

Row processing:

```
4921 \LWR@maybe newtablerow%
```

Begin the opening table data tag:

```
4922 \LWR@htmltag{td colspan="#1"
4923 class="td%
```

Print the column type:

```
4924 \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”.

```
4925 \ifthenelse{\equal{\LWR@gettexpparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
```

Close the class tag’s opening quote:

```
4926 "%
4927 }% end of the opening table data tag
4928 \global\boolfalse{\LWR@intabularmetadata}%
4929 \LWR@parsemulticolumnalignment{#2}{\LWR@printmccoldata}%
4930 }
```

55.14.3 Longtable captions

Bool `LWR@starredlongtable` Per the caption package, step the counter if `longtable*`.

```
4931 \newbool{\LWR@starredlongtable}
4932 \boolfalse{\LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
4933 \providecommand*\LTcapttype{table}
```

`\LWR@longtabledatacaptiontag` * [*toc entry*] {*caption*}

```
4934 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
4935 {%
```

Remember the latest name for `\nameref`:

```
4936 \IfValueTF{#2}{% optional given?
4937 \ifthenelse{\equal{#2}{}}{% optional empty?
4938 {\LWR@setlatestname{#3}}{% empty
4939 {\LWR@setlatestname{#2}}{% given and non-empty
4940 }% optional given
4941 {\LWR@setlatestname{#3}}{% no optional
```


create a multicolumn across all the columns

```

4942 \LWR@domulticolumn{\theLWR@tabletotalcols}{P}{% \LWR@domulticolumn
4943 % \IfBooleanTF{#1}% star?
4944 % {\IfValueTF{#2}{\LWR@origcaption*{#2}{#3}}{\LWR@origcaption*{#3}}}
4945 % {\IfValueTF{#2}{\LWR@origcaption{#2}{#3}}{\LWR@origcaption{#3}}}
4946 \IfBooleanTF{#1}% star?

```

Star version, show a caption but do not make a LOT entry:

```

4947 {% yes star
4948 \LWR@htmlblocktag{figcaption}%
4949 #3%
4950 \LWR@htmlblocktag{/figcaption}%
4951 }%
4952 {% No star:

```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```

4953 \ifbool{LWR@starredlongtable}%
4954 {%
4955 \ifthenelse{\equal{#2}{}}% TOC entry
4956 {}%
4957 {%
4958 \refstepcounter{\LTcapttype}%
4959 \protected@edef\@currentlabel{%
4960 \csuse{p@\LTcapttype}\csuse{the\LTcapttype}}%
4961 }%
4962 }{}%

```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```

4963 \LWR@htmlblocktag{figcaption}%
4964 \csuse{fnum@\LTcapttype}\CaptionSeparator#3%
4965 \LWR@htmlblocktag{/figcaption}%

```

See if an optional caption was given:

```

4966 \ifthenelse{\equal{#2}{}}% TOC entry empty

```

if the optional caption was given, but empty, do not form a TOC entry

```

4967 {}%

```

If the optional caption was given, but might only be []:

```

4968 {% TOC entry not empty

```

4969 \IfNoValueTF{#2}% No TOC entry?

The optional caption is []:

```

4970 {% No TOC entry
4971 \addcontentsline%
4972 {\csuse{ext@LTcapttype}}%
4973 {\LTcapttype}%
4974 {%
4975 \protect\numberline%
4976 {\csuse{p@LTcapttype}\csuse{theLTcapttype}}%
4977 {\ignorespaces #3\protect\relax}%
4978 }%
4979 }% end of No TOC entry

```

The optional caption has text enclosed:

```

4980 {% yes TOC entry
4981 \addcontentsline%
4982 {\csuse{ext@LTcapttype}}%
4983 {\LTcapttype}%
4984 {%
4985 \protect\numberline%
4986 {\csuse{p@LTcapttype}\csuse{theLTcapttype}}%
4987 {\ignorespaces #2\protect\relax}%
4988 }%
4989 }% end of yes TOC entry
4990 }% end of TOC entry not empty
4991 }% end of no star
4992 }% end of \LWR@domulticolumn
4993
4994 \addtocounter{LWR@tablecolspos}{\theLWR@tabletotalcols}
4995 \addtocounter{LWR@tablecolspos}{-1}
4996
4997 }

```

55.14.4 \tabledatamulticolumntag

`\LWR@tabledatamulticolumntag` $\{ \langle numcols \rangle \} \{ \langle alignment \rangle \} \{ \langle text \rangle \}$

```

4998 \NewDocumentCommand{\LWR@tabledatamulticolumntag}{m m +m}%
4999 {%
5000 \LWR@domulticolumn{#1}{#2}{#3}%
5001 \addtocounter{LWR@tablecolspos}{#1}%
5002 \addtocounter{LWR@tablecolspos}{-1}%
5003 }

```

55.15 Multirow

Pkg multirow

```
\LWR@tabledatamultirowtag {\numrows} [(\bigstruts)] {\width} [(\fixup)] {\text}
```

```
5004 \NewDocumentCommand{\LWR@tabledatamultirowtag}{m o m o m}%
```

```
5005 {%
```

```
5006 \LWR@maybenewtablerow%
```

Print the start of a new table data cell:

```
5007 \LWR@htmltag{td rowspan="#1" class="td%
```

Append this column's spec:

```
5008 \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag:

```
5009 \ifthenelse{\equal{\LWR@gettexpparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
```

```
5010 "{}}
```

While printing the text, redefine \\\ to generate a new line

```
5011 \begingroup \let\\\LWR@endoffline #5 \endgroup
```

```
5012 \LWR@stoppars%
```

```
5013 \global\boolfalse{\LWR@intabularmetadata}%
```

```
5014 }%
```

55.16 Utility macros inside a table

```
5015 \newcommand*{\LWR@donothing}{}%
```

```
5016 \newcommand*{\LWR@domidrule}{\booltrue{\LWR@doinghline}}%
```

```
5017 \newcommand*{\LWR@dotbrule}{\booltrue{\LWR@doingtbrule}}%
```

55.17 Checking for a new table cell

`\LWR@tabledatacolumnntag` Open a new HTML table cell unless the next token is for a macro which does not create data, such as `\hline`, `\toprule`, etc:

```
5018 \newbool{\LWR@exitingtabular}
```

```
5019 \newcommand*{\LWR@tabledatacolumnntag}%
```

```
5020 {%
```

`\show\LWR@mynexttoken` to see what tokens to look for

If not any of the below, start a new table cell:

```
5021 \let\mynext\LWR@tabledatasinglecolumnntag%
```

If exiting the tabular:

```
5022 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\end}}{%
5023 {\booltrue\LWR@exitingtabular}}{ }%
```

`longtable` can have a caption in a cell

```
5024 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\caption}}{%
5025 {\let\mynext\LWR@donothing}}{ }%
```

Look for other things which would not start a table cell:

```
5026 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multicolumn}}{%
5027 {\let\mynext\LWR@donothing}}{ }%
5028 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multirow}}{%
5029 {\let\mynext\LWR@donothing}}{ }%
```

if come to an `\mrowcell`, this is a cell to be skipped over

```
5030 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\mrowcell}}{%
5031 {\let\mynext\LWR@donothing}}{ }%
5032 %
5033 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\hline}}{%
5034 {\let\mynext\LWR@donothing}}{ }%
5035 %
5036 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\toprule}}{%
5037 { }%
5038 \let\mynext\LWR@donothing}}{ }%
5039 %
5040 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\midrule}}{%
5041 {\let\mynext\LWR@donothing}}{ }%
5042 %
5043 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cmidrule}}{%
5044 {\let\mynext\LWR@donothing}}{ }%
5045 %
5046 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cline}}{%
5047 {\let\mynext\LWR@donothing}}{ }%
5048 %
5049 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\bottomrule}}{%
5050 {\let\mynext\LWR@donothing}}{ }%
5051 %
5052 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpprintonly}}
```

```

5053 {\let\mynext\LWR@donothing}{}%
5054 %
5055 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpHTMLonly}}
5056 {\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:


```

5057 \mynext%
5058 }

5059 \end{warpHTML}

```

55.18 `\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:

```

5060 \begin{warpall}
5061 \newcommand*{\mrowcell}{}
5062 \end{warpall}

```

55.19 New `\tabular` definition

for HTML output:

```

5063 \begin{warpHTML}

```

Env `LWR@tabular` [*⟨verticalposition⟩*] {*⟨colspecs⟩*}

The new tabular environment will be `\let` in `\LWR@LwarpStart`, since `siunitx` might redefine `tabular` in the user's document.

```

5064 \newenvironment*{LWR@tabular}[2]{}
5065 {%
5066 \LWR@traceinfo{tabular started}%
5067 \begingroup%
5068 \addtocounter{LWR@tabulardepth}{1}%

```

Not yet started a table row:

```

5069 \global\boolfalse{LWR@startedrow}%

```

Not yet doing an hline:

```
5070 \global\boolfalse{LWR@doinghline}%
```

Not yet doing a top/bottom rule:

```
5071 \global\boolfalse{LWR@doingtbrule}%
```

Have not yet found the end of tabular command:

```
5072 \boolfalse{LWR@exitingtabular}%
```

Create the table tag:

```
5073 \global\booltrue{LWR@intabularmetadata}%
5074 \LWR@forcenewpage
5075 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
5076 \LWR@parsetablecols{#2}%
```

Table col spec is: \LWR@tablecolspec which is a string of llccrr, etc.

Do not place the table inside a paragraph:

```
5077 \LWR@stoppars%
```

Track column # for setting text-align:

```
5078 \setcounter{LWR@tablecolspos}{1}%
```

Start looking for midrules:

```
5079 \LWR@clearmidrules%
```

\\ becomes a macro to end the table row:

```
5080 \let\\ \LWR@tabularendoffline%
```

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:

```
5081 \renewcommand*{\hline}{\LWR@domidrule\LWR@getmynexttoken}%
5082 \newcommand*{\midrule}{\LWR@domidrule\LWR@getmynexttoken}%
5083 \NewDocumentCommand{\cmidrule}{o d() m}%
5084 {\LWR@docmidrule[##1](##2){##3}\LWR@getmynexttoken}%
```

```

5085 \RenewDocumentCommand{\cline}{m}%
5086 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%
5087 \newcommand*{\toprule}{\LWR@dotbrule\LWR@getmynexttoken}%
5088 \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`.

```

5089 \let\multicolumn\LWR@tabledatamulticolumntag%
5090 \let\multirow\LWR@tabledatamultirowtag%
5091 \renewcommand*{\mrowcell}{\global\booltrue{\LWR@skippingmrowcell}}%
5092 \let\caption\LWR@longtabledatacaptiontag%

```

Reset for new processing:

```

5093 \global\boolfalse{\LWR@tableparcell}%
5094 \global\boolfalse{\LWR@skippingmrowcell}%

```

Look ahead for a possible table data cell:

```

5095 \LWR@getmynexttoken%
5096 }%

```

Ending the environment:

```

5097 {%
5098 \LWR@closetabledatacell%
5099 \LWR@htmlblocktag{/tr}%
5100 \LWR@htmlblocktag{/table}%
5101 \global\boolfalse{\LWR@intabularmetadata}%

5102 \addtocounter{\LWR@tabulardepth}{-1}%
5103 \endgroup%
5104 }

5105 \end{warpHTML}

```

55.20 Array

Pkg `array`

`array` is also automatically loaded by `siunitx`.

56 Cross-references

Sectioning commands have been emulated from scratch, so the cross-referencing commands are custom-written for them. Emulating both avoids several layers of patches.

The `zref` package is used to remember section name, file, and lateximage depth and number for each label.

Table 9 shows the data structures related to cross-referencing.

for HTML output: 5106 `\begin{warpHTML}`

56.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

5107 `\newcommand*{\@currentlabelname}{}`

`\LWR@stripperperiod` `{\text}\[<.]`

Removes a trailing period.

5108 `\def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%`

`\LWR@setlatestname` `{\object name}`

Removes `\label`, strips any final period, and remembers the result.

5109 `\newcommand*{\LWR@setlatestname}[1]{%`

Remove `\label` and other commands from the name, the strip any final period.
See `zref-titleref` and `getttitlestring`.

5110 `\GetTitleStringExpand{#1}%`

5111 `\edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%`

5112 `\edef\@currentlabelname{%`

5113 `\expandafter\LWR@stripperperiod\@currentlabelname%`

5114 `\ltx@empty.\ltx@empty\@nil%`

5115 `}%`

5116 `}`

Table 9: Cross-referencing data structures

Original L^AT_EX:	(print and HTML)
\refstepcounter: Steps the counter and sets \currentlabel.	
\currentlabel: \p@<ctr>\the<ctr> Updated by \refstepcounter.	
\label: Writes to the .aux file: \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 lwarp:	(HTML only)
\label: Adds HTML tags (section 56.3), plus \slabel data (section 56.2): 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 lwarp. See section 56.4.	
\ref and \nameref: Adds HTML tags. See section 56.4.	
Added by amsmath:	(print and HTML)
\label: Execution is delayed until the math environment is completed.	
\ltx@label: L ^A T _E X \label, (HTML: patched by lwarp,) later patched by cleveref.	
Added by cleveref:	(print and HTML)
\refstepcounter: Added: sets \cref@currentlabel.	
\cref@currentlabel: (<type>=<ctr> unless an alias is used): [<type>][\arabic{<ctr>}][<parent ctrs>]{\p@<ctr>\the<ctr>} Also see section 41.4 for use with footnotes.	
\label: Writes to the .aux file: \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 lwarp:	(HTML only)
\cref, etc.: Modified for lwarp. See section 65.	
\label inside math: See section 60.4.1.	
Footnotes: See \noteentry in section 41.4.	

56.2 Zref setup

See:

[http://tex.stackexchange.com/questions/57194/
extract-section-number-from-equation-reference](http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference)

Create a new property list called special:

```
5117 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
5118 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
5119 \zref@newprop{zLWR@htmlfilenumber}{%
5120 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilenumber}}%
5121 }%
```

Additional properties for lateximages:

```
5122 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
5123 \zref@newprop{zLWR@lateximagenumber}{\arabic{LWR@lateximagenumber}}
```

`zLWR@htmlfilenumber` property holds the file number or name

Add a `LWR@htmlfilenumber` property, and lateximage properties to special:

```
5124 \zref@addprop{special}{zLWR@name}
5125 \zref@addprop{special}{zLWR@htmlfilenumber}
5126 \zref@addprop{special}{zLWR@lateximagedepth}
5127 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
5128 \newcommand*{\LWR@spreff}[2]{%
5129 \zref@extractdefault{#1}{#2}{??}}
```

`\LWR@nameref` $\{\langle label \rangle\}$ Returns the section name for this label:

```
5130 \newcommand*{\LWR@nameref}[1]{%
5131 \LWR@spreff{#1}{zLWR@name}%
5132 }
```

`\LWR@htmlfileref` $\{\langle label \rangle\}$ Returns the file number for this label:

```

5133 \newcommand*{\LWR@htmlfileref}[1]{%
5134 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
5135 \LWR@sprep{#1}{z\LWR@htmlfilenumber}%
5136 }

```

`\LWR@lateximagedepthref` $\{\langle label \rangle\}$ Returns the `lateximagedepth` for this label:

```

5137 \newcommand*{\LWR@lateximagedepthref}[1]{%
5138 \LWR@sprep{#1}{z\LWR@lateximagedepth}%
5139 }

```

`\LWR@lateximagenumberref` $\{\langle label \rangle\}$ Returns the `lateximagenumber` for this label:

```

5140 \newcommand*{\LWR@lateximagenumberref}[1]{%
5141 \LWR@sprep{#1}{z\LWR@lateximagenumber}%
5142 }

```

`\LWR@splabel` $\{\langle label \rangle\}$ Sanitize the name and then creates the label:

```

5143 \newcommand*{\LWR@splabel}[1]{%
5144 \LWR@setlatestname{\@currentlabelname}%
5145 \zref@labelbylist{#1}{special}}

```

56.3 Labels

`\LWR@subsublabel` $\{\langle label \rangle\}$ Creates an HTML id tag.

```

5146 \newcommand*{\LWR@subsublabel}[1]{%

```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```

5147 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}{>}{0}}{%
5148 {}%
5149 {% not lateximage

```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```

5150 \ifbool{\LWR@doingstartpars}%
5151 {% pars allowed
5152 \ifbool{\LWR@doingapar}
5153 {% par started
5154 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5155 }% par started
5156 {% par not started

```

```

5157 \LWR@stoppars%
5158 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5159 \LWR@startpars%
5160 }% par not started
5161 }% pars allowed
5162 {% pars not allowed
5163 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5164 }% pars not allowed
5165 }% not lateximage
5166 }

```

`\LWR@newlabel` $\{\langle label \rangle\}$ [$\langle type \rangle$]

`\label` during HTML output when not in math mode, removing extra spaces around the label, as done by regular L^AT_EX `\label`.

`cleveref` later encases this to add its own cross-referencing.

The optional $\langle type \rangle$ is per the `ntheorem` package, and is ignored.

```

5167 \NewDocumentCommand{\LWR@newlabel}{m o}{%
5168 \LWR@traceinfo{LWR@newlabel: starting}%
5169 \LWR@traceinfo{LWR@newlabel: !#1!}%
5170 % \@bsphack%

```

Create a traditional LaTeX label, as modified by `cleveref`:

```

5171 \LWR@origlabel{#1}%

```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```

5172 \LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
5173 \LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
5174 \LWR@traceinfo{LWR@newlabel: LWR@htmlfilenumber is \theLWR@htmlfilenumber}%
5175 \LWR@splabel{#1}%
5176 \LWR@subsublabel{#1}%
5177 % \@esphack%
5178 \LWR@traceinfo{LWR@newlabel: done}%
5179 }

```

56.4 References

`\LWR@startref` $\{\langle label \rangle\}$ (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, `#` character, and a label.

```

5180 \newcommand*{\LWR@startref}[1]
5181 {%
5182 \edef\LWR@lhref{\LWR@lateximagedepthref{#1}}%
5183 \LWR@traceinfo{\LWR@startref A: !#1!}%

```

Create the filename part of the link:

```

5184 \LWR@htmltag{a href="%
5185 \LWR@traceinfo{\LWR@startref B}%
5186 \LWR@htmlrefsectionfilename{#1}%
5187 \LWR@traceinfo{\LWR@startref C}%
5188 \#%

```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```

5189 \LWR@traceinfo{\LWR@startref D: !#1!}%
5190 \ifthenelse{\equal{\LWR@lhref}{??}}%

```

“??” if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:

```

5191 {%
5192 \LWR@traceinfo{\LWR@startref D0: ??}%
5193 ??}%

```

If `LWR@lateximagedepth` is known. Use a `lateximage` if the depth is greater than zero, or a regular link otherwise:

```

5194 {%
5195 \LWR@traceinfo{\LWR@startref D1: \LWR@lhref}%
5196 \ifthenelse{\cinttest{\LWR@lhref}{>}{0}}%
5197 {%
5198 \LWR@traceinfo{\LWR@startref D2: \LWR@lhref}%
5199 lateximage\LWR@lateximagenumberref{#1}%
5200 }%
5201 {%
5202 \LWR@traceinfo{\LWR@startref D3}%
5203 #1%
5204 }%
5205 }%
5206 \LWR@traceinfo{\LWR@startref E}%

```

Closing quote:

```

5207 "}}%
5208 \LWR@traceinfo{\LWR@startref F}%
5209 }

```

`\LWR@subnewref` $\{\langle label \rangle\}$ $\{\langle label \text{ or } sub@label \rangle\}$

Factored for the subfig package. Uses the original label for the hyper-reference, but prints its own text, such as “1(b)”.

```
5210 \NewDocumentCommand{\LWR@subnewref}{m m}{%
5211 \LWR@traceinfo{\LWR@subnewref #1 #2}%
5212 \LWR@startref{#1}%
5213 \LWR@origref{#2}%
5214 \LWR@htmltag{/a}%
5215 }
```

`\ref` * $\{\langle label \rangle\}$ `\ref` is `\let` to `\LWR@newref`

`\LWR@newref` * $\{\langle label \rangle\}$ Create an internal document reference link, or without a link if starred per hyperref.

```
5216 \NewDocumentCommand{\LWR@newref}{s m}{%
5217 \LWR@traceinfo{\LWR@newref #2}%
5218 \IfBooleanTF{#1}%
5219 {\LWR@origref{#2}}%
5220 {\LWR@subnewref{#2}{#2}}%
5221 }
```

`\pagerefPageFor` Text for starred page references.

```
5222 \newcommand*{\pagerefPageFor}{see }
```

`\pageref` * $\{\langle label \rangle\}$ Create an internal document reference, or just the unlinked number if starred, per hyperref.

```
5223 \NewDocumentCommand{\LWR@newpageref}{s m}{%
5224 \IfBooleanTF{#1}%
5225 {(\pagerefPageFor\LWR@origref{#2})}%
5226 {(\cpageref{#2})}%
5227 }
```

`\nameref` $\{\langle label \rangle\}$

```
5228 \newcommand*{\nameref}[1]{%
5229 \LWR@traceinfo{\nameref A}%
5230 \LWR@startref{#1}%
5231 \LWR@traceinfo{\nameref B}%
5232 \LWR@nameref{#1}%
5233 \LWR@traceinfo{\nameref C}%
5234 \LWR@htmltag{/a}%

```

```
5235 \LWR@traceinfo{nameref D}%
5236 }
```

`\Nameref {⟨label⟩}` In print, adds the page number. In HTML, does not.

```
5237 \let\Nameref\nameref
```

56.5 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.

```
5238 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF:
5239 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Create a link with a text name:

`\LWR@subhyperref {⟨URL⟩} {⟨text⟩}`

```
5240 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
5241 \LWR@htmltag{a href="#1" target="_{blank}\LWR@orignewline}#2\LWR@htmltag{/a}%
5242 \LWR@ensuredoingapar%
5243 }
```

`\LWR@subhyperrefclass {⟨URL⟩} {⟨text⟩} {⟨htmlclass⟩}`

```
5244 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
5245 \LWR@htmltag{a href="#1"
5246 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
5247 \LWR@ensuredoingapar%
5248 }
```

`\href [⟨options⟩] {⟨URL⟩} {⟨text⟩}`

Create a link with accompanying text:

```

5249 \NewDocumentCommand{\LWR@hrefb}{0{} m +m}{%
5250 \LWR@subhyperref{#2}{#3}%
5251 \endgroup%
5252 \LWR@ensuredoingapar%
5253 }
5254
5255 \newcommand{\href}{%
5256 \LWR@ensuredoingapar%
5257 \begingroup%
5258 \catcode'\_ =12
5259 \LWR@hrefb%
5260 }
```

`\nolinkurl` $\{\langle URL \rangle\}$

Print the name of the link without creating the link:

```

5261 \newcommand*\LWR@nolinkurlb{[1]{#1\endgroup\LWR@ensuredoingapar}
5262
5263 \newcommand{\nolinkurl}{%
5264 \LWR@ensuredoingapar%
5265 \begingroup\catcode'\_ =12
5266 \LWR@nolinkurlb%
5267 }
```

`\url` $\{\langle URL \rangle\}$

Create a link whose text name is the address of the link:

```

5268 \newcommand*\LWR@urlb{[1]{%
5269 \href{#1}{#1}%
5270 \endgroup%
5271 \LWR@ensuredoingapar%
5272 }
5273
5274 \newcommand{\url}{%
5275 \LWR@ensuredoingapar%
5276 \begingroup\catcode'\_ =12
5277 \LWR@urlb%
5278 }
```

`\LWR@subinlineimage` $[\langle alttag \rangle] \{\langle class \rangle\} \{\langle filename \rangle\} \{\langle extension \rangle\} \{\langle style \rangle\}$

```

5279 \newcommand*\LWR@subinlineimage{[5] [] {%
5280 \ifthenelse{\equal{#1}{}}%
5281 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"}}}%

```

```
5282 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{}%  
5283 }  
  
5284 \end{warpHTML}
```

Table 10: Float data structures

For each `<type>` of float (figure, table, etc.) there exists the following:

counter `<type>`: A counter called `<type>`, such as `figure`, `table`.

`\<type>name`: Name. `\figurename` prints “Figure”, etc.

`\ext@<type>`: File extension. `\ext@figure` prints “lof”, etc.

`\fps@<type>`: Placement.

`\the<type>`: Number. `\thetable` prints the number of the table, etc.

`\p@<type>`: Parent’s number. Prints the number of the [within] figure, etc.

`\fnum@<type>`: Prints the figure number for the caption.

`\<type>name \the<type>`, “Figure 123”.

`\<type>`: Starts the float environment. `\figure` or `\begin{figure}`

`\end<type>`: Ends the float environment. `\endfigure` or `\end{figure}`

`\tf@<ext>`: The L^AT_EX file identifier for the output file.

`LWR@have<type>`: A boolean remembering whether a `\listof` was requested for a float of this type.

File with extension `lo<f,t,a-z>`: An output file containing the commands to build the `\listof<type><name>` “table-of-contents” structure.

Cross-referencing names: For `cleveref`’s `\cref` and related, `\crefname` and `\Crefname` assign human-readable names for references to this float type.

57 Floats

Floats are supported, although partially through emulation.

Table 10 shows the data structure associated with each `<type>` of float.

`\@makecaption` is redefined to print the float number and caption text, separated by `\CaptionSeparator`, which works with the `babel` package to adjust the caption separator according to the language. French, for example, uses an en-dash instead of a colon: “Figure 123 – Caption text”.

57.1 Float captions

for HTML output: 5285 \begin{warpHTML}

\LWR@floatbegin {<type>} [<placement>]

Begins a \newfloat environment.

```
5286 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
5287 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
5288
5289 === #1 begin
5290
5291 }{}%
5292 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
5293 \addtocounter{LWR@thisfloat}{1}%
5294 \booltrue{LWR@freezethisfloat}%
```

```
5295 \begingroup
```

Settings while inside the environment:

```
5296 \LWR@origraggedright
```

Open an HTML figure tag:

```
5297 \LWR@htmltag{figure id="autofloat-\arabic{LWR@thisfloat}" class="#1"}

5298 \renewcommand*{\@capttype}{#1}
5299 \caption@settype{#1}
5300 \LWR@startpars
5301 }
```

\@float Support packages which create floats directly.
 \@dblfloat

```
5302 \let\@float\LWR@floatbegin
5303 \let\@dblfloat\LWR@floatbegin
```

\LWR@floatend Ends a \newfloat environment.

```
5304 \newcommand*{\LWR@floatend}{%
5305 \LWR@stoppars%
5306 \LWR@htmllementend{figure}%
```

```

5307 \endgroup%
5308 \boolfalse{LWR@freezethisfloat}%
5309 \LWR@startpars%
5310 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
5311
5312 === end
5313
5314 }{}%
5315 }

```

`\end@float` Support packages which create floats directly.
`\end@dblfloat`

```

5316 \let\end@float\LWR@floatend
5317 \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.

```

5318 \newcounter{LWR@thisfloat}

```

Bool `LWR@freezethisfloat` Prevents multiple increments of `\LWR@thisfloat` inside a float.

```

5319 \newbool{LWR@freezethisfloat}
5320 \boolfalse{LWR@freezethisfloat}

```

`\LWR@maybeinthisfloat`

```

5321 \newcommand*{\LWR@maybeinthisfloat}{%
5322 \ifbool{LWR@freezethisfloat}{\addtocounter{LWR@thisfloat}{1}}%
5323 }

```

`\@capttype` Remembers which float type is in use.

```

5324 \newcommand*{\@capttype}{}

```

57.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```

5325 \AtBeginDocument{\providecommand*\CaptionSeparator{:-}}

```

`\@makecaption` $\{\langle name \text{ and } num \rangle\} \{\langle text \rangle\}$

Prints the float type and number, the caption separator, and the caption text.

```

5326 \AtBeginDocument{\renewcommand{\@makecaption}[2]{\#1\CaptionSeparator\#2}}

```

57.1.2 Caption and LOF linking and tracking

When a new HTML file is marked in the \LaTeX PDF file, the \LaTeX page number at that point is stored in `LWR@latestautopage`, (and the associated filename is remembered by the special \LaTeX labels). This page number is used to generate an `autofloat` HTML `<id>` in the HTML output at the start of the new HTML file. Meanwhile, there is a float counter used to generate an HTML `autofloat <id>` at the start of the float itself in the HTML file. The `autopage` and `autofloat` values to use for each float are written to the `.lof`, etc. files just before each float's entry. These values are used by `\l@figure`, etc. to create the HTML links in the List of Figures, etc.

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.

```
5327 \newcounter{LWR@nextautofloat}
5328 \newcounter{LWR@nextautopage}
```

```
\LWRsetnextfloat {\langle autopage \rangle} {\langle autofloat \rangle}
```

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.

```
5329 \newcommand*{\LWRsetnextfloat}[2]{%
5330 \setcounter{LWR@nextautopage}{#1}%
5331 \setcounter{LWR@nextautofloat}{#2}%
5332 }
```

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.

```
5333 \newcounter{LWR@latestautopage}
5334 \setcounter{LWR@latestautopage}{1}

5335 \let\LWR@origcaption@begin\caption@begin
5336 \let\LWR@origcaption@end\caption@end
5337 \let\LWR@orig@@par\@@par
```

`\LWR@caption@begin` Low-level patches to create HTML tags for captions.

```
5338 \newcommand{\LWR@caption@begin}
5339 {
5340 \LWR@traceinfo{LWR@caption@begin}%
```

Keep par and minipage changes local:

```
5341 \begingroup%
```

The caption code was not allowing the closing par tag:

```
5342 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a minipage or \parbox inside the caption:

```
5343 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{-}{-}%
```

```
5344 \RenewDocumentCommand{\parbox}{0{t} o 0{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
5345 \LWR@htmlblocktag{figcaption}%
```

```
5346 \LWR@origcaption@begin%
```

```
5347 }
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
5348 \newcommand{\LWR@caption@end}
```

```
5349 {%
```

```
5350 \LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
5351 \vspace*{\baselineskip}%
```

Closing tag:

```
5352 \LWR@htmlblocktag{/figcaption}%
```

```
5353 \endgroup%
```

```
5354 % \leavevmode% avoid bad space factor (0) error
```

```
5355 \LWR@traceinfo{\LWR@caption@end: done}%
```

```
5356 }
```

`\caption@begin` Low-level patches to create HTML tags for captions.

`\caption@end`

```
5357 \AtBeginDocument{
```

```
5358 \let\caption@begin\LWR@caption@begin
```

```
5359 \let\caption@end\LWR@caption@end
```

```
5360 }
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```

5361 \let\LWR@origcaptionlistentry\captionlistentry
5362
5363 \renewcommand*{\captionlistentry}{%
5364 \LWR@maybeinthisfloat%
5365 \LWR@ensuredoingapar%
5366 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5367 \LWR@origcaptionlistentry%
5368 }
5369
5370 \def\LWR@LTcaptionlistentry{%
5371 \LWR@ensuredoingapar%
5372 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5373 \bgroup
5374 \@ifstar{\egroup\LWR@LT@captionlistentry}% gobble *
5375 {\egroup\LWR@LT@captionlistentry}}%
5376 \def\LWR@LT@captionlistentry#1{%
5377 \caption@listentry\@firstoftwo[LTcaptype]{#1}}%

```

`\addcontentsline` Patched to write the autpage 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.

```

5378 \let\LWR@origaddcontentsline\addcontentsline
5379
5380 \renewcommand*{\addcontentsline}[3]{%
5381 \ifthenelse{\equal{#1}{toc}}{}{%
5382 \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{}
5383 \addtocontents{\@nameuse{ext@#2}}{%
5384 \protect\LWRsetnextfloat%
5385 {\arabic{LWR@latestautopage}}}%
5386 {\arabic{LWR@thisfloat}}}%
5387 }% addtocontents
5388 }% not toc
5389 \LWR@origaddcontentsline{#1}{#2}{#3}%
5390 }

```

`\captionof` Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

5391 \AtBeginDocument{
5392 \let\LWR@origcaptionof\captionof
5393
5394 \renewcommand*{\captionof}{%
5395 \LWR@maybeinthisfloat%
5396 \LWR@stoppars
5397 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5398 \LWR@origcaptionof%
5399 }

```

```
5400 }
5401 \end{warpHTML}
```

58 Table of Contents, LOF, LOT

This section controls the generation of the TOC, LOF, LOT.

The `.toc`, `.lof`, and `.lot` files are named by the source code `\jobname`.

In HTML, the printed tables are placed inside a div of class `.toc`, `.lof`, or `.lot`.

A “sidetoc” is provided which prints a subset of the TOC on the side of each page other than the homepage.

The regular \LaTeX infrastructure is used for TOC, along with some patches to generate HTML output.

for HTML output: 5402 `\begin{warpHTML}`

58.1 Reading and printing the TOC

```
\LWR@myshorttoc {\toc/lof/lot}
```

Reads in and prints the TOC/LOF/LOT at the current position. While doing so, makes the `@` character into a normal letter to allow formatting commands in the section names.

Unlike in regular \LaTeX , the file is not reset after being read, since the TOC may be referred to again in each HTML page, and is used for the sideTOC.

```
5403 \newcommand*\LWR@myshorttoc}[1]{
5404 \LWR@ensuredoingapar
```

Only if the file exists:

```
5405 \IfFileExists{\jobname.#1}{
```



Make `@` a regular letter. Many of the commands in the file will have `@` characters in them, so `@` must be made a regular letter.



For `pdf \LaTeX` , also change to `latin1` encoding. When reading back a file with accented characters, the encoding change seems to be required, rather than leaving it `utf8`.


```

5406 \begingroup
5407 % \ifxetexorluatex%
5408 % \else
5409 % \inputencoding{latin1}% currently disabled
5410 % \fi
5411 \makeatletter

```

Read in the TOC file:

```

5412 \@input{\jobname.#1}
5413 % \makeatother
5414 \endgroup
5415 }%
5416 {}%
5417 }

```

`\LWR@subtableofcontents` $\{\langle toc/lof/lot \rangle\} \{\langle sectionstarname \rangle\}$

Places a TOC/LOF/LOT at the current position.

```

5418 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%

```

Closes previous levels:

```

5419 \@ifundefined{chapter}
5420 {\LWR@closeprevious{\LWR@depthsection}}
5421 {\LWR@closeprevious{\LWR@depthchapter}}

```

Prints any pending footnotes so that they appear above the potentially large TOC:

```

5422 \LWR@printpendingfootnotes

```

Place the list into its own chapter (if defined) or section:

```

5423 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}

```

Create a new HTML nav containing the TOC/LOF/LOT:

```

5424 \LWR@htmlclass{nav}{#1}

```

Create the actual list:

```

5425 \LWR@myshorttoc{#1}

```

Close the nav:

```

5426 \LWR@htmlclassend{nav}{#1}
5427 }

```

Patch `\@starttoc` to encapsulate the TOC inside HTML tags:

```
5428 \let\LWR@orig@starttoc\@starttoc
5429
5430 \renewcommand{\@starttoc}[1]{
5431 \LWR@htmlelementclass{nav}{#1}
5432 \LWR@orig@starttoc{#1}
5433 \LWR@htmlelementclassend{nav}{#1}
5434 }
```

Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.

```
5435 \let\LWR@origtableofcontents\tableofcontents
5436 \let\LWR@origlistoffigures\listoffigures
5437 \let\LWR@origlistoftables\listoftables
5438
5439 \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:

```
5440 \ifbool{FormatWordProcessor}{}{
```

Copy the `.toc` file to `.sidetoc` for printing the sideTOC. The original `.toc` file is renewed when `\tableofcontents` is finished.

```
5441 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
5442 \LWR@printpendingfootnotes
5443 \LWR@origtableofcontents
5444 }
5445 }
5446 \renewcommand*{\listoffigures}{
5447 \ifbool{FormatWordProcessor}{}{
5448 \LWR@printpendingfootnotes
5449 \LWR@origlistoffigures
5450 }
5451 }
5452
5453 \renewcommand*{\listoftables}{
5454 \ifbool{FormatWordProcessor}{}{
5455 \LWR@printpendingfootnotes
5456 \LWR@origlistoftables
5457 }
5458 }
```

58.2 High-level TOC commands

`\listof` $\{\langle type \rangle\}$ $\{\langle title \rangle\}$

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.

```
5459 \NewDocumentCommand{\listof}{m +m}{%
5460 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
5461 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
5462 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
5463     \jobname.\csuse{ext@#1}\relax
5464 }
```

58.3 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.

```
5465 \end{warpHTML}
```

for HTML & PRINT: 5466 `\begin{warpall}`

Ctrl SideTOCDepth Controls how deep the side-TOC gets. Use a standard L^AT_EX section level similar to `tocdepth`.

```
5467 \newcounter{SideTOCDepth}
5468 \setcounter{SideTOCDepth}{1}
```

`\sidetocname` Holds the default name for the sideTOC.

```
5469 \newcommand{\sidetocname}{Contents}
```

```
5470 \end{warpall}
```

for HTML output: 5471 \begin{warpHTML}

\LWR@sidetoc Creates the actual side-TOC.

```
5472 \newcommand*\LWR@sidetoc){
5473 \LWR@forcenewpage
5474 \LWR@stoppars
5475
```

The entire sideTOC is placed into a nav of class `sidetoc`.

```
5476 \LWR@htmlclass{nav}{sidetoc}
5477
5478 \setcounter{tocdepth}{\value{SideTOCDepth}}
5479
```

The title is placed into a div of class `sidetoctitle`, and may contain paragraphs.

```
5480 \begin{BlockClass}{sidetoctitle}
5481 \sidetocname
5482 \end{BlockClass}
```

The table of contents is placed into a div of class `sidetoccontents`.

```
5483 \begin{BlockClass}{sidetoccontents}
5484 \LinkHome
5485
5486 \LWR@myshorttoc{sidetoc}
5487 \end{BlockClass}
5488 \LWR@htmlclassend{nav}{sidetoc}
5489 }
```

58.4 Low-level TOC line formatting

\numberline $\{\langle number \rangle\}$

(Called from each line in the `.aux`, `.lof` files.)

Record this section number for further use:

```
5490 \renewcommand*\numberline[1]{%
5491 \LWR@sectionnumber{#1}%
5492 }
```

`\hypertoc` $\langle 1: depth \rangle \langle 2: type \rangle \langle 3: name \rangle \langle 4: page \rangle$

Called by `\l@section`, etc. to create a hyperlink to a section.

The autopage label is always created just after the section.

#1 is depth

#2 is `section`, `subsection`, etc.

#3 the text of the caption

#4 page number

```
5493 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
5494 \ifthenelse{\cnttest{#1}{<=}}{\value{tocdepth}}{%
```

```
5495 \LWR@startpars%
```

Create an HTML link to `filename#autosec-(page)`, with text of the caption, of the given HTML class.

```
5496 \LWR@subhyperrefclass{%
```

```
5497 \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
```

```
5498 \LWR@stoppars%
```

```
5499 }
```

```
5500 {}
```

```
5501 }
```

Ctrl `lofdepth` TOC depth for figures.

```
5502 \newcounter{lofdepth}
```

```
5503 \setcounter{lofdepth}{1}
```

Ctrl `lotdepth` TOC depth for tables.

```
5504 \newcounter{lotdepth}
```

```
5505 \setcounter{lotdepth}{1}
```

`\hypertocfloat` $\langle 1: depth \rangle \langle 2: type \rangle \langle 3: ext\ of\ parent \rangle \langle 4: caption \rangle \langle 5: page \rangle$

#1 is depth

#2 is `figure`, `table`, etc.

#3 is `lof`, `lot`, of the parent.

#4 the text of the caption

#5 page number

```
5506 \newcommand{\hypertocfloat}[5]{%
5507 \LWR@startpars
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
5508 \@ifundefined{c@#3depth}{%
5509 \newcounter{#3depth}%
5510 \setcounter{#3depth}{1}%
5511 }{}%
```

Respond to `lofdepth`, etc.:

```
5512 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
5513 \ifthenelse{\cnttest{#1}{<=}}{\arabic{#3depth}}{}%
5514 \LWR@startpars%
```

Create an HTML link to `filename#autofloat-(float number)`, with text of the caption, of the given HTML class.

```
5515 \LWR@subhyperrefclass{%
5516 \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
5517 \#autofloat-\arabic{LWR@nextautofloat}}%
5518 {#4}{toc#2}%
5519 \LWR@stoppars%
5520 }{}%
5521 }
```

Automatically called by `\contentsline`:

```
5522 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
5523 \DeclareDocumentCommand{\l@chapter}{m m}
5524 {\hypertoc{0}{chapter}{#1}{#2}}
5525 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
5526 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
5527 \renewcommand{\l@subsubsection}[2]
5528 {\hypertoc{3}{subsubsection}{#1}{#2}}
5529 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
5530 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
5531 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
5532 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

5533 \end{warpHTML}
```

59 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: 5534 `\begin{warpHTML}`

```
5535 \newcounter{LWR@autoindex}
5536 \setcounter{LWR@autoindex}{0}
5537
5538 \newcounter{LWR@autoglossary}
5539 \setcounter{LWR@autoglossary}{0}
```

`\LWR@indexsection` Controls whether the index will be in a section or a chapter, depending on the documentclass.

```
5540 \@ifundefined{chapter}
5541 {\newcommand*{\LWR@indexsection}{\section{\indexname}}}
5542 {\newcommand*{\LWR@indexsection}{\chapter{\indexname}}}
```

`\printindex`

```
5543 \let\LWR@origprintindex\printindex
5544
5545 \renewcommand*{\printindex}
5546 {
5547 \LWR@indexsection
5548 \LWR@startpars
5549 \LWR@origprintindex
5550 }
```

Env `theindex`

```
5551 \renewenvironment*{theindex}{%
5552 \let\item\LWR@indexitem%
5553 \let\subitem\LWR@indexsubitem%
5554 \let\subsubitem\LWR@indexsubsubitem%
5555 }{}
```

`\LWR@indexitem`

```

5556 \newcommand{\LWR@indexitem}{
5557
5558 \InlineClass@indexitem}{}
5559 }

```

\LWR@indexitem

```

5560 \newcommand{\LWR@indexsubitem}{
5561
5562 \InlineClass@indexsubitem}{}
5563 }

```

\LWR@indexitem

```

5564 \newcommand{\LWR@indexsubsubitem}{
5565
5566 \InlineClass@indexsubsubitem}{}
5567 }

```

\@wrindex {*{term}*} Redefined to write the LWR@latestautopage counter instead of page

```

5568 \def\@wrindex#1{%
5569 \addtocounter{LWR@autoindex}{1}%
5570 \LWR@newlabel{LWRindex-\theLWR@autoindex}%
5571 \protected@write\@indexfile{%
5572 {\string\indexentry{#1}{\theLWR@autoindex}}%
5573 \endgroup
5574 \@esphack}

```

\@wrglossary {*{term}*} Redefined to write the LWR@latestautopage counter instead of page

```

5575 \def\@wrglossary#1{%
5576 \addtocounter{LWR@autoglossary}{1}%
5577 \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
5578 \protected@write\@glossaryfile{%
5579 {\string\glossaryentry{#1}{\theLWR@autoglossary}}%
5580 \endgroup
5581 \@esphack}

```

\hyperindexref {*{autosecnumber}*}

\hyperindexref{web address} is inserted into *.ind by the xindy style file
lwarp.xdy

```

5582 \newcommand*\hyperindexref[1]{\nameref{LWRindex-#1}}

```



```
5583 \end{warpHTML}
```

for PRINT output: A null command for print mode, in case `hyperref` was not used:

```
5584 \begin{warpprint}
5585 \newcommand{\hyperindexref}[1]{#1}
5586 \end{warpprint}
```

for HTML & PRINT: For the `glossaries` package, try to prevent an error where `\glo@name` was not found:

```
5587 \begin{warpall}
5588 \providecommand{\glo@name}{}
5589 \end{warpall}
```

60 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 \LaTeX into the initial PDF file using the current font⁹, then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by \LaTeX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML ALT tag carries the \LaTeX code which generated the math, allowing copy/paste of the \LaTeX math expression into other documents.

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

SVG files As currently implemented, each instance of math creates a new SVG file. In text with many references to math variables, this can result in a large number of files with duplicate content. In the future, some method of content-based naming and checksumming may be used to remove the need for duplicate files.

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

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.

⁹See section 165 regarding fonts and fractions.

MathML Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 7 regarding EPUB output with MathJax.

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

Prog **MathJax**

When MathJax is enabled, math is rendered twice:

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

MathJax limitations Limitations when using MathJax include:

Prog **MathJax**

chapter numbers

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

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

subequations

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

footnotes in math

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

lateximage

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

siunitx

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

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

 **siunitx inside an equation**

\LaTeX macros

- MathJax does not automatically support custom \LaTeX macros, but they may be set up by the user.

custom MathJax macros As an example of using custom 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: 5590 `\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.

```

5591 \let\LWR@origtextdollar\$
5592
5593 \renewcommand*{\$}{%
5594 \ifthenelse{\cinttest{\value{LWR@lateximagedepth}}{>}{0}}{
5595 {\LWR@origtextdollar}%
5596 {\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
5597 }

```

Ctr LWR@externalfilecnt Counter for the external files which are generated and then referenced from the HTML:

```
5598 \newcounter{LWR@externalfilecnt}
```

60.1 Inline and display math

```

5599 \let\LWR@origdollar=$
5600 \let\LWR@secondorigdollar=% balance for editor syntax highlighting

5601 \let\LWR@origopenparen\(\
5602 \let\LWR@origcloseparen\)
5603 \let\LWR@origopenbracket\[
5604 \let\LWR@origclosebracket\]

```

`$` Redefine the dollar sign to place math inside a `lateximage`, or use MathJax:
`$$`

```

5605 \begingroup
5606 \catcode'\$=\active%
5607 \protected\def$\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

`\LWR@doubledollar` Redefine the double dollar sign to place math inside a `lateximage`, or use MathJax:

```

5608 \gdef\LWR@doubledollar$#1$$#{
5609 \ifbool{mathjax}%

```

For MathJax, print the math between `\[` and `\]`:

```

5610 {\textbackslash[\LWR@HTMLsanitize{#1}\textbackslash]}

```

For SVG, print the math inside a `lateximage`, with an ALT tag of the \LaTeX code:

```

5611 {% not mathjax
5612
5613 \begin{lateximage}%
5614 [\textbackslash{[] \LWR@HTMLsanitize{#1} \textbackslash{[]}]%
5615 \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
5616 \end{lateximage}
5617
5618 }
5619 }%

```

`\LWR@singledollar` Redefine the single dollar sign to place math inside a `lateximage`, or use MathJax:

```

5620 \gdef\LWR@singledollar#1${%
5621 \ifbool{mathjax}%

```

For MathJax, print the math between `\(` and `\)`:

```

5622 {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}

```

For SVG, print the math inside a `lateximage`, with an ALT tag of the \LaTeX code:

```

5623 {% not mathjax
5624 \begin{lateximage}%
5625 [\textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)]%
5626 \LWR@origdollar#1\LWR@origdollar%
5627 \end{lateximage}%
5628 }%
5629 }%

```

`\(` Redefine to the above dollar macros.

`\)`

```

5630 \gdef\(#1\){$#1$}
5631 \gdef\[#1\]{$$#1$$}

```

```
5632
5633 \endgroup
```

Remove the old `math` and `displaymath` environments:

```
5634 \let\math\relax
5635 \let\endmath\relax
5636 \let\displaymath\relax
5637 \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`.

```
5638 \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`.

```
5639 \NewEnviron{displaymath}{\expandafter\[\BODY\]\@ignoretrue}
```

When the document begins, the dollar sign must be made active to trigger the new math macros:

```
5640 \AtBeginDocument{\catcode'\$=\active}
```

60.2 MathJax support

Ctr `LWR@nextequation` Used to add one to compute the next equation number.

```
5641 \newcounter{LWR@nextequation}
```

`\LWR@syncmathjax` Sets the MathJax equation format and number for the following equations.

These MathJax commands are printed inside “`\(`” and “`\)`” characters. They are printed to HTML output, not interpreted by \LaTeX .

```
5642 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
5643 \ifcsdef{thechapter}{
5644 \BlockClassSingle{hidden}{
5645 \textbackslash(
```

```

5646 \textbackslash{}seteqsection \{\thechapter\}
5647 \textbackslash)
5648 }
5649 }
5650 {}% not using chapters
5651

```

MathJax doesn't allow setting the equation number to 1:

```

5652 \ifthenelse{\cnttest{\value{equation}}>0}
5653 {

```

Tell MathJax that the next set of equations begins with the current L^AT_EX equation number, plus one.

```

5654 \setcounter{LWR@nextequation}{\value{equation}}
5655 \addtocounter{LWR@nextequation}{1}

```

Place the MathJax command inside “\(>” and “\)>” characters, to be printed to HTML, not interpreted by L^AT_EX.

```

5656 \BlockClassSingle{hidden}{
5657 \textbackslash(
5658 \textbackslash{}seteqnumber \{\arabic{LWR@nextequation}\}
5659 \textbackslash)>
5660 }
5661 {}% not eq > 1
5662 }

```

LWR@restoremathlatexformatting While producing math, use regular L^AT_EX formatting instead of HTML tags.

```

5663 \newcommand*{\LWR@restoremathlatexformatting}{%
5664 \let\hspace\LWR@orighspace%
5665 \let\rule\LWR@origrule%
5666 \let\,\LWR@origcomma% disable HTML short unbreakable space
5667 \let\textit\LWR@origtextit%
5668 \let\textbf\LWR@origtextbf%
5669 \let\texttt\LWR@origtexttt%
5670 \let\textsc\LWR@origtextsc%
5671 \let\textsf\LWR@origtextsf%
5672 \let\textrm\LWR@origtextrm%
5673 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}}%
5674 \let\textsuperscript\LWR@origtextsuperscript%
5675 \let\textsubscript\LWR@origtextsubscript%
5676 \let~\LWR@origtilde%
5677 \let\enskip\LWR@origenskip%
5678 \let\quad\LWR@origquad%
5679 \let\qquad\LWR@origqquad%
5680 }

```

`\LWR@hidelatexequation` $\{\langle environment \rangle\} \{\langle contents \rangle\}$

Creates the \LaTeX version of the equation inside an HTML comment.

5681 `\NewDocumentCommand{\LWR@hidelatexequation}{m +m}{\%`

Stop HTML paragraph handling and open an HTML comment:

5682 `\LWR@stoppars`

5683 `\LWR@htmlopencomment`

5684

Start the LaTeX math environment inside the HTML comment:

5685 `\begingroup`

5686 `\csuse{\LWR@orig#1}`

While in the math environment, restore various commands to their \LaTeX meanings.

5687 `\LWR@restoremathlatexformatting`

See `\LWR@htmlmathlabel` in section [60.4.1](#).

Print the contents of the equation:

5688 `#2`

End the \LaTeX math environment inside the HTML comment:

5689 `\csuse{\LWR@origend#1}`

5690 `\endgroup`

5691

Close the HTML comment and resume HTML paragraph handling:

5692 `\LWR@htmlclosecomment`

5693 `\LWR@startpars`

5694 `}`

`\LWR@addmathjax` $\{\langle environment \rangle\} \{\langle contents \rangle\}$

Given the name of a math environment and its contents, create a MathJax instance.
The contents are printed to HTML output, not interpreted by \LaTeX .

5695 `\NewDocumentCommand{\LWR@addmathjax}{m +m}{\%`

5696

Enclose the MathJax environment inside printed “\($\)” and “\($\)” characters.$$

```
5697 \textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
5698 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
5699 \textbackslash{}end\{#1\}
```

```
5700
```

```
5701 }
```

60.3 Equation environment

Remember existing `equation` environment:

```
5702 \let\LWR@origequation\equation
```

```
5703 \let\LWR@origendequation\endequation
```

Remove existing `equation` environment:

```
5704 \let\equation\relax
```

```
5705 \let\endequation\relax
```

Env `equation` The new `equation` environment is created with `\NewEnviron` (from the `environ` package), which stores the contents of its environment in a macro called `\BODY`.

For SVG math output, the contents are typeset using the original `equation` inside a `lateximage`, along with an ALT tag containing a detokenized copy of the \LaTeX source for the math.

For MathJax output, the contents are typeset in an original `equation` environment placed inside a HTML comment, with special processing for `\labels`. The contents are also printed to the HTML output for processing by the MathJax script.

```
5706 \NewEnviron{equation}{%
```

```
5707
```

```
5708 \ifbool{mathjax}
```

MathJax output:

```
5709 {
```

Print commands to synchronize MathJax's equation number and format to the current \LaTeX chapter/section and equation number:

5710 `\LWR@syncmathjax`

Print the \LaTeX math inside an HTML comment:

5711 `\LWR@hidelatexequation{equation}{\BODY}`
 5712 `}`

SVG output: Create the `lateximage` along with an HTML ALT tag having an equation number, the \LaTeX equation environment commands, and the contents of the environment's `\BODY`.

5713 `{% not mathjax`

Begin the `lateximage` with an ALT tag containing the math source:

5714 `\begin{lateximage}[(\theequation) \textbackslash{begin\{equation\}}]{%`
 5715 `\LWR@HTMLsanitizeexpand{\detokenize\expandafter\BODY}}{%`
 5716 `\textbackslash{end\{equation\}}]{% alt tag`

Create the actual \LaTeX -formatted equation inside the `lateximage` using the contents of the environment.

5717 `\LWR@origequation`
 5718 `\BODY% contents collected by NewEnviron`
 5719 `\LWR@origendequation`
 5720 `\end{lateximage}%`
 5721 `}`
 5722

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

5723 `}{\ifbool{mathjax}{\LWR@addmathjax{equation}{\BODY}}{}}`

60.4 AMS Math environments

60.4.1 Support macros

Bool `LWR@amsmultline` True if processing a multiline environment.

To compensate for multiline-specific code, `LWR@amsmultline` is used to add extra horizontal space in `\LWR@htmlmathlabel` if is used in an `amsmath` environment which is not a multiline environment and not an equation.

5724 `\newbool{LWR@amsmultline}`
 5725 `\boolfalse{LWR@amsmultline}`

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

`lwarp` points `\ltx@label` here. This is used by `\label` when inside a \LaTeX AMS math environment's math display environment.

`\LWR@origltx@label` points to the \LaTeX original, modified by `lwarp`, then by `amsmath`, then by `cleveref`.

```
5726 \newcommand*\LWR@htmlmathlabel}[1]{%
5727 \LWR@traceinfo{\LWR@htmlmathlabel #1}%
5728 \ifbool{mathjax}{%
```

The combined \LaTeX & HTML label is printed in a `\text` field:

```
5729 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
5730 \ifbool{\LWR@amsmultline}{\hspace*{\totwidth@}}
```

Temporarily end the HTML comment, insert the \LaTeX & HTML label, then resume the HTML comment. `\@firstofone` is required to remove extra braces introduced by the `amsmath` package.)

```
5731 \LWR@htmlclosecomment%
5732 \LWR@origltx@label{#1}%
5733 \LWR@htmlopencomment%
5734 }% text
5735 }% mathjax
5736 {%
5737 \LWR@origltx@label{#1}%
5738 }%
5739 }
```

`\LWR@beginhideamsmath` Starts hiding \LaTeX math inside an HTML comment.

```
5740 \newcommand*\LWR@beginhideamsmath){
5741 \LWR@stoppars
5742
5743 \LWR@htmlopencomment
5744
5745 \begingroup
5746 \LWR@restoremathlatexformatting
5747 }
```

`\LWR@endhideamsmath` Ends hiding \LaTeX math inside an HTML comment.

```

5748 \newcommand*{\LWR@endhideamsmath}{
5749 \endgroup
5750
5751 \LWR@htmlclosecomment
5752
5753 \LWR@startpars
5754 }

```

60.4.2 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 `LWRAMSMATHBODY` argument telling `lateximage` to use as the HTML ALT tag the environment's contents which were automatically captured by the $\mathscr{A}\mathscr{M}\mathscr{S}$ 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 `multline`

```

5755 \BeforeBeginEnvironment{multline}{
5756 \ifbool{mathjax}
5757 {
5758 \LWR@syncmathjax
5759 \booltrue{LWR@amsmultline}
5760 \LWR@beginhideamsmath
5761 }
5762 {
5763 \lateximage[LWRAMSMATHBODY]
5764 }
5765 }
5766
5767 \AfterEndEnvironment{multline}{
5768
5769 \ifbool{mathjax}
5770 {
5771 \LWR@endhideamsmath
5772 \boolfalse{LWR@amsmultline}
5773 \LWR@addmathjax{multline}{\the\@envbody}
5774 }
5775 {\endlateximage}
5776
5777 }

```

Env **multline***

```

5778 \BeforeBeginEnvironment{multline*}{
5779 \ifbool{mathjax}
5780 {
5781 \LWR@syncmathjax
5782 \booltrue{LWR@amsmultline}
5783 \LWR@beginhideamsmath
5784 }
5785 {
5786 \lateximage[LWRMSMATHBODY]
5787 }
5788 }
5789
5790 \AfterEndEnvironment{multline*}{
5791
5792 \ifbool{mathjax}
5793 {
5794 \LWR@endhideamsmath
5795 \boolfalse{LWR@amsmultline}
5796 \LWR@addmathjax{multline*}{\the\@envbody}
5797 }
5798 {\endlateximage}
5799
5800 }
5801

```

Env **gather**

```

5802 \BeforeBeginEnvironment{gather}{
5803 \ifbool{mathjax}
5804 {
5805 \LWR@syncmathjax
5806 \boolfalse{LWR@amsmultline}
5807 \LWR@beginhideamsmath
5808 }
5809 {
5810 \lateximage[LWRMSMATHBODY]
5811 }
5812 }
5813
5814 \AfterEndEnvironment{gather}{
5815
5816 \ifbool{mathjax}
5817 {
5818 \LWR@endhideamsmath
5819 \LWR@addmathjax{gather}{\the\@envbody}
5820 }
5821 {\endlateximage}

```

```
5822
5823 }
```

Env **gather***

```
5824 \BeforeBeginEnvironment{gather*}{
5825 \ifbool{mathjax}
5826 {
5827 \LWR@syncmathjax
5828 \boolfalse{LWR@amsmultline}
5829 \LWR@beginhideamsmath
5830 }
5831 {
5832 \lateximage[LWR@SMATHBODY]
5833 }
5834 }
5835
5836 \AfterEndEnvironment{gather*}{
5837
5838 \ifbool{mathjax}
5839 {
5840 \LWR@endhideamsmath
5841 \LWR@addmathjax{gather*}{\the\@envbody}
5842 }
5843 {\endlateximage}
5844
5845 }
```

Env **align**

```
5846 \BeforeBeginEnvironment{align}{
5847 \ifbool{mathjax}
5848 {
5849 \LWR@syncmathjax
5850 \boolfalse{LWR@amsmultline}
5851 \LWR@beginhideamsmath
5852 }
5853 {
5854 \lateximage[LWR@SMATHBODY]
5855 }
5856 }
5857
5858 \AfterEndEnvironment{align}{
5859
5860 \ifbool{mathjax}
5861 {
5862 \LWR@endhideamsmath
5863 \LWR@addmathjax{align}{\the\@envbody}
5864 }
```

```
5865 {\endlateximage}  
5866  
5867 }
```

Env **align***

```
5868 \BeforeBeginEnvironment{align*}{  
5869 \ifbool{mathjax}  
5870 {  
5871 \LWR@syncmathjax  
5872 \boolfalse{LWR@amsmultline}  
5873 \LWR@beginhideamsmath  
5874 }  
5875 {  
5876 \lateximage[LWR@SMATHBODY]  
5877 }  
5878 }  
5879  
5880 \AfterEndEnvironment{align*}{  
5881  
5882 \ifbool{mathjax}  
5883 {  
5884 \LWR@endhideamsmath  
5885 \LWR@addmathjax{align*}{\the\@envbody}  
5886 }  
5887 {\endlateximage}  
5888  
5889 }
```

Env **flalign**

```
5890 \BeforeBeginEnvironment{flalign}{  
5891 \ifbool{mathjax}  
5892 {  
5893 \LWR@syncmathjax  
5894 \boolfalse{LWR@amsmultline}  
5895 \LWR@beginhideamsmath  
5896 }  
5897 {  
5898 \lateximage[LWR@SMATHBODY]  
5899 }  
5900 }  
5901  
5902 \AfterEndEnvironment{flalign}{  
5903  
5904 \ifbool{mathjax}  
5905 {  
5906 \LWR@endhideamsmath  
5907 \LWR@addmathjax{flalign}{\the\@envbody}
```

```

5908 }
5909 {\endlateximage}
5910
5911 }

```

Env **flalign***

```

5912 \BeforeBeginEnvironment{flalign*}{
5913 \ifbool{mathjax}
5914 {
5915 \LWR@syncmathjax
5916 \boolfalse{LWR@amsmultline}
5917 \LWR@beginhideamsmath
5918 }
5919 {
5920 \lateximage[LWR@SMATHBODY]
5921 }
5922 }
5923
5924 \AfterEndEnvironment{flalign*}{
5925
5926 \ifbool{mathjax}
5927 {
5928 \LWR@endhideamsmath
5929 \LWR@addmathjax{flalign*}{\the\@envbody}
5930 }
5931 {\endlateximage}
5932
5933 }

5934 \end{warpHTML}

```

61 Lateximages

A `\lateximage` is typeset on its own PDF page inside an HTML comment which starts on the preceeding page and ends on following page, and instructions are written to `lateximage.txt` for `lwarpmk` to extract the `\lateximage` from the page of the PDF file then generate an accompanying `.svg` file image file. Meanwhile, instructions to show this image are placed into the HTML file after the comment.

An HTML span is created to hold both the HTML comment, which will have the `pdftotext` conversion, and also the link to the final `.svg` image.

A \LaTeX label is used to remember which PDF page has the image. A label is used because footnotes, endnotes, and pagenotes may cause the image to appear

at a later time. The label is declared along with the image, and so it correctly remembers where the image finally ended up.

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

for HTML output: 5935 `\begin{warpHTML}`

Ctr `LWR@lateximagenumber` Sequence the images.

```
5936 \newcounter{LWR@lateximagenumber}
5937 \setcounter{LWR@lateximagenumber}{0}
```

Ctr `LWR@lateximagedepth` Do not create `\lateximage` inside of `\lateximage`.

```
5938 \newcounter{LWR@lateximagedepth}
5939 \setcounter{LWR@lateximagedepth}{0}
```

Declare the `\LWR@file` for writing to generate file `lateximages.txt`:

```
5940 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
```

A few utility macros to write special characters:

```
5941 \edef\LWR@hashmark{\string#} % for use in \write
5942 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctr `LWR@Lipage` Used to reference the PDF page number of a `lateximage` to be written into `lateximages.txt`.

```
5943 \newcounter{LWR@Lipage}
```

```
5944 \end{warpHTML}
```

for HTML & PRINT: 5945 `\begin{warpall}`

`\LateximageFontSizeName` Declares how large to write text in the `\lateximage`. The `.svg` file text size should blend well with the surrounding HTML text size.

Do not include the leading backslash in the name.

```
5946 \newcommand*{\LateximageFontSizeName}{large}
```

```
5947 \end{warpall}
```

for HTML output: 5948 `\begin{warpHTML}`

`\LWR@HTMLsanitize` $\{\langle text \rangle\}$

Math expressions are converted to `lateximages`, and some math environments may contain “&”, “<”, or “>”, which should not be allowed inside an HTML ALT tag, so must convert them to HTML entities.

Two versions follow, depending on expansion needs. There may be a better way...

```

5949 \newcommand{\LWR@HTMLsanitize}[1]{%
5950 \protect\StrSubstitute{\detokenize{#1}}{%
5951 {\detokenize{&}}{%
5952 {\detokenize{&}}}{\LWR@strresult}%
5953 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
5954 {\detokenize{<}}{%
5955 {\detokenize{<}}}{\LWR@strresult}%
5956 [\LWR@strresult]%
5957 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
5958 {\detokenize{>}}{%
5959 {\detokenize{>}}}{\LWR@strresult}%
5960 [\LWR@strresult]%
5961 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
5962 {\detokenize{##}}{%
5963 {\#}}{%
5964 [\LWR@strresult]%
5965 \LWR@strresult%
5966 }

```

`\LWR@HTMLsanitizeexpand` $\{\langle text \rangle\}$

This version expands the argument before sanitizing it.

```

5967 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
5968 \protect\StrSubstitute{\detokenize\expandafter{#1}}{%
5969 {\detokenize{&}}{%
5970 {\detokenize{&}}}{\LWR@strresult}%
5971 [\LWR@strresult]%
5972 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
5973 {\detokenize{<}}{%
5974 {\detokenize{<}}}{\LWR@strresult}%
5975 [\LWR@strresult]%
5976 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
5977 {\detokenize{>}}{%
5978 {\detokenize{>}}}{\LWR@strresult}%
5979 [\LWR@strresult]%
5980 \LWR@strresult%
5981 }

```

Env `lateximage` $[\langle alttag \rangle]$

```

5982 \NewDocumentEnvironment{lateximage}{0{image}}{%
5983 \LWR@traceinfo{lateximage: starting on page \arabic{page}}%
5984 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%

```

If nesting inside an already-existing lateximage, simply record one more level:

```

5985 {%
5986 \addtocounter{LWR@lateximagedepth}{1}%
5987 }%

```

Otherwise, this is the outer-most lateximage:

```

5988 {% start of outer-most lateximage

```

Starting a new lateximage:

```

5989 \addtocounter{LWR@lateximagenumber}{1}%
5990 \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%

```

While inside a lateximage, do not use mathjax:

```

5991 \boolfalse{mathjax}

```

Be sure that are doing a paragraph:

```

5992 \LWR@ensuredoingapar%

```

Next file:

```

5993 \addtocounter{LWR@externalfilecnt}{1}%
5994 \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%

```

Figure out what the next page number will be:

```

5995 \setcounter{pageref}{LWR@Lipage}{LWR@lateximage\theLWR@lateximagenumber}%
5996 \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:

```

5997 \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
5998 class="lateximagesource">{}} \LWR@orignewline

```

Write instructions to the `lateximages.txt` file:

```

5999 \immediate\write\LWR@file{|\theLWR@Lipage|\theLWR@externalfilecnt|}%

```

Place an open comment tag at the bottom of page; footnotes will be above this tag. This will hide any traces of the lateximage PDF page which were picked up by pdftotext.

```
6000 \LWR@htmlopencomment%
6001 \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
6002 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
6003 \LWR@origminipage{6in}%
6004 \csuse{LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for \hspace, etc. inside a lateximage.

```
6005 \let\hspace\LWR@orighspace%
6006 \let\rule\LWR@origrule%
6007 \let\,\LWR@origcomma% disable HTML short unbreakable space
6008 \let\textit\LWR@origtextit%
6009 \let\textbf\LWR@origtextbf%
6010 \let\texttt\LWR@origtexttt%
6011 \let\textsc\LWR@origtextsc%
6012 \let\textsf\LWR@origtextsf%
6013 \let\textrm\LWR@origtextrm%
6014 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
6015 \let\textsuperscript\LWR@origtextsuperscript%
6016 \let\textsubscript\LWR@origtextsubscript%
6017 \let~\LWR@origtilde%
6018 \let\enskip\LWR@origenskip%
6019 \let\quad\LWR@origquad%
6020 \let\qquad\LWR@origqquad%
6021 \let\tabular\LWR@origtabular%
6022 \let\endtabular\LWR@origendtabular%
6023 \let\newline\LWR@orignewline%
6024 \LWR@origlabel{LWR\lateximage\arabic{LWR@lateximagenumber}}}%
6025 }% end of outer-most lateximage
6026 }% end of \begin{lateximage}
6027 {% start of \end{lateximage}
6028 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
6029 {%
6030 \addtocounter{LWR@lateximagedepth}{-1}%
6031 }%
```

if this is the outer-most lateximage:

```
6032 {% end of outer-most lateximage
```

Finish the lateximage minipage and start a new PDF page:

```
6033 \LWR@origendminipage%
6034 \LWR@orignewpage%
6035 \LWR@origscriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by `pdftotext`:

```
6036 \LWR@htmlclosecomment{}\LWR@orignewline%
6037 \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}%
```

Create a link to the lateximage, allowing its natural height:

If the alt tag is given as “LWRAMSMATHBODY”, then use the text collected by the `amsmath` `multline`, `gather`, or `align` environments.

```
6038 \ifthenelse{\equal{#1}{LWRAMSMATHBODY}}{%
6039 {%
6040 \LWR@subinlineimage[%
6041 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}}%
6042 ]%
6043 {lateximage}%
6044 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}%
6045 {svg}%
6046 {}%
6047 }%
6048 {%
6049 \LWR@subinlineimage[#1]{lateximage}%
6050 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}{svg}{}%
6051 }%
6052 % \LWR@orignewline% Removed to prevent extra space.
```

Be sure that are doing a paragraph:

```
6053 \LWR@ensuredoingapar%
```

Close the HTML span which has the `pdftotext` comment and also the link to the `.svg` image:

```
6054 \LWR@htmltag{/span}%
6055 \ifbool{HTMLDebugComments}{%
6056 \LWR@htmlcomment{End of lateximage}%
6057 }{}%
6058 % \LWR@orignewline% Removed to prevent extra space.
```

Undo one lateximage level:

```
6059 \addtocounter{LWR@lateximagedepth}{-1}%
6060 }% end of outer-most lateximage
6061 \LWR@traceinfo{lateximage: done}
6062 }%
6063 \end{warpHTML}
```

for PRINT output: 6064 \begin{warpprint}
6065 \newenvironment{lateximage}[1][\minipage{\linewidth}]{\endminipage}
6066 \end{warpprint}

62 center, flushleft, flushright

for HTML output: 6067 \begin{warpHTML}

Env **center** Replace center functionality with CSS tags:

```
6068 \renewenvironment*{center}
6069 {
6070 \LWR@forcenewpage
6071 \BlockClass{center}
6072 }
6073 {\endBlockClass}
```

Env **flushright**

```
6074 \renewenvironment*{flushright}
6075 {
6076 \LWR@forcenewpage
6077 \BlockClass{flushright}
6078 }
6079 {\endBlockClass}
```

Env **flushleft**

```
6080 \renewenvironment*{flushleft}
6081 {
6082 \LWR@forcenewpage
6083 \BlockClass{flushleft}
6084 }
6085 {\endBlockClass}
```

```
6086 \end{warpHTML}
```

63 Siunitx

Pkg **siunitx**

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

for HTML output: 6087 `\begin{warpHTML}`

Options for siunitx:

```
6088 \PassOptionsToPackage{
6089 detect-mode=true,
6090 per-mode=symbol,% fraction is not seen by pdftotext
6091 text-celsius = {\HTMLentity{deg}C},
6092 text-degree = {\HTMLentity{deg}},
6093 }{siunitx}

6094 \end{warpHTML}
```

64 Graphics

Pkg **graphics**

Pkg **graphicx**

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

⚠ **viewports**

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

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

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

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

⚠ image file types

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

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

⚠ browser support

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

for HTML output: 6095 `\begin{warpHTML}`

64.1 `\graphicspath`

`\graphicspath` `{\path}`

6096 `\newcommand*\thisgraphicspath{\{}}`

6097 `\renewcommand*\graphicspath[1]{\renewcommand*\thisgraphicspath{\#1}}`

`\DeclareGraphicsExtensions` `{\list}`

`\DeclareGraphicsRule` `{\langle}` `{\rangle}` `{\langle}` `{\rangle}`

6098 `\renewcommand*\DeclareGraphicsExtensions[1]{}`

6099 `\renewcommand*\DeclareGraphicsRule[4]{}`

64.2 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.

6100 `\renewcommand*\unitspace{}`

Used to store the user’s selected dimensions and HTML class.

The class defaults to “inlineimage” unless changed by a `class=xyx` option.

6101 `\newlength{\LWR@igwidth}`

6102 `\newlength{\LWR@igheight}`

```

6103 \newcommand*{\LWR@igwidthstyle}{%
6104 \newcommand*{\LWR@igheightstyle}{%
6105 \newcommand*{\LWR@igorigin}{%
6106 \newcommand*{\LWR@igangle}{%
6107 \newcommand*{\LWR@igxscale}{1}
6108 \newcommand*{\LWR@igyscale}{1}
6109 \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:

```

6110 \define@key{igraph}{width}{%
6111 \setlength{\LWR@igwidth}{#1}%
6112 \ifthenelse{\lengthtest{\LWR@igwidth > Opt}}{%
6113 {%

```

Default to use the converted fixed length given:

```

6114 \uselengthunit{PT}%
6115 \renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%

```

If ex or em dimensions were given, use those instead:

```

6116 \IfEndWith{#1}{ex}%
6117 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes ex
6118 {}% not ex
6119 \IfEndWith{#1}{em}%
6120 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes em
6121 {}% not em
6122 \IfEndWith{#1}{\}%
6123 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes percent
6124 {}% not percent
6125 \IfEndWith{#1}{px}%
6126 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
6127 {}% not px
6128 }{}% end of length > Opt
6129 }

```

If an optional height was given, set an HTML style:

```

6130 \define@key{igraph}{height}{%
6131 \setlength{\LWR@igheight}{#1}%
6132 \ifthenelse{\lengthtest{\LWR@igheight > Opt}}{%
6133 {%

```

Default to use the converted fixed length given:


```

6134 \uselengthunit{PT}%
6135 \renewcommand*{\LWR@igheightstyle}{%
6136 height:\rndprintlength{\LWR@igheight} %
6137 }%

```

If ex or em dimensions were given, use those instead:

```

6138 \IfEndWith{#1}{ex}%
6139 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
6140 {}% not ex
6141 \IfEndWith{#1}{em}%
6142 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
6143 {}% not em
6144 \IfEndWith{#1}{\}%
6145 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
6146 {}% not percent
6147 \IfEndWith{#1}{px}%
6148 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
6149 {}% not px
6150 }{}% end of length > 0pt
6151 }

```

Handle origin key:

```

6152 \define@key{igraph}{origin}{%
6153 \renewcommand*{\LWR@igorigin}{#1}%
6154 }

```

Handle angle key:

```

6155 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}

```

Handle class key:

```

6156 \define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}
6157

```

It appears that graphicx does not have separate keys for xscale and yscale. scale adjusts both at the same time.

```

6158 \define@key{igraph}{scale}{%
6159 \renewcommand*{\LWR@igxscale}{#1}%
6160 \renewcommand*{\LWR@igyyscale}{#1}}

```

Numerous ignored keys:

```

6161 \define@key{igraph}{bb}{}
6162 \define@key{igraph}{bblx}{}
6163 \define@key{igraph}{bbly}{}

```

```

6164 \define@key{igraph}{bburx}{}
6165 \define@key{igraph}{bbury}{}
6166 \define@key{igraph}{natwidth}{}
6167 \define@key{igraph}{natheight}{}
6168 \define@key{igraph}{hiresbb}{}
6169 \define@key{igraph}{viewport}{}
6170 \define@key{igraph}{trim}{}
6171 \define@key{igraph}{totalheight}{}
6172 \define@key{igraph}{keepaspectratio}{}
6173 \define@key{igraph}{clip}{}
6174 \define@key{igraph}{draft}{}
6175 \define@key{igraph}{type}{}
6176 \define@key{igraph}{ext}{}
6177 \define@key{igraph}{read}{}
6178 \define@key{igraph}{command}{}

```

`\LWR@rotstyle` $\{\langle prefix \rangle\} \{\langle degrees \rangle\}$

Prints the rotate style with the given prefix.

`prefix` is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:rotate` style.

```

6179 \newcommand*{\LWR@rotstyle}[2]{%
6180   #1transform:rotate(-#2deg);
6181 }

```

`\LWR@scalestyle` $\{\langle prefix \rangle\} \{\langle xscale \rangle\} \{\langle yscale \rangle\}$

Prints the scale style with the given prefix.

`prefix` is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:scale` style.

```

6182 \newcommand*{\LWR@scalestyle}[3]{%
6183   #1transform:scale(#2,#3);
6184 }

```

64.3 \includegraphics

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```

6185 \newbool{LWR@infloatrow}
6186 \boolfalse{LWR@infloatrow}

6187 \newcommand*{\LWR@imageextension}{}
6188 \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.

```
6189 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
6190 {%
```

Start the image tag on a new line, allow PDF output word wrap:

```
6191 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
6192 \ifthenelse{\cnttest{\value{\LWR@minipagedepth}}{=}{0}}{%
6193 \ifbool{\LWR@infloatrow}%
6194 {}
6195 {% not in a minipage or a floatrow:
6196 \setlength{\linewidth}{6in}%
6197 \setlength{\textwidth}{6in}%
6198 \setlength{\textheight}{9in}%
6199 }%
6200 }{}%
```

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.

```
6201 \edef\LWR@expgraphicsfilename{#4}
6202 \renewcommand*{\LWR@imageextension}{}%
6203 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.jpg}%
6204 {\renewcommand*{\LWR@imageextension}{.jpg}}{}%
6205 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.JPG}%
6206 {\renewcommand*{\LWR@imageextension}{.JPG}}{}%
6207 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.png}%
6208 {\renewcommand*{\LWR@imageextension}{.png}}{}%
6209 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.PNG}%
6210 {\renewcommand*{\LWR@imageextension}{.PNG}}{}%
6211 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.gif}%
6212 {\renewcommand*{\LWR@imageextension}{.gif}}{}%
6213 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.GIF}%
6214 {\renewcommand*{\LWR@imageextension}{.GIF}}{}%
6215 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svg}%
6216 {\renewcommand*{\LWR@imageextension}{.svg}}{}%
```

```

6216 {\renewcommand*{\LWR@imageextension}{.svg}}{}%
6217 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVG}%
6218 {\renewcommand*{\LWR@imageextension}{.SVG}}{}%
6219 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svgz}%
6220 {\renewcommand*{\LWR@imageextension}{.svgz}}{}%
6221 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVGZ}%
6222 {\renewcommand*{\LWR@imageextension}{.SVGZ}}{}%

```

Convert a PDF extension to SVG, leave the result in \LWR@strresult:

Must also \detokenize .pdf and .svg comparison strings.

```

6223 \StrSubstitute{\detokenize\expandafter{\LWR@expgraphicsfilename}}%
6224 {\detokenize{.pdf}}{\detokenize{.svg}}[\LWR@strresult]%
6225 %
6226 \StrSubstitute{\LWR@strresult}%
6227 {\detokenize{.PDF}}{\detokenize{.SVG}}[\LWR@strresult]%

```

For correct em sizing during the width and height conversions:

```

6228 \large%

```

Reset some defaults, possibly will be changed below if options were given:

```

6229 \setlength{\LWR@igwidth}{0pt}%
6230 \setlength{\LWR@igheight}{0pt}%
6231 \renewcommand*{\LWR@igwidthstyle}{}%
6232 \renewcommand*{\LWR@igheightstyle}{}%
6233 \renewcommand*{\LWR@igorigin}{}%
6234 \renewcommand*{\LWR@igangle}{}%
6235 \renewcommand*{\LWR@igxscale}{1}%
6236 \renewcommand*{\LWR@igyscale}{1}%
6237 \renewcommand*{\LWR@igclass}{inlineimage}%

```

If #3 is empty, only one optional argument was given, thus graphicx syntax:

```

6238 \IfValueTF{#3}{}{}%
6239 \IfValueTF{#2}%
6240 {\setkeys{igraph}{#2}}%
6241 {\setkeys{igraph}{}%
6242 }%

```

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.

```

6243 \href{\thisgraphicspath\LWR@strresult\LWR@imageextension}%

```

```

6244 {% start of href
6245 \LWR@htmltag{% start of image tags
6246 img src="\thisgraphicspath\LWR@strresult\LWR@imageextension" \LWR@orignewline
6247 \LWR@origtilde{} alt="\LWR@strresult" \LWR@orignewline

```

Only include a style tag if a width, height, angle, or scale was given:

```

6248 \ifthenelse{
6249 \NOT\equal{\LWR@igwidthstyle}{}} \OR
6250 \NOT\equal{\LWR@igheightstyle}{}} \OR
6251 \NOT\equal{\LWR@igorigin}{}} \OR
6252 \NOT\equal{\LWR@igangle}{}} \OR
6253 \NOT\equal{\LWR@igxscale}{1}} \OR
6254 \NOT\equal{\LWR@igyscale}{1}}
6255 }%
6256 {\LWR@origtilde{} style="%
6257 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{}}%
6258 {\LWR@igwidthstyle;}}}%
6259 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{}}%
6260 {\LWR@igheightstyle;}}}%
6261 \ifthenelse{\NOT\equal{\LWR@igorigin}{}}%
6262 {\LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{}}%
6263 \ifthenelse{\NOT\equal{\LWR@igangle}{}}%
6264 {%
6265 \LWR@rotstyle{-ms-}{\LWR@igangle}%
6266 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
6267 \LWR@rotstyle-}{\LWR@igangle%
6268 }}}%
6269 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}}\OR%
6270 \NOT\equal{\LWR@igyscale}{1}}%
6271 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}%
6272 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
6273 \LWR@scalestyle-}{\LWR@igxscale}{\LWR@igyscale}}}%
6274 " \LWR@orignewline}{}}%

```

Set the class:

```

6275 \LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%
6276 }% end of image tags
6277 }% end of href
6278 \endgroup

```

Return to small-sized output:

```

6279 \LWR@origscriptsize
6280 }

```

`\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.

```
6281 \renewcommand*{\includegraphics}
6282 {%
```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```
6283 \LWR@ensuredoingapar%
6284 \begin{group}%
6285 \LWR@includegraphicsb%
6286 }
```

```
6287 \end{warpHTML}
```

for PRINT output: For print output, accept and then discard the new class key:

```
6288 \begin{warpprint}
6289 \define@key{Gin}{class}{}
6290 \end{warpprint}
```

64.4 \rotatebox, \scalebox, \reflectbox

for HTML output: 6291 \begin{warpHTML}

\LWR@rotboxorigin Holds the origin key letters.

```
6292 \newcommand*{\LWR@rotboxorigin}{}%
```

\LWR@originname {\letter}

Given one L^AT_EX origin key value, translate into an HTML origin word:

```
6293 \newcommand*{\LWR@originname}[1]{%
6294 \ifthenelse{\equal{#1}{t}}{top}{}%
6295 \ifthenelse{\equal{#1}{b}}{bottom}{}%
6296 \ifthenelse{\equal{#1}{c}}{center}{}%
6297 \ifthenelse{\equal{#1}{l}}{left}{}%
6298 \ifthenelse{\equal{#1}{r}}{right}{}%
6299 }
```

`\LWR@originnames` $\{\langle letters \rangle\}$

Given one- or two-letter L^AT_EX origin key values, translate into HTML origin words:

```
6300 \newcommand*{\LWR@originnames}[1]{%
6301 \StrChar{#1}{1}[\LWR@strresult]%
6302 \LWR@originname{\LWR@strresult}
6303 \StrChar{#1}{2}[\LWR@strresult]%
6304 \LWR@originname{\LWR@strresult}
6305 }
```

Handle the origin key for `\rotatebox`:

```
6306 \define@key{krotbox}{origin}{%
6307 \renewcommand*{\LWR@rotboxorigin}{#1}%
6308 }
```

These keys are ignored:

```
6309 \define@key{krotbox}{x}{}
6310 \define@key{krotbox}{y}{}
6311 \define@key{krotbox}{units}{}

```

`\rotatebox` $[\langle keyval list \rangle] \{\langle angle \rangle\} \{\langle text \rangle\}$

Will `\let\rotatebox\LWR@rotatebox` at `\LWR@LwarpStart`, in case `\rotatebox` was over-written by a later package load.

```
6312 \NewDocumentCommand{\LWR@rotatebox}{0{ m +m}{%

```

Reset the origin to “none-given”:

```
6313 \renewcommand*{\LWR@rotboxorigin}{}

```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
6314 \setkeys{krotbox}{#1}%

```

Select `inline-block` so that HTML will transform this span:

```
6315 \LWR@htmltagc{span style="display: inline-block; %

```

If an origin was given, translate and print the origin information:

```
6316 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}{%
6317 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}}%

```

Print the rotation information:

```
6318 \LWR@rotstyle{-ms-}{#2} %
6319 \LWR@rotstyle{-webkit-}{#2} %
6320 \LWR@rotstyle{}{#2} %
6321 "{}\LWR@orignewline%
```

Print the text to be rotated:

```
6322 \begin{LWR@nestspan}%
6323 #3%
```

Close the span:

```
6324 \LWR@htmltagc{/span}%
6325 \end{LWR@nestspan}%
6326 }
```

`\scalebox` $\{\langle h\text{-scale}\rangle\} [\langle v\text{-scale}\rangle] \{\langle text\rangle\}$

Will `\let\scalebox\LWR@scalebox` at `\LWR@LwarpStart`, in case `\scalebox` was over-written by a later package load.

```
6327 \NewDocumentCommand{\LWR@scalebox}{m o m}{%
```

Select `inline-block` so that HTML will transform this span:

```
6328 \LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
6329 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6330 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6331 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6332 "{}}%
```

Print the text to be scaled:

```
6333 \begin{LWR@nestspan}%
6334 #3%
```

Close the span:

```
6335 \LWR@htmltagc{/span}%
6336 \end{LWR@nestspan}%
6337 }
```

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

Will `\let\reflectbox\LWR@reflectbox` at `\LWR@LwarpStart`, in case `\reflectbox` was over-written by a later package load.

```
6338 \newcommand{\LWR@reflectbox}[1]{\LWR@scalebox{-1}[1]{#1}}
6339 \end{warpHTML}
```

64.5 Null functions

These functions are not supported by `lwarp`'s HTML conversion.

for HTML output: 6340 `\begin{warpHTML}`

```
\resizebox {<h-length>} {<v-length>} {<text>}
```

Simply prints its text argument.

```
6341 \renewcommand{\resizebox}[3]{#3}
6342 \end{warpHTML}
```

65 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 `lwarp` loads it last. If the user's document preamble uses `cleveref` options, or functions such as `\crefname`, then `cleveref` may be loaded in the user's preamble near the end, and `lwarp`'s additional loading of `cleveref` will have no effect.

Table 9 on 265 shows the data structure of the label/reference system as revised by `lwarp` and `cleveref`.

A few patches allow `cleveref` to work as-is:

for HTML output: 6343 `\begin{warpHTML}`

```
\AtEndPreamble forces cleveref to be loaded last:
```

The following patches are applied after `cleveref` has loaded, and after `\AtBeginDocument`:

```

6344 \AfterEndPreamble{

\@@setcref  {\<kindofref\>} {\<label\>}

6345 \renewcommand*{\@@setcref}[2]{\#1{\ref{\#2}}{\}\{\}}

\@@setcrefrange  {\<text\>} {\<label\>} {\<label\>}

6346 \renewcommand{\@@setcrefrange}[3]{%
6347 \#1{\ref{\#2}}{\ref{\#3}}{\}\{\}\{\}}

\cpagerefFor  Redefinable word between “page(s)” and the page numbers.

6348 \newcommand*{\cpagerefFor}{for}

\@@setcpageref  {\<typeofref\>} {\<label\>}, where typeofref is “page” or “pages”

6349 \renewcommand*{\@@setcpageref}[2]{%
6350 \#1{\cpagerefFor\ \cref{\#2}}{\}\{\}}%
6351 }

6352 \renewcommand{\@@setcpagerefrange}[3]{%
6353 \#1{\cpagerefFor\ \cref{\#2}}{\cref{\#3}}{\}\{\}\{\}}
6354 }% AfterEndPreamble

```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

```

6355 \let\LWR@origlabel\label
6356 \let\label\LWR@newlabel
6357 \let\LWR@origref\ref
6358 \let\ref\LWR@newref% \end{ syntax highlighting
6359 \let\LWR@origpageref\pageref
6360 \let\pageref\LWR@newpageref
6361
6362
6363
6364 \end{warpHTML}

```

66 Picture

Env `picture` The `picture` environment is enclosed inside a `\lateximage`.

for HTML output: 6365 `\begin{warpHTML}`

Env `picture`


```
6366 \BeforeBeginEnvironment{picture}{%
6367 \lateximage%
6368 \let\makebox\LWR@origmakebox%
6369 }
6370
6371 \AfterEndEnvironment{picture}{\endlateximage}

6372 \end{wrapHTML}
```

67 Boxes and Minipages

A CSS flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

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.

for HTML output: `6373 \begin{wrapHTML}`

67.1 Counters and lengths

Ctr `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
6374 \newcounter{LWR@minipagedepth}
6375 \setcounter{LWR@minipagedepth}{0}
```

Len `\WR@minipagewidth` Used to convert the width into printable units.

```
6376 \newlength{\LWR@minipagewidth}
```

Len `\WR@minipageheight` Used to convert the height into printable units.

```
6377 \newlength{\LWR@minipageheight}
```

Remember the original definitions:

```
6378 \let\LWR@origminipage\minipage
6379 \let\LWR@origendminipage\endminipage
```

67.2 Footnote handling

Also see section [41](#) for other forms of footnotes.

67.3 Minipage handling

`\LWR@endminipage` Used to close a minipage.

Copied the L^AT_EX definition and modified to create a `mpfootnotes` div class:

```
6380 \def\LWR@endminipage{%
6381   \par
6382   \unskip
6383   \ifvoid\@mpfootins\else
6384     \vskip\skip\@mpfootins
6385     \normalcolor
6386 \LWR@htmldivclass{mpfootnotes}
6387 \LWR@origmedskip
6388   \unvbox\@mpfootins
6389 \LWR@htmldivclassend{mpfootnotes}
6390   \fi
6391   \@minipagefalse
6392   \color@endgroup
6393 \egroup
6394 \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.

```
6395 \newcommand*{\LWR@subminipage}{%
6396 \LWR@stoppars
6397 \LWR@origminipage{6in}
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
6398 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
6399 \LWR@startpars%
6400 }
```

`\LWR@endsubminipage` Closes the subminipage.

```
6401 \newcommand*{\LWR@endsubminipage}{%
6402 \LWR@stoppars%
6403 \LWR@endminipage% The following empty line is required:
6404
6405 }
```

Bool `LWR@minipagefullwidth` Should the next minipage have no HTML width?

```
6406 \newbool{LWR@minipagefullwidth}
6407 \boolfalse{LWR@minipagefullwidth}
```

`\minipagefullwidth` Requests that the next minipage have no width tag in HTML:

for HTML output:

```
6408 \newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}
6409 \end{warpHTML}
```

for PRINT output:

```
6410 \begin{warpprint}
6411 \newcommand*{\minipagefullwidth}{}
6412 \end{warpprint}
```

for HTML output:

```
6413 \begin{warpHTML}
```

Bool `LWR@minipagethispar` Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
6414 \newbool{LWR@minipagethispar}
6415 \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.

```
6416 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}
6417 {%
```

Pre-compute the given width and height:

Reset the text area if are starting the outer-most minipage:

```
6418 \LWR@traceinfo{starting minipage of width #4}%
6419 \uselengthunit{in}%
6420 \setlength{\LWR@minipagewidth}{#4}%
6421 \ifthenelse{\cnttest{\value{\LWR@minipagedepth}}{=}{0}}{%
6422 \addtolength{\LWR@minipagewidth}{3em}% room for frames
6423 \setlength{\linewidth}{6in}%
6424 \setlength{\textwidth}{6in}%
6425 \setlength{\textheight}{9in}%
6426 }{}%
6427 \LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}
6428 \addtocounter{\LWR@minipagedepth}{1}%
6429 \setlength{\LWR@minipageheight}{\textheight}% default unless specified
6430 \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.

```
6431 \LWR@stoppars%
```

Create the `<div>` tag with optional alignment style:

```
6432 \LWR@traceinfo{minipage: creating div class}%
6433 \LWR@orignewpage%
6434 \LWR@htmltag{div class="minipage" style="%
6435 \ifthenelse{\equal{#1}{t}}{vertical-align: bottom ; }{}%
6436 \ifthenelse{\equal{#1}{c}}{vertical-align: middle ; }{}%
6437 \ifthenelse{\equal{#1}{b}}{vertical-align: top ; }{}%
6438 \ifthenelse{\equal{#3}{t}}{justify-content: flex-start ; }{}%
6439 \ifthenelse{\equal{#3}{c}}{justify-content: center ; }{}%
6440 \ifthenelse{\equal{#3}{b}}{justify-content: flex-end ; }{}%
6441 \ifthenelse{\equal{#3}{s}}{justify-content: space-between ; }{}%
```

Print the width and optional height styles:

```

6442 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
6443 \uselengthunit{PT}%
6444 \ifbool{\LWR@minipagefullwidth}%
6445 {\boolfalse{\LWR@minipagefullwidth}}%
6446 {%
6447 \ifthenelse{\lengthtest{#4}=\linewidth}%
6448 {}%
6449 {width:\rndprintlength{\LWR@minipagewidth} ; }%
6450 }%
6451 \LWR@traceinfo{minipage: about to print the height}%
6452 \IfValueTF{#2}{height:\rndprintlength{\LWR@minipageheight} ; }{}%
6453 "{}%

```

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.

```

6454
6455 \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.

```

6456 \setlength{\linewidth}{#4}% the original width
6457 \setlength{\textwidth}{6in}%
6458 \setlength{\textheight}{9in}%

```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```

6459 \LWR@origraggedright%

```

Resume paragraph tag handling for the contents of the minipage:

```

6460 \LWR@startpars%
6461 \LWR@traceinfo{minipage: finished starting the minipage}%
6462 }

```

End the environment with L^AT_EX processing and closing tag:

```

6463 {%
6464 \LWR@stoppars%
6465 \LWR@endminipage% The following empty line is required:
6466
6467 \LWR@htmldivclassend{minipage}%
6468 \vspace{1\baselineskip}% required for subcaption
6469 \addtocounter{\LWR@minipagedepth}{-1}%

```

```
6470 \LWR@startpars%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
6471 \global\booltrue{LWR@minipagethispar}%
6472 }
```

67.4 Parbox, makebox, framebox, fbox, raisebox

`\parbox` [*<pos>*] [*<height>*] [*<inner-pos>*] {*<width>*} {*<text>*}

A parbox uses the minipage code:

```
6473 \RenewDocumentCommand{\parbox}{0{t} o 0{t} m +m}
6474 {
6475 \LWR@traceinfo{parbox of width #4}%
6476 \begin{minipage}[#1][#2][#3]{#4}
6477 #5
6478 \end{minipage}
6479 }
```

`\makebox` [*<width>*] [*<pos>*] {*<text>*}

Width and position are ignored.

```
6480 \let\LWR@origmakebox\makebox
6481
6482 \RenewDocumentCommand{\makebox}{o o m}{%
6483 \mbox{#3}
6484 }
```

`\framebox` [*<width>*] [*<pos>*] {*<text>*}

Width and position are ignored.

```
6485 \RenewDocumentCommand{\framebox}{o o m}{%
6486 \fbox{#3}
6487 }
```

`\fbox` {*<text>*}

```
6488 \let\LWR@origfbox\fbox
6489 %
6490 \renewcommand*\fbox{[1]{%
```



```
6491 \InlineClass{framebox}{#1}%
6492 }
```

```
\raisebox {<raiselen>} [<height>] [<depth>] {<text>}
```

```
6493 \RenewDocumentCommand{\raisebox}{m o o m}{%
6494 #4%
6495 }
```

```
6496 \end{warpHTML}
```

68 Direct formatting

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

For high-level block and inline custom CSS classes, see section [35.7](#).

for HTML output: 6497 \begin{warpHTML}

```
\emph {<text>}
```

```
6498 \renewcommand{\emph}[1]{\LWR@htmlspan{em}{#1}}
```

```
\textmd {<text>}
```

```
6499 \renewcommand{\textmd}[1]{\LWR@htmlspan{textmd}{#1}}
```

```
\textbf {<text>}
```

```
6500 \renewcommand{\textbf}[1]{\LWR@htmlspan{b}{#1}}
```

```
\textrm {<text>}
```

```
6501 \renewcommand{\textrm}[1]{\InlineClass{textrm}{#1}}
```

```
\textsf {<text>}
```

```
6502 \renewcommand{\textsf}[1]{\InlineClass{textsf}{#1}}
```

```
\texttt {<text>}
```

```
6503 \renewcommand{\texttt}[1]{\LWR@htmlspan{kbd}{#1}}
```

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

6504 `\renewcommand{\textup}[1]{\InlineClass{textup}{#1}}`

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

6505 `\renewcommand{\textit}[1]{\LWR@htmlspan{i}{#1}}`

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

6506 `\renewcommand{\textsc}[1]{\InlineClass{textsc}{#1}}`

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

6507 `\renewcommand{\textnormal}[1]{\textmd{\textrm{\textup{#1}}}}`

`\mdseries`

6508 `\renewcommand*{\mdseries}{}`

`\bfseries`

6509 `\renewcommand*{\bfseries}{}`

`\rmfamily`

6510 `\renewcommand*{\rmfamily}{}`

`\sffamily`

6511 `\renewcommand*{\sffamily}{}`

`\ttfamily`

6512 `\renewcommand*{\ttfamily}{}`

`\upshape`

6513 `\renewcommand*{\upshape}{}`

`\itshape`

6514 `\renewcommand*{\itshape}{}`

`\scshape`

```
6515 \renewcommand*{\scshape}{}

```

`\scshape`

```
6516 \renewcommand*{\normalfont}{}

```

`\sp` $\langle text \rangle$

For siunitx. Must work in math mode.

```
6517 \renewcommand{\sp}[1]{\text{<sup>#1</sup>}}

```

`\sb` $\langle text \rangle$

For siunitx. Must work in math mode.

```
6518 \renewcommand{\sb}[1]{\text{<sub>#1</sub>}}

```

`\textsuperscript` $\langle text \rangle$

```
6519 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}

```

`\textsubscript` $\langle text \rangle$

```
6520 \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.

```
6521 \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.

```
6522 \AtBeginDocument{\let\fup\textsuperscript}

```

`\hfill`

```
6523 \renewcommand*{\hfill}{\quad}
```

`\hrulefill`

```
6524 \renewcommand*{\hrulefill}{\rule{1in}{1pt}}
```

`\dotfill`

```
6525 \renewcommand*{\dotfill}{\dots}
```

```
6526 \end{warpHTML}
```

69 Skips, spaces, font sizes

for HTML output: 6527 `\begin{warpHTML}`

`\`, must be redefined after `\RequirePackage{printlen}`

```
6528 \let\LWR@origcomma\,
6529 \let\LWR@origtilde~
6530 \let\LWR@origenskip\enskip
6531 \let\LWR@origquad\quad
6532 \let\LWR@origqqquad\qqquad
6533 \let\LWR@orighspace\hspace
6534 \let\LWR@origrule\rule
6535 \let\LWR@origmedskip\medskip
```

Direct-formatting space commands become HTML entities:

```
6536 \renewcommand*{\,}{\HTMLUnicode{202f}} % HTML thin non-breakable space
```

```
6537 \renewcommand*{~}{\HTMLentity{nbsp}}
```

```
6538 \renewcommand*{\textellipsis}{\HTMLUnicode{2026}}
```

Direct-formatting font sizes are ignored:

```
6539 \let\LWR@orignormalsize\normalsize
6540 \let\LWR@origsmall\small
6541 \let\LWR@origfootnotesize\footnotesize
6542 \let\LWR@origscriptsize\scriptsize
6543 \let\LWR@origtiny\tiny
6544 \let\LWR@origlarge\large
```

```

6545 \let\LWR@origLarge\Large
6546 \let\LWR@origLARGE\LARGE
6547 \let\LWR@orighuge\huge
6548 \let\LWR@origHuge\Huge
6549 \renewcommand*\normalsize{}
6550 \renewcommand*\small{}
6551 \renewcommand*\footnotesize{}
6552 \renewcommand*\scriptsize{}
6553 \renewcommand*\tiny{}
6554 \renewcommand*\large{}
6555 \renewcommand*\Large{}
6556 \renewcommand*\LARGE{}
6557 \renewcommand*\huge{}
6558 \renewcommand*\Huge{}
6559
6560 \renewcommand*\onecolumn{}
6561
6562 \renewcommand*\twocolumn[1][]{
6563
6564 #1
6565
6566 }

```

`\newline` Uses HTML `
` tag

```

6567 \newcommand*\LWR@newlinebr{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
6568 \let\newline\LWR@newlinebr

```

`\\` Redefined to `\LWR@endoffline` or `\LWR@tabularendoffline`.

`\LWR@endoffline` * `[<len>]`

`\\` is assigned to `\LWR@endoffline` at `\LWR@LwarpStart`.

Inside `tabular`, `\\` is temporarily changed to `\LWR@tabularendoffline`.

```

6569 \let\LWR@origendoffline\\
6570 \NewDocumentCommand{\LWR@endoffline}{s o}
6571 {%
6572 \newline%
6573 }

```

`\LWR@minipagestartpars` Minipages are often placed side-by-side inside figures, with a bit of horizontal space to separate them. Since HTML does not allow a `<div>` to be inside a `p`, paragraphs must be turned off during the generation of the minipage, then turned on after the minipage is complete. When this occurs between side-by-side minipages, `lwarp` correctly suppresses the paragraph tags between the minipages, unless some other

text is between the minipages. Such text forms its own paragraph, resulting in text after a minipage to be on its own line. Since people often place small horizontal space between minipages, it is desirable to maintain this space if possible. `lwarp` tries to do this by remembering that a minipage has been seen, in which case paragraph tags are suppressed around `\hspace`, `\enskip`, `\quad`, and `\qquad` until the end of the paragraph, when the closing `p` tag is created.

When a minipage is seen, the boolean `LWR@minipagethispar` is set, telling the following horizontal whitespace commands to try to suppress their surrounding paragraph tags. `LWR@minipagethispar` is cleared at the next end of paragraph, when the HTML paragraph closing tag is generated.

Placed just before `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
6574 \newcommand*{\LWR@minipagestartpars}{%
6575 \ifbool{LWR@minipagethispar}%
6576 {%
6577 \LWR@startpars%
6578 }{}%
6579 }
```

`\LWR@minipagestoppars` Placed just after `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
6580 \newcommand*{\LWR@minipagestoppars}{%
6581 \ifbool{LWR@minipagethispar}%
6582 {%
6583 \LWR@stoppars%
6584 }{}%
6585 }
```

`\quad` Handles special minipage & horizontal space interactions.

```
6586 \renewcommand*{\quad}{%
6587 \LWR@minipagestoppars%
6588 \HTMLUnicode{2001}%
6589 \LWR@minipagestartpars%
6590 }
```

`\qquad` Handles special minipage & horizontal space interactions.

```
6591 \renewcommand*{\qquad}{\quad\quad}
```

`\enskip` Handles special minipage & horizontal space interactions.

```
6592 \renewcommand*{\enskip}{%
6593 \LWR@minipagestoppars%
6594 \HTMLUnicode{2000}%
```

```
6595 \LWR@minipagestartpars%
6596 }
```

```

Len  \WR@tempwidth  Used to compute span width, height, raise for \hspace and \rule:
Len  \WR@tempheight 6597 \newlength{\LWR@tempwidth}
Len  \WR@tempraise  6598 \newlength{\LWR@tempheight}
                        6599 \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`.

```
6600 \NewDocumentCommand{\LWR@hspace}{s m}{%
6601 \setlength{\LWR@tempwidth}{#2}%
```

If `\fill`, change to `\qquad`:

```
6602 \ifnum\gluestretchorder\LWR@tempwidth>0%
6603 \setlength{\LWR@tempwidth}{2em}%
6604 \fi%
```

Only if the width is not zero:

```
6605 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{Opt}}{ }{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
6606 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
6607 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{.16667em}}
6608 {%
6609 \HTMLUnicode{2009}% thin breakable space
6610 }%
```

Print the span with the converted width. Not rounded.

```
6611 {%
6612 \uselengthunit{PT}%
6613 \LWR@htmltagc{%
6614 span style="width:\printlength{\LWR@tempwidth}; display:inline-block;"%
```

```

6615 }%
6616 \LWR@htmltagc{/span}%
6617 }%

```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

6618 \LWR@minipagestartpars%
6619 }%
6620 }

```

`\hspace` * $\langle length \rangle$

Handles special minipage & horizontal space interactions.

```

6621 \let\hspace\LWR@hspace

```

`\linebreak` [$\langle num \rangle$] Inserts an HTML `br` tag.

```

6622 \renewcommand*{\linebreak}[1] [] {\newline}

```

`\nolinebreak` [$\langle num \rangle$]

```

6623 \renewcommand*{\nolinebreak}[1] [] {}

```

`\pagebreak` [$\langle num \rangle$] Starts a new paragraph.

```

6624 \renewcommand*{\pagebreak}[1] [] {
6625
6626 }

```

`\nopagebreak` [$\langle num \rangle$]

```

6627 \renewcommand*{\nopagebreak}[1] [] {}

```

`\enlargethispage` * $\langle len \rangle$

```

6628 \RenewDocumentCommand{\enlargethispage}{s m}{}

```

`\LWR@rule` [$\langle raise \rangle$] $\langle width \rangle$ $\langle height \rangle$

Handles special minipage & horizontal space interactions.

Creates a span of a given width and height. Ignores the optional star.

`\fill` is zero-width, so `\hspace{\fill}` is ignored.

```
6629 \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:

```
6630 \setlength{\LWR@tempwidth}{#2}%
```

If it's zero-width then skip the entire rule:

```
6631 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}{
6632 {}}% zero- width
6633 {}% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
6634 \ifthenelse{\lengthtest{\LWR@tempwidth>0pt}\AND%
6635 \lengthtest{\LWR@tempwidth<1pt}}{%
6636 {\setlength{\LWR@tempwidth}{1pt}}{}}%
```

Likewise with height:

```
6637 \setlength{\LWR@tempheight}{#3}%
6638 \ifthenelse{\lengthtest{\LWR@tempheight>0pt}\AND%
6639 \lengthtest{\LWR@tempheight<1pt}}{%
6640 {\setlength{\LWR@tempheight}{1pt}}{}}%
```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
6641 \LWR@minipagestoppars%
```

Print the span with the converted width and height. The width and height are NOT rounded, since a height of less than 1pt is quite common in L^AT_EX code.

```
6642 \uselengthunit{PT}%
6643 \LWR@htmltagc{%
6644 span
6645 style=" %
```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```
6646 background:\LWR@currenttextcolor; %
```

The width and height are printed, converted to PT:

```
6647 width:\printlength{\LWR@tempwidth}; %
6648 height:\printlength{\LWR@tempheight}; %
```

The raise height is converted to a CSS transform. The *2 raise multiplier is to approximately match HTML output's X height. Conversion to a \LaTeX length allows a typical \LaTeX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a \LaTeX length limits the allowable syntax. To do: A superior method would compute a ratio of \LaTeX ex height, then print that to HTML with an ex unit.

```
6649 \IfValueTF{#1}%
6650 {%
6651 \setlength{\LWR@tempraise}{Opt-#1}%
6652 \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
6653 \LWR@orignewline%
6654 -ms-transform: translate(Opt,\printlength{\LWR@tempraise}); %
6655 \LWR@orignewline%
6656 -webkit-transform: translate(Opt,\printlength{\LWR@tempraise}); %
6657 \LWR@orignewline%
6658 transform: translate(Opt,\printlength{\LWR@tempraise}); %
6659 \LWR@orignewline%
6660 }{}%
```

Display inline-block to place the span inline with the text:

```
6661 display:inline-block; "%
6662 }%
6663 \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
6664 \LWR@minipagestartpars%
6665 }% non-zero width
6666 }
```

```
\rule [\langle raise \rangle] {\langle width \rangle} {\langle height \rangle}
```

Handles special minipage & horizontal space interactions.

```
6667 \let\rule\LWR@rule
```

```
6668 \end{warpHTML}
```

70 `\phantomsection`

for HTML output: 6669 `\begin{warpHTML}`

`\phantomsection` Emulate the hyperref `\phantomsection` command, often used to insert the bibliography into table of contents:

```
6670 \newcommand*{\phantomsection}{\section*{}}
```

```
6671 \end{warpHTML}
```

71 `\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>

71.1 HTML logos

for HTML output: 6672 `\begin{warpHTML}`

`\TeX` $\mathrm{T}_{\mathrm{E}}\mathrm{X}$

`latexlogo` is a CSS class used to properly typeset the E and A in $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and friends.

`latexlogofont` is a CSS class used to select the font for the rest of the logo in $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{LuaT}_{\mathrm{E}}\mathrm{X}$, $\mathrm{ConT}_{\mathrm{E}}\mathrm{Xt}$, etc.

```
6673 \newcommand*{\LWR@TeX}
```

```
6674 {\InlineClass{latexlogofont}%
```

```
6675 {\InlineClass{latexlogo}{T\textsubscript{e}X}}}
```

`\LaTeX` $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X} 2_{\varepsilon}$

`\LaTeXe`

```
6676 \newcommand*{\LWR@LaTeX}
```

```
6677 {\InlineClass{latexlogofont}%
```

```

6678 {\InlineClass{latexlogo}%
6679 {L\textsuperscript{a}T\textsubscript{e}X}}
6680
6681 \renewcommand*{\LaTeXe}
6682 {\LaTeX\InlineClass{latexlogofont}%
6683 {\,2\textsubscript{\textit{\HTMLUnicode{3B5}}}}}

```

`\LuaTeX` $\text{LuaT}_{\text{E}}\text{X}$, $\text{LuaL}^{\text{A}}\text{T}_{\text{E}}\text{X}$
`\LuaLaTeX`

```

6684 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}\Lua\TeX}
6685 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}\Lua\LaTeX}

```

`\XeTeX` $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$, $\text{X}_{\text{E}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
`\XeLaTeX`

`xetexlogo` is a CSS class which aligns the backwards E in $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$ and spaces $\text{T}_{\text{E}}\text{X}$ appropriately.

`xelatexlogo` is a CSS class which aligns the backwards E in $\text{X}_{\text{E}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ and spaces $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ appropriately.

```

6686 \newcommand*{\Xe}
6687 {X\textsubscript{\HTMLUnicode{18e}}}
6688 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}\Xe\TeX}
6689 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}\Xe\LaTeX}

```

`\ConTeXt` $\text{ConT}_{\text{E}}\text{Xt}$

```

6690 \newcommand*{\LWR@ConTeXt}
6691 {\InlineClass{latexlogofont}\Con\TeX{}}%
6692 \InlineClass{latexlogofont}\t}

```

`\BibTeX` $\text{BIBT}_{\text{E}}\text{X}$, *MakeIndex*
`\MakeIndex`

```

6693 \providecommand*{\BibTeX}
6694 {\InlineClass{latexlogofont}\B\textsc{ib}\TeX}
6695
6696 \newcommand*{\MakeIndex}
6697 {\InlineClass{latexlogofont}\textit{MakeIndex}}

```

`\AmS` $\mathcal{A}\mathcal{M}\mathcal{S}$

`amslogo` is a CSS class used for the $\mathcal{A}\mathcal{M}\mathcal{S}$ logo.

```

6698 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}}
6699 {\InlineClass{amslogo}\textit{A\textsubscript{M}S}}}

```

`\MiKTeX` `MiKTeX`

```
6700 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}
```

`\LyX` `LyX`

`lyxlogo` is a CSS class used for the `LyX` logo.

```
6701 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
```

```
6702 \end{warpHTML}
```

71.2 Print logos

for PRINT output:

```
6703 \begin{warpprint}
6704 \newcommand*{\XeTeXrevE}
6705   {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
6706 \providecommand*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
6707 \providecommand*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
6708 \providecommand*{\AmS}{\%}
6709 \leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex\%}
6710 \hbox{$\mathcal M\kern-.2em\mathcal S$}
6711 \newcommand*{\LyX}{\textsf{LyX}}
6712 \providecommand*{\LuaTeX}{\mbox{Lua\TeX}}
6713 \providecommand*{\LuaLaTeX}{\mbox{Lua\LaTeX}}
6714 \providecommand*{\BibTeX}{\mbox{B\textsc{ib}\TeX}}
6715 \providecommand*{\MakeIndex}{\mbox{\textit{MakeIndex}}}
6716 \providecommand*{\ConTeXt}{\mbox{Con\TeX{t}}}
6717 \providecommand*{\MiKTeX}{\mbox{MiK\TeX}}
6718 \end{warpprint}
```

72 `\AtBeginDocument`, `\AtEndDocument`

for HTML output:

```
6719 \begin{warpHTML}
```

`\LWR@LwarpStart` Automatically sets up the HTML-related actions for the start and end of the document.

`\LWR@LwarpEnd`

```
6720 \AfterEndPreamble{\LWR@LwarpStart}
```

```
6721 \AtEndDocument{\LWR@LwarpEnd}
```

```
6722 \end{warpHTML}
```

73 Trademarks

- \TeX is a trademark of American Mathematical Society.
- Adobe® and Adobe Framemaker® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.
- Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Mac OS® is a trademark of Apple Inc.
- MadCap Flare™ is the property of MadCap Software, Inc.
- MathJax is copyright 2009 and later. The MathJax Consortium is a joint venture of the American Mathematical Society (AMS) and the Society for Industrial and Applied Mathematics (SIAM) to advance mathematical and scientific content on the web.
- Microsoft®, Encarta, MSN, and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- UNIX® is a registered trademark of The Open Group.

Package 2

lwarp-abstract.sty

74 Abstract

(Based on original code by PETER WILSON.)

`Pkg abstract` abstract is supported and patched by lwarp.

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

for HTML output: Accept all options for lwarp-abstract:

```

1 \LWR@ProvidesPackagePass{abstract}

2 \AtBeginDocument{
3 \BeforeBeginEnvironment{abstract}{
4 \LWR@forcenewpage
5 \BlockClass{abstract}
6 }
7 \AfterEndEnvironment{abstract}{\endBlockClass}
8 }
9
10 \renewcommand{\@bsrunintitle}{%
11 \hspace*\abstitleskip}%
12 {\abstractnamefont%
13 \InlineClass{abstractrunintitle}{\abstractname}%
14 \@bslabeldelim}%
15 }
16
17 \if@titlepage
18 \renewenvironment{abstract}{%
19 % \titlepage
20 \null\vfil
21 \@beginparpenalty\@lowpenalty
22 \if@bsrunin
23 \else
24 \if@bsstyle
25 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
26 \else
27 \ifnumber@bs
28 \num@bs

```

```

29         \else
30         \begin{\absnamepos}%
31 \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
32         \endparpenalty\@M
33         \end\absnamepos%
34 %%         \vspace{\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 3

lwarp-afterpage.sty

75 Afterpage

Pkg afterpage Not used.

for **HTML output**: Discard all options for lwarp-afterpage:

```
1 \LWR@ProvidesPackageDrop{afterpage}
```

```
2 \newcommand{\afterpage}[1]{#1}
```

Package 4

lwarp-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 5

lwarp-alltt.sty

77 Alltt

Pkg alltt alltt is patched for use by lwarp.

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 6

lwarp-amsthm.sty

78 AMSthm

(Based on original code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)

Pkg amsthm amsthm is patched for use by lwarp.

CSS styling of theorems and proofs:

Theorem: <div> of class amsthmbody<theoremstyle>
Theorem Name: of class amsthmname<theoremstyle>
Theorem Number: of class amsthmnumber<theoremstyle>
Theorem Note: of class amsthmnote<theoremstyle>
Proof: <div> of class amsthmproof
Proof Name: of class amsthmproofname
 where <theoremstyle> is plain, definition, etc.

for HTML output: 1 \LWR@ProvidesPackagePass{amsthm}

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
3 \renewcommand{\theoremstyle}[1]{%
4   \@ifundefined{th@#1}{%
5     \PackageWarning{amsthm}{Unknown theoremstyle '#1'}%
6     \thm@style{plain}%
7   \renewcommand{\LWR@newtheoremstyle}{plain}% new
8   }{%
9     \thm@style{#1}%
10  \renewcommand{\LWR@newtheoremstyle}{#1}% new
11  }%
12 }
```

Patched to remember the style for this theorem type:

```

13 \def\@xnthm#1#2{%
14   \csedef{LWR@thmstyle#2}{\LWR@newtheoremstyle}% new
15   \let\@tempa\relax
16   \@xp\@ifdefinable\csname #2\endcsname{%
17     \global\@xp\let\csname end#2\endcsname\@endtheorem
18     \ifx *#1% unnumbered, need to get one more mandatory arg
19       \edef\@tempa##1{%
20         \gdef\@xp\@nx\csname#2\endcsname{%
21           \@nx\@thm{\@xp\@nx\csname th@\the\thm@style\endcsname}%
22           }\{##1\}}%
23       \else % numbered theorem, need to check for optional arg
24         \def\@tempa{\@oparg{\@ynthm{#2}}{}}%
25       \fi
26 \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#2}}}% new
27 }%
28 \@tempa
29 }

```

Patched to enclose with CSS:

```

30 \newcommand{\LWR@haveamsthmname}{
31 \renewcommand{\thmname}[1]{\InlineClass{amsthmname\LWR@thisthmstyle}{##1}}
32 }
33
34 \newcommand{\LWR@haveamsthmnumber}{
35 \renewcommand{\thmnumber}[1]{\InlineClass{amsthmnumber\LWR@thisthmstyle}{##1}}
36 }
37
38 \newcommand{\LWR@haveamsthmnote}{
39 \renewcommand{\thmnote}[1]{\InlineClass{amsthmnote\LWR@thisthmstyle}{##1}}
40 }
41
42 \LWR@haveamsthmname
43 \LWR@haveamsthmnumber
44 \LWR@haveamsthmnote

```

Patches for CSS:

```

45 \def\@begintheorem#1#2[#3]{%
46 \LWR@forcenewpage% new
47 \BlockClass{amsthmbody\LWR@thisthmstyle}% new
48 \deferred@thm@head{
49 \the\thm@headfont \thm@indent
50   \@ifempty{#1}{\let\thmname\@gobble}{\LWR@haveamsthmname}% new
51   \@ifempty{#2}{\let\thmnumber\@gobble}{\LWR@haveamsthmnumber}% new
52   \@ifempty{#3}{\let\thmnote\@gobble}{\LWR@haveamsthmnote}% new
53   \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
54   \the\thm@headpunct~
55   \thmheadnl % possibly a newline.
56   \hskip\thm@headsep

```

```

57 }%
58 \ignorespaces}

```

Patched for CSS:

```

59 \def\@endtheorem{\endBlockClass\endtrivlist\@endpefalse }

```

Proof QED symbol:

```

60 \AtBeginDocument{
61 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
62 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
63 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
64 }

```

Patched for CSS:

```

65 \renewenvironment{proof}[1][\proofname]{\par
66 \LWR@forcenewpage% new
67 \BlockClass{amsthmproof}% new
68 \pushQED{\qed}%
69 \normalfont \topsep6\p@\@plus6\p@\relax
70 \trivlist
71 \item[\hskip\labelsep
72 \InlineClass{amsthmproofname}{#1\@addpunct{.}}]\ignorespaces% changes
73 }{%
74 \InlineClass{theoremendmark}{\popQED}\endtrivlist%
75 \endBlockClass% new
76 \@endpefalse
77 }

```

Package 7

lwarp-bookmark.sty

79 Bookmark

`\Pkg bookmark` bookmark is emulated during HTML output, and the bookmark package is ignored.

for HTML output: Discard all options for lwarp-bookmark:

```

1 \LWR@ProvidesPackageDrop{bookmark}
2 \newcommand*{\bookmarksetup}[1]{ }

```

```
3 \newcommand*{\bookmarksetupnext}[1]{}  
4 \newcommand*{\bookmark}[2] [] {}  
5 \newcommand*{\bookmarkdefinestyle}[2] {}  
6 \newcommand*{\bookmarkget}[1] {}  
7 \newcommand{\BookmarkAtEnd}[1] {}
```

Package 8

lwarp-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]{}%
```

Package 9

lwarp-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 10

lwarp-changepage.sty

82 Changepage

Pkg changepage changepage is ignored.

for HTML output: Discard all options for lwarp-changepage:

```
1 \LWR@ProvidesPackageDrop{changepage}

2 \newif\ifoddpage
3 \DeclareRobustCommand{\checkoddpage}{\oddpagetrue}
4 \DeclareRobustCommand{\changetext}[5]{\}
5 \DeclareRobustCommand{\changeage}[9]{\}
6 \newenvironment{adjustwidth}[2]{\}\}
7 \newenvironment{adjustwidth*}[2]{\}\}
```

Package 11

lwarp-cutwin.sty

83 Cutwin

Pkg cutwin Emulated.

for HTML output: Discard all options for lwarp-cutwin:

```

1 \LWR@ProvidesPackageDrop{cutwin}

2 \newcommand*{\opencutleft}{}
3 \newcommand*{\opencutright}{}
4 \newcommand*{\opencutcenter}{}
5 \newcommand*{\cutfuzz}{}
6
7 \newenvironment{cutout}[4]
8 {\marginpar{\windowpagestuff}}
9 {}
10
11 \newcommand*{\windowpagestuff}{}
12
13 \newcommand*{\pageinwindow}{%
14 % \begin{minipage}{.3\linewidth}
15 \windowpagestuff
16 % \end{minipage}
17 }
18
19 \newenvironment{shapedcutout}[3]
20 {\marginpar{\picinwindow}}
21 {}
22
23 \newcommand*{\putstuffinpic}{}
24
25 \newcommand*{\picinwindow}{%
26 \begin{picture}(0,0)
27 \putstuffinpic
28 \end{picture}}
```

Package 12

lwarp-dcolumn.sty

84 Dcolumn

Pkg dcolumn dcolumn is emulated during HTML output, and the dcolumn package is ignored.

```
1 \LWR@ProvidesPackageDrop{dcolumn}
```

Package 13

lwarp-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 14

lwarp-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 15

lwarp-emptypage.sty

87 Emptypage

Pkg emptypage emptypage is ignored.

for **HTML output**: Discard all options for lwarp-emptypage:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

Package 16

lwarp-endnotes.sty

88 Endnotes

(Based on original code by JOHN LAVAGNINO.)

Pkg endnotes

Discard all options for lwarp-endnotes:

```
for HTML output: 1 \LWR@ProvidesPackagePass{endnotes}

2 \def\noteformat{%
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 17

lwarp-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 `lwarp` does not actually load `enumerate`. While generating HTML, `lwarp` 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 `lwarp-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 18

lwarp-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 19

lwarp-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 20

lwarp-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 21

lwarp-extramarks.sty

93 Extramarks

Pkg `extramarks` `extramarks` is not used.

for HTML output: Discard all options for `lwarp-extramarks`:

```

1 \LWR@ProvidesPackageDrop{extramarks}

2 \newcommand*{\extramarks}[2]{}
3 \newcommand*{\firstleftxmark}{}
4 \newcommand*{\lastleftxmark}{}
5 \newcommand*{\firstrightxmark}{}
6 \newcommand*{\lastrightxmark}{}
7 \newcommand*{\firstxmark}{}
8 \newcommand*{\lastxmark}{}
9 \newcommand*{\topxmark}{}
10 \newcommand*{\topleftxmark}{}
11 \newcommand*{\firstleftmark}{}
12 \newcommand*{\lastrightmark}{}

```

Package 22

lwarp-fancyhdr.sty

94 Fancyhdr

Pkg fancyhdr fancyhdr is nullified.

for HTML output: Discard all options for lwarp-fancyhdr:

```

1 \LWR@ProvidesPackageDrop{fancyhdr}

2 \newcommand*{\fancyhead}[2] [] {}
3 \newcommand*{\fancyfoot}[2] [] {}
4 \newcommand*{\fancyhf}[2] [] {}
5 \newcommand*{\fancypagestyle}[2] {}
6 \newcommand*{\lhead}[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 23

lwarp-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 58.2 for the `\listof` command.

`\newfloat` `{\langle 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` $\{\langle style \rangle\}$

Float styles are ignored.

```
17 \newcommand{\floatstyle}[1]{%  
18 }
```

`\restylefloat` $* \{\langle style \rangle\}$

Float styles are ignored.

```
19 \NewDocumentCommand{\restylefloat}{s m}{%  
20 }
```

Package 24

lwarp-floatflt.sty

96 Floatflt

Pkg floatflt Emulated.

for HTML output: Discard all options for lwarp-floatflt:

```
1 \LWR@ProvidesPackageDrop{floatflt}
```

Borrowed from the lwarp version of keyfloat:

```
2 \NewDocumentEnvironment{KFLTfloatflt@marginfloat}{0{-1.2ex} m}
3 {% start
4 \LWR@maybeinthisfloat%
5 \LWR@forcenewpage
6 \LWR@stoppars%
7 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
8 \LWR@startpars%
9 \captionsetup{type=#2}%
10 }
11 {
12 \LWR@htmldivclassend{div}
13 }
14
15 \DeclareDocumentEnvironment{floatingfigure}{o m}
16 {\begin{KFLTfloatflt@marginfloat}{figure}}
17 {\end{KFLTfloatflt@marginfloat}}
18
19 \DeclareDocumentEnvironment{floatingtable}{o +m}
20 {\begin{KFLTfloatflt@marginfloat}{table}#2}
21 {\end{KFLTfloatflt@marginfloat}}
```

Package 25

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

After everything has loaded, remember whether subcaption was loaded. If not, it is assumed that subfig is used instead:

```
2 \newbool{LWR@subcaptionloaded}
3
4 \AtBeginDocument{
5 \@ifpackageloaded{subcaption}
6 {\booltrue{LWR@subcaptionloaded}}
7 {\boolfalse{LWR@subcaptionloaded}}
8 }
```

```
\floatbox [<1 preamble>] {<2 captype>} [<3 width>] [<4 height>] [<5 vert pos>]
{<6 caption>} {<7 object>}
```

Only parameters for captype, width, caption, and object are used.

LWR@insubfloatrow is true if inside a subfloatrow environment.

There are two actions, depending on the use of `subcaption` or `subfig`.

```

9 \NewDocumentCommand{\floatbox}{o m o o +m +m}{%
10 \ifbool{LWR@subcaptionloaded}%
11 {% subcaption

```

For `subcaption`:

```

12 \ifbool{LWR@insubfloatrow}%
13 {% subcaption in a subfloatrow

```

`subfigure` and `subtable` environments take width as an argument.

```

14 \IfValueTF{#3}%
15 {\@nameuse{sub#2}{#3}}%
16 {\@nameuse{sub#2}{\linewidth}}%
17 }% subcaption in a subfloatrow
18 {% subcaption not in subfloatrow

```

`figure` and `table` environments do not take a width argument.

```

19 \@nameuse{#2}%
20 }% subcaption not in subfloatrow
21 #6
22
23 #7

```

End the environments:

```

24 \ifbool{LWR@insubfloatrow}%
25 {\@nameuse{endsub#2}}%
26 {\@nameuse{end#2}}%
27 }% subcaption
28 {% assume subfig

```

For `subfig`:

```

29 \ifbool{LWR@insubfloatrow}%
30 {% subfig in a subfloatrow

```

`\subfloat` is a macro, not an environment.

Package `subfig`'s `\subfloat` command takes an optional argument which is the caption, but `\floatbox` argument `#6` contains commands to create the caption and label, not the caption itself. Thus, `\caption` is temporarily disabled to return its own argument without braces.

```

31 \begingroup
32 \let\caption\@firstofone

```



```

33 \subfloat[#6]{#7}
34 \endgroup
35 }% subfig in a subfloatrow
36 {% subfig package, but not a subfig

```

figure and table are environments:

```

37 \@nameuse{#2}
38 #6
39
40 #7
41 \@nameuse{end#2}
42 }% subfig package, but not a subfig
43 }% assume subfig
44 }

```

Not used:

```

45 \newcommand*{\nocapbeside}{}
46 \newcommand*{\capbeside}{}
47 \newcommand*{\captop}{}
48 \newlength{\FBwidth}
49 \setlength{\FBwidth}{.3\linewidth}
50 \newlength{\FBheight}
51 \setlength{\FBheight}{2in}
52 \newcommand*{\useFCwidth}{}
53 \newcommand{\floatsetup}[2][{}]{
54 \newcommand{\thisfloatsetup}[1]{
55 \newcommand{\clearfloatsetup}[1]{
56 \newcommand*{\killfloatstyle}{}

```

Preamble and default width are ignored.

```

57 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
58 \@namedef{#1}{
59 \floatbox{#2}
60 }
61 }

```

Preamble and default width are ignored.

```

62 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
63 \@namedef{#1}{%
64 \floatbox{#2}
65 }
66 }

67 \newfloatcommand{ffigbox}{figure}[\nocapbeside][

```

```
68 \newfloatcommand{ttabbox}{table}[\captop][\FBwidth]
```

```
69 \newfloatcommand{fcapside}{figure}[\capbeside][]
```

The row of floats is placed into a <div> of class `floatrow`.

```
70 \newenvironment*{floatrow}[1][2]
```

```
71 {
```

```
72 \LWR@forcenewpage
```

```
73 \BlockClass{floatrow}
```

While inside the `floatrow`, divide the `\linewidth` by the number of floats.

```
74 \booltrue{LWR@infloatrow}
```

```
75 \setlength{\linewidth}{6in/#1}
```

```
76 }
```

```
77 {
```

```
78 \boolfalse{LWR@infloatrow}
```

```
79 \endBlockClass
```

```
80 }
```

Keys for `\DeclareNewFloatType`:

```
81 \newcommand*{\LWR@frowkeyplacement}{}%
```

```
82 \newcommand*{\LWR@frowkeyname}{}%
```

```
83 \newcommand*{\LWR@frowkeyfileext}{}%
```

```
84 \newcommand*{\LWR@frowkeywithin}{}%
```

```
85 \newcommand*{\LWR@frowkeycapstyle}{}%
```

```
86
```

```
87 \define@key{frowkeys}{placement}{}%
```

```
88 \define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}%
```

```
89 \define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}%
```

```
90 \define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}%
```

```
91 \define@key{frowkeys}{relatedcapstyle}{}%
```

Use `\listof{type}{Title}` to print a list of the floats.

```
92 \newcommand*{\DeclareNewFloatType}[2]{%
```

Reset key values:

```
93 \renewcommand*{\LWR@frowkeyplacement}{}%
```

```
94 \renewcommand*{\LWR@frowkeyname}{}%
```

```
95 \renewcommand*{\LWR@frowkeyfileext}{}%
```

```
96 \renewcommand*{\LWR@frowkeywithin}{}%
```

```
97 \renewcommand*{\LWR@frowkeycapstyle}{}%
```

Read new key values:

```

98 \LWR@traceinfo{about to setkeys frowkeys}%
99 \setkeys{frowkeys}{#2}%
100 \LWR@traceinfo{finished setkeys frowkeys}%

```

Create a new float with optional [within]:

```

101 \ifthenelse{\equal{\LWR@frowkeywithin}{}}{
102 {
103 \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\
104 \LWR@frowkeyfileext}%
105 \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}
106 }%
107 {%
108 \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\
109 \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
110 \newfloat{#1}{\LWR@frowkeyplacement}%
111 {\LWR@frowkeyfileext}[\LWR@frowkeywithin}%
112 \LWR@traceinfo{finished newfloat #1}
113 }%

```

Rename the float if a name was given:

```

114 \ifthenelse{\equal{\LWR@frowkeyname}{}}{
115 {}
116 {\floatname{#1}{\LWR@frowkeyname}}%
117 }

```

Not used:

```

118 \newcommand{\buildFBBBOX}[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 26

lwarp-fontenc.sty

98 Fontenc

Pkg fontenc Error if fontenc is loaded after lwarp.

Discard all options for lwarp-fontenc:

for HTML output: 1 \LWR@ProvidesPackageDrop{fontenc}
2 \LWR@loadbefore{fontenc}

Package 27

lwarp-fontspec.sty

99 Fontspec

Pkg fontspec Error if fontspec is loaded after lwarp.

Discard all options for lwarp-fontspec:

for HTML output: 1 \LWR@ProvidesPackageDrop{fontspec}
2 \LWR@loadbefore{fontspec}

Package 28

lwarp-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\p@ \hrule \kern2.6\p@}
15
16 \providecommand*\multiplefootnotemarker}{3sp}
17 \providecommand*\multfootsep}{,}
```

Using cleveref:

```
18 \providecommand*\footref}[1]{\labelcref{#1}}
```

The following work as-is:

```
19 \newcommand\mpfootnotemark{%
20   \@ifnextchar[%
21     \@xmpfootnotemark
22     {%
23       \stepcounter\@mpfn
24       \protected@xdef\@thefnmark{\thempfn}%
25       \@footnotemark
26     }%
27 }
28 \def\@xmpfootnotemark[#1]{%
29   \begingroup
30     \csname c@\@mpfn\endcsname #1\relax
31     \unrestored@protected@xdef\@thefnmark{\thempfn}%
32   \endgroup
33   \@footnotemark
34 }
```

Package 29

lwarp-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 lwarp
6 }
7
8 % \let\fn@endnote\color@endgroup% *** conflicts with lwarp
9 \newcommand*{\fn@endnote}{\LWR@htmltagc{/LWR@tagregularparagraph}}
```

Removed print-version formatting:

```
10 \def\fn@startfntext{%
11   \setbox\z@\vbox\bgroup%
12     \fn@startnote%
13     \fn@prefntext%
14     \ignorespaces%
15 }
```

Removed print-version formatting, added closing paragraph tag:

```
16 \def\fn@endfntext{%
17 \LWR@htmltagc{/LWR@tagregularparagraph}%
18   \fn@postfntext%
19   \egroup%
20   \begingroup%
21   \let\@makefntext\@empty%
22   \let\@finalstrut\@gobble%
23   \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 30

lwarp-footnotehyper.sty

102 Footnotehyper

Pkg footnotehyper footnotehyper is a hyperref-safe version of footnote. For lwarp, footnotehyper is emulated.

for HTML output: Discard all options for lwarp-footnotehyper:

```
1 \RequirePackage{footnote}
2 \LWR@ProvidesPackageDrop{footnotehyper}
```


Package 31

lwarp-framed.sty

103 Framed

(Based on original code by DONALD ARSENEAU.)

Pkg `framed` `framed` is supported and patched by `lwarp`.

for HTML output: Accept all options for `lwarp-framed`:

```

1 \LWR@ProvidesPackagePass{framed}

2
3 \renewenvironment{framed}{%
4 \LWR@forcenewpage
5 \BlockClass{framed}%
6 }
7 {\endBlockClass}
8
9 \renewenvironment{oframed}{%
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 {<toptitle>} {<bottitle>} {<thicknesstop>} {<bottom>} {<left>} {<right>}
{<text contents>}

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}}{0pt}{% not empty
66 \begin{BlockClass}{framedtitle}[background: \#\LWR@tempcolor]%
67 \textcolor{TFTitleColor}{\textbf{#1}}%
68 \end{BlockClass}
69 }% not empty
70
71 #7
72
73 \ifthenelse{\isempty{#2}}{0pt}{% not empty
74 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
75 \begin{BlockClass}{framedtitle}[background: \#\LWR@tempcolor]%
76 \textcolor{TFTitleColor}{\textbf{#2}}%
77 \end{BlockClass}
78 }% not empty
79 \end{BlockClass}

```

```

80 }

\TitleBarFrame [\marker] {\i\title} {\i\contents}

81 \renewcommand\TitleBarFrame[3] [] {
82 \CustomFBox
83   {\i2}{\i}%
84   \fboxrule\fboxrule\fboxrule\fboxrule
85   {\i3}%
86 }

87 \renewcommand{\TF@Title}[1]{\i1}

MakeFramed {\i\settings}

88 \let\MakeFramed\relax
89 \let\endMakeFramed\relax
90
91 \NewEnviron{MakeFramed}[1]{%
92 \FrameCommand{\begin{minipage}{\i\linewidth}\BODY\end{minipage}}%
93 }

\fb@put@frame {\i\frame cmd no split} {\i\frame cmd split}

94 \renewcommand*\fb@put@frame[2]{%
95 \relax%
96 \@tempboxa%
97 }

```

Package 32

lwarp-ftnright.sty

104 Ftnright

Pkg `ftnright` `ftnright` is ignored.

for HTML output: Discard all options for `lwarp-ftnright`:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

Package 33

lwarp-geometry.sty

105 Geometry

Pkg `geometry` `geometry` is preloaded by `lwarp`, but must be nullified as seen by the user's source code.

for HTML output: Discard all options for `lwarp-geometry`:

```
1 \LWR@ProvidesPackageDrop{geometry}

2 \renewcommand*{\geometry}[1]{}
3 \renewcommand*{\newgeometry}[1]{}
4 \renewcommand*{\restoregeometry}{}
5 \renewcommand*{\savegeometry}[1]{}
6 \renewcommand*{\loadgeometry}[1]{}

```

Package 34

lwarp-glossaries.sty

106 Glossaries

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

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

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

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

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

for HTML output:

```

1 \PassOptionsToPackage{xindy}{glossaries}
2 \LWR@ProvidesPackagePass{glossaries}
3 \setupglossaries{nonumberlist}
4 \setglossarystyle{index}
```

Package 35

lwarp-graphics.sty

107 Graphics

Pkg graphics graphics is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphics}

Package 36

lwarp-graphicx.sty

108 Graphicx

Pkg graphicx graphicx is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphicx}

Package 37

lwarp-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 lwarp html version of package 'hyperref' -- discarding options.}
3 \typeout{   Are not using ProvidesPackage, so that other packages}
4 \typeout{   do not attempt to patch lwarp's version of 'hyperref'.}
5 % \ProvidesPackage{lwarp-#1-#2}
6 \DeclareOption*{}
7 \ProcessOptions\relax

8 \newcommand*{\hypersetup}[1]{}
9 \newcommand*{\hyperbaseurl}[1]{}

```

Insert an image with alt text:

```

10 \NewDocumentCommand{\LWR@hyperimageb}{m +m}{%
11 \LWR@htmltag{img src="#1" alt="#2" class="hyperimage"{} }%
12 \endgroup%
13 \LWR@ensuredoingapar%
14 }
15
16 \newcommand{\hyperimage}{%
17 \LWR@ensuredoingapar%
18 \begingroup\catcode'\_ =12
19 \LWR@hyperimageb%
20 }

```

Creates an HTML anchor to category.name with the given text.

```

21 \NewDocumentCommand{\hyperdef}{m m +m}{%
22 \LWR@ensuredoingapar%
23 \LWR@subsublabel{#1.#2}%
24 #3%
25 }

```

Creates an HTML link to URL#category.name with the given text.

```

26 \NewDocumentCommand{\LWR@hyperrefb}{m m m +m}{%
27 \LWR@htmltag{a href="#1\LWR@hashmark#2.#3"%

```

```

28 #4%
29 \LWR@htmltag{/a}%
30 \endgroup%
31 }

```

Creates text as an HTML link to the L^AT_EX label.

```

32 \NewDocumentCommand{\LWR@hyperrefc}{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 38

lwarp-indentfirst.sty

110 Indentfirst

Pkg indentfirst indentfirst is ignored.

Discard all options for lwarp-indentfirst:

for HTML output: 1 \LWR@ProvidesPackageDrop{indentfirst}

Package 39

lwarp-inputenc.sty

111 Inputenc

Pkg inputenc Error if inputenc is loaded after lwarp.

Discard all options for lwarp-inputenc:

for HTML output: 1 \LWR@ProvidesPackageDrop{inputenc}

2 \LWR@loadbefore{inputenc}

Package 40

lwarp-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}{0{-1.2ex} m}
15 {% start
16 \LWR@maybeinthisfloat%
17 \LWR@forcenewpage
18 \LWR@stoppars%
19 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
20 \LWR@startpars%
21 \captionsetup{type=#2}%
22 }
23 {
24 \LWR@htmldivclassend{div}
25 }

26 \DeclareDocumentEnvironment{marginfigure}{o}
27 {\begin{KFLT@marginfloat}{figure}}
28 {\end{KFLT@marginfloat}}
29
30 \DeclareDocumentEnvironment{margintable}{o}
31 {\begin{KFLT@marginfloat}{table}}
32 {\end{KFLT@marginfloat}}
```

```
33 \DeclareDocumentEnvironment{keywrap}{m +m}
34 {%
35 \begin{BlockClass}{marginblock}
36 \setlength{\linewidth}{#1}
37 #2%
38 \end{BlockClass}
39 }
40 {%
41 }

42 }% AtBeginDocument
```

Package 41

lwarp-layout.sty

113 Layout

Pkg layout layout is ignored.

for HTML output: Discard all options for lwarp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}
2 \NewDocumentCommand{\layout}{s}{}
```

Package 42

lwarp-letterspace.sty

114 Letterspace

Pkg letterspace letterspace is a subset of microtype, which is pre-loaded by lwarp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for lwarp-letterspace:

```
1 \LWR@ProvidesPackageDrop{letterspace}
2 \newcommand*\lsstyle{}
3 \newcommand\textls[2] [] {}
4 \def\textls#1#{}
5 \newcommand*\lslig[1]{#1}
```

Package 43

lwarp-lettrine.sty

115 Lettrine

(Based on original code by DANIEL FLIPO.)

Pkg lettrine Emulated.

for HTML output: Discard all options for lwarp-lettrine:

```
1 \LWR@ProvidesPackageDrop{lettrine}
```

The initial letter is in a `` of class `lettrine`, and the following text is in a `` of class `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 44

lwarp-lips.sty

116 Lips

Pkg lips lips is emulated during HTML output, and the lips package is ignored.

```
1 % \LWR@ProvidesPackageDrop{lips}
2 \PackageInfo{lwarp}{Using the lwarp version of package 'lips'.}%
3 \ProvidesPackage{lwarp-lips}
4
5 \NewDocumentCommand{\Lips}{-}{\textellipsis}
6
7 \NewDocumentCommand{\BracketedLips}{-}{[\textellipsis]}
8
9 \let\lips\Lips
10 \let\olips\lips
11
12 \DeclareOption*{}
13 \DeclareOption{mla}{
14 \let\lips\BracketedLips
15 }
16 \ProcessOptions\relax
17
18 \newcommand \LPNobreakList {}
```

Package 45

lwarp-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@origlsthkEveryPar\lsthk@EveryPar`

`7`

`8 \renewcommand{\l@lstlisting}[2]{\hypertocfloat{1}{\lstlisting}{1ol}{#1}{#2}}`

Done at the start of a listing.

`9 \renewcommand{\lst@Init}[1]{%`

First, perform the listings initialization:

`10 \LWR@traceinfo{\lst@Init}%`

`11 \renewcommand*\@capttype{\lstlisting}%`

`12 \LWR@origlst@Init{#1}%`

`13 \LWR@traceinfo{finished origlst@Init}%`

`14 \lst@ifdisplaystyle%`

Creating a display.

Disable line numbers, produce the `<pre>`, then reenable line numbers.

`15 \LWR@traceinfo{About to create verbatim.}%`

`16 \let\lsthk@EveryPar\relax%`

`17 \LWR@forcenewpage`

`18 \LWR@atbeginverbatim{programlisting}%`

`19`

`20 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%`

`21 \else%`

Inline, so open a ``

```
22 \ifbool{LWR@verbtags}{\LWR@htmltag{span class="inlineprogramlisting"}}{}%
23 \fi%
24 }
```

```
25 \renewcommand*{\lst@DeInit}{%
26 \lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the `</pre>`, then reenable line numbers:

```
27 \let\lsthk@EveryPar\relax%
28
29 \LWR@afterendverbatim%
30 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
31 \else%
```

Inline, so create the closing ``:

```
32 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
33 \fi%
```

Final listings deinit:

```
34 \LWR@origlst@DeInit%
35 }
```

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```
36 \def\lst@MakeCaption#1{%
37 \LWR@traceinfo{MAKING CAPTION at #1}%
38 \lst@ifdisplaystyle
39 \LWR@traceinfo{making a listings display caption}%
40 \ifx #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\lstname\lst@name
50 \lst@ifnolol\else
51 \ifx\lst@@caption\@empty
```

```

52             \ifx\lst@caption\@empty
53             \ifx\lst@intname\@empty \else \def\lst@temp{ }%
54             \ifx\lst@intname\lst@temp \else

```

This code places a contents entry for a non-float. This would have to be modified for lwarp:

```

55 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
56 %             \addcontentsline{lol}{lstlisting}{\lst@name}
57             \fi\fi
58             \fi
59             \else

```

This would have to be modified for lwarp:

```

60 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
61             \addcontentsline{lol}{lstlisting}%
62 {\protect\numberline{\thelstlisting}%
63 {\protect\ignorespaces \lst@@caption \protect\relax}}%
64             \fi
65             \fi
66             \fi
67             \ifx\lst@caption\@empty\else
68 \LWR@traceinfo{lst@caption not empty-}%
69             \lst@ifsubstring #1\lst@captionpos
70             {\begingroup
71 \LWR@traceinfo{at the selected position}%

```

These space and box commands are not needed for HTML output:

```

72 %             \let\@vskip\vskip
73 %             \def\vskip{\afterassignment\lst@vskip \@tempkipa}%
74 %             \def\lst@vskip{\nobreak\@vskip\@tempkipa\nobreak}%
75 %             \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
76 %             \ifx #1t\allowbreak \fi
77             \ifx\lst@title\@empty

```

New lwarp code to create a caption:

```

78             \lst@makecaption\fnnum\lstlisting{\ignorespaces \lst@caption}
79             \else

```

New lwarp code to create a title:

```

80 %             \lst@maketitle\lst@title % (AS)
81 \LWR@traceinfo{Making title: \lst@title}%
82 \begin{BlockClass}{lstlistingtitle}% lwarp
83 \lst@maketitle\lst@title% lwarp
84 \end{BlockClass}% lwarp

```

```

85             \fi
86 \LWR@traceinfo{About to assign label: !\lst@label!}%
87             \ifx\lst@label\@empty\else
88 \leavevmode% gets rid of bad space factor error
89 \GetTitleStringExpand{\lst@caption}%
90 \edef\LWR@lntemp{\GetTitleStringResult}%
91 \edef\@currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
92 \label{\lst@label}\fi
93 \LWR@traceinfo{Finished assigning the label.}%

```

Not needed for lwarp:

```

94 %             \ifx #1b\allowbreak \fi
95             \endgroup{}%
96     \fi
97 \LWR@traceinfo{end of making a listings display caption}%
98     \else
99 \LWR@traceinfo{INLINE}%
100    \fi
101 \LWR@traceinfo{DONE WITH CAPTION at #1}%
102 }

```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```

103 \lst@Key{numbers}{none}{%
104     \let\lst@PlaceNumber\@empty
105     \lstKV@SwitchCases{#1}%
106     {none&\\%
107         left&\def\lst@PlaceNumber{%
108 % \llap{
109 \LWR@orignormalfont%
110 \lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
111 % }
112 }
113 \\%
114     right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont
115         \kern\VerbatimHTMLWidth \kern\lst@numbersep
116         \lst@numberstyle{\thelstnumber}}}%
117     }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}
118 \end{warpHTML}

```

Package 46

lwarp-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` `* [⟨horizontal⟩] {⟨colspec⟩}` Emulates the `longtable` environment.

Per the `caption` package, the starred version steps the counter per caption. The unstarred version steps the counter once at the beginning, but not at each caption.

Options `[c]`, `[l]`, and `[r]` are thrown away.

```
2 \newenvironment{longtable*}[2][{}]{%
3 \LWR@floatbegin{table}%
4 \setcaptiontype{LTcaption}%
5 \caption@setoptions{longtable}%
6 \caption@setoptions{@longtable}%
7 \caption@LT@setup%
8 \booltrue{LWR@starredlongtable}%
9 \let\captionlistentry\LWR@LTcaptionlistentry%
10 \LWR@tabular{#2}
11 }
```

```

12 {\endLWR@tabular\LWR@floatend}
13
14 \newenvironment{longtable}[2][\%
15 \LWR@floatbegin{table}%
16 \setcaptiontype{\LTcapttype}%
17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{\LTcapttype}%
21 \let\captionlistentry\LWR@LTcaptionlistentry%
22 \LWR@tabular{#2}
23 }
24 {\endLWR@tabular\LWR@floatend}
25

```

Provided for compatibility, but ignored:

```

26 \newcounter{LTchunksz}
27 \def\endhead{\LWR@tabularendofline}% throws away options //[dim] and /*
28 \def\endfirsthead{\LWR@tabularendofline}
29 \def\endfoot{\LWR@tabularendofline}
30 \def\endlastfoot{\LWR@tabularendofline}
31 \newcommand\tabularnewline{\LWR@tabularendofline}
32 \newcommand{\setlongtables}{}% Obsolete command, does nothing.
33 \newlength{\LTleft}
34 \newlength{\LTright}
35 \newlength{\LTpre}
36 \newlength{\LTpost}
37 \newlength{\LTcapwidth}

38 \renewcommand*{\kill}{\LWR@tabularendofline}

```

Package 47

lwarp-lscape.sty

119 Lscape

Pkg lscape lscape is nullified.
 for HTML output: Discard all options for lwarp-lscape.

```

1 \LWR@ProvidesPackageDrop{lscape}

2 \newenvironment*{landscape}{}{}

```

Package 48

lwarp-ltcaption.sty

120 Ltcaption

Pkg ltcaption ltcaption is emulated during HTML output, and the ltcaption package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{ltcaption}`

`\LTcaptype` is already defined by lwarp.

`longtable*` is already defined by lwarp-longtable.

```

2 \newlength{\LTcapskip}
3 \newlength{\LTcapleft}
4 \newlength{\LTcapright}
5 \newcommand*{\LTcapmarginfalse}{}

```

Package 49

lwarp-marginfix.sty

121 Marginfix

Pkg marginfix Not used.

for HTML output: Discard all options for lwarp-marginfix:

```
1 \LWR@ProvidesPackageDrop{marginfix}

2 \newcommand*{\marginskip}[1]{}
3 \newcommand*{\clearmargin}{}
4 \newcommand*{\softclearmargin}{}
5 \newcommand*{\extendmargin}[1]{}
6 \newcommand*{\mparshift}[1]{}
7 \newdimen\marginheightadjustment
8 \newdimen\marginposadjustment
9 \newcommand*{\blockmargin}[1][\marginheightadjustment]{}
10 \newcommand*{\unblockmargin}[1][\marginposadjustment]{}
11 \newcommand*{\marginphantom}[2][\marginheightadjustment]{}

```

Package 50

lwarp-marginnote.sty

122 Marginnote

Pkg marginnote Emulated.

for HTML output: Discard all options for lwarp-marginnote:

```
1 \LWR@ProvidesPackageDrop{marginnote}

2 \NewDocumentCommand{\marginnote}{o +m o}{\marginpar{#2}}
3 \newcommand*{\marginnoteleftadjust}{}
4 \newcommand*{\marginnoterightadjust}{}
5 \newcommand*{\marginnotetextwidth}{}
6 \let\marginnotetextwidth\textwidth
7 \newcommand*{\marginnotevadjust}{}
8 \newcommand*{\marginfont}{}
9 \newcommand*{\raggedleftmarginnote}{}
10 \newcommand*{\raggedrightmarginnote}{}

```

Package 51

lwarp-mcaption.sty

123 Mcaption

Pkg mcaption mcaption is nullified.

for HTML output: Discard all options for lwarp-mcaption:

```
1 \LWR@ProvidesPackageDrop{mcaption}

2 \newenvironment{margincap}{}{}
3 \newcommand*{\margincapalign}{}
4 \newlength{\margincapsep}

```


Package 52


lwarp-mdframed.sty

124 Mdfamed

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

font For title font, use

`frametitlefont=\textbf,`

instead of

`frametitlefont=\bfseries,`

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

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

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

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

CSS classes Environments created or encapsulated by **mdframed** are enclosed in a `<div>` of class `md<environmentname>`, or **mdframed** otherwise.

Frame titles are placed into a `` of class `mdframedtitle`. Subtitles are in a `` of class `mdframedsubtitle`, and likewise for subsubtitles.

Pre-existing hooks are used to patch extra functions before and after the frames.

amsthm must be loaded before **mdframed**

`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@htmltagc{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@htmltagc{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 0pt 0pt
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%
```

Close the span and unnest the paragraph tag restriction:

```
75 \LWR@htmltagc{/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 {% the following empty line is required
```

```
81
```

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@htmltagc{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@htmltagc{/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@mdfsubsubtitlecommon{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 }

```

$[\langle \textit{numberedlike} \rangle]$ $\{\langle \textit{caption} \rangle\}$ $[\langle \textit{within} \rangle]$

Modified from the original to remember the environment.

```

139 \let\LWR@origmdtheorem\mdtheorem
140
141 \DeclareDocumentCommand{\LWR@mdtheorem}{0}{ 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

```

$[\langle \textit{numberedlike} \rangle]$ $\{\langle \textit{caption} \rangle\}$ $[\langle \textit{within} \rangle]$

Modified from the original to remember the environment.

```

147 \DeclareDocumentCommand\newmdtheoremenv{0}{ 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     }%
154   \BeforeBeginEnvironment{#2}{%
155     \renewcommand*\LWR@mdthisenv}{md#2}%
156     \begin{mdframed}[#1]}%
157   \AfterEndEnvironment{#2}{%
158     \end{mdframed}}%
159 }

```

Package 53

lwarp-microtype.sty

125 Microtype

Pkg microtype microtype is pre-loaded by lwarp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for lwarp-microtype:

```

1 \LWR@ProvidesPackageDrop{microtype}

2 \DeclareDocumentCommand{\DeclareMicrotypeSet}{o m m}{}
3 \DeclareDocumentCommand{\UseMicrotypeSet}{o m}{}
4 \DeclareDocumentCommand{\DeclareMicrotypeSetDefault}{o m}{}
5 \DeclareDocumentCommand{\SetProtrusion}{o m m}{}
6 \DeclareDocumentCommand{\SetExpansion}{o m m}{}
7 \DeclareDocumentCommand{\SetTracking}{o m m}{}
8 \DeclareDocumentCommand{\SetExtraKerning}{o m m}{}
9 \DeclareDocumentCommand{\SetExtraSpacing}{o m m}{}
10 \DeclareDocumentCommand{\DisableLigatures}{o m}{}
11 \DeclareDocumentCommand{\DeclareCharacterInheritance}{o m m}{}
12 \DeclareDocumentCommand{\DeclareMicrotypeVariants}{m}{}
13 \DeclareDocumentCommand{\DeclareMicrotypeAlias}{m m}{}
14 \DeclareDocumentCommand{\LoadMicrotypeFile}{m}{}
15 \DeclareDocumentCommand{\DeclareMicrotypeBabelHook}{m m}{}
16 \DeclareDocumentCommand{\microtypesetup}{m}{}
17 \DeclareDocumentCommand{\microtypecontext}{m}{}
18 \DeclareDocumentCommand{\textmicrotypecontext}{m m}{#2}
19 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
20 \DeclareDocumentCommand{\lsstyle}{}{}
21 \DeclareDocumentCommand{\textls}{o +m}{}
22 \DeclareDocumentCommand{\slig}{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 54

lwarp-mparhack.sty

126 Mparhack

Pkg mparhack Not used.

for **HTML output:** Discard all options for lwarp-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```

Package 55

lwarp-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 56

lwarp-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 57

lwarp-nameref.sty

129 Nameref

Pkg nameref nameref is emulated by lwarp.

for HTML output: Discard all options for lwarp-nameref:

```
1 \typeout{Using the lwarp html version of package 'nameref' -- discarding options.}
2 \typeout{   Are not using ProvidesPackage, so that other packages}
3 \typeout{   do not attempt to patch lwarp's version of 'nameref'.}
4 \DeclareOption*{}
5 \ProcessOptions\relax
```

Package 58

lwarp-needspace.sty

130 Needspace

Pkg `needspace` `needspace` is not used during HTML conversion.

for HTML output: Discard all options for `lwarp-needspace`:

```
1 \LWR@ProvidesPackageDrop{needspace}  
2  
3 \newcommand*{\needspace}[1]{  
4 \DeclareDocumentCommand{\Needspace}{s m}{}
```

Package 59

lwarp-newclude.sty

131 Newclude

Pkg `newclude` Error if `newclude` is loaded after `lwarp`.

Discard all options for `lwarp-newclude`:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{newclude}  
  
2 \LWR@loadbefore{newclude}
```

Package 60

lwarp-newunicodechar.sty

132 Newunicodechar

Pkg newunicodechar Error if newunicodechar is loaded after lwarp.
Discard all options for lwarp-newunicodechar:

for HTML output: 1 \LWR@ProvidesPackageDrop{newunicodechar}

2 \LWR@loadbefore{newunicodechar}

Package 61

lwarp-nextpage.sty

133 Nextpage

Pkg nextpage nextpage is nullified.

for HTML output: Discard all options for lwarp-nextpage.

1 \LWR@ProvidesPackageDrop{nextpage}

2 \newcommand{\cleartoevenpage}[1] [] {}

3 \newcommand{\movetoevenpage}[1] [] {}

4 \newcommand{\cleartooddpage}[1] [] {}

5 \newcommand{\movetooddpage}[1] [] {}

Package 62

lwarp-nowidow.sty

134 Nowidow

Pkg **nowidow** nowidow is not used during HTML conversion.

Discard all options for lwarp-nowidow:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{nowidow}

2 \newcommand*{\nowidow}[1] [] {}
3 \newcommand*{\setnowidow}[1] [] {}

4 \newcommand*{\noclub}[1] [] {}
5 \newcommand*{\setnoclub}[1] [] {}
```

Package 63

lwarp-ntheorem.sty

135 Ntheorem

(Based on original code by WOLFGANG MAY, ANDREAS SCHEDLER.)

Pkg `ntheorem` `ntheorem` is patched for use by `lwarp`.

CSS styling of theorems and proofs:

Theorem: `<div>` of class `theorembody<theoremstyle>`

Theorem Header: `` of class `theoremheader<style>`

where `<theoremstyle>` is `plain`, `break`, etc.

⚠ **Font control** This conversion is not total. Font control is via CSS, and the custom L^AT_EX font settings are ignored.

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

Options `amsthm` or `standard` choose which set of theorems and proofs to initialize.

⚠ **Disabled options** The options `thmmarks` and `amsmath` are disabled, since they heavily modify the underlying math code. Theorem marks are emulated. The AMS-math modifications are not done.

Option `thref` is disabled because `cleveref` functions are used instead. `\thref` is emulated.

Option `hyperref` is disabled because `lwarp` emulated `hyperref`.

for HTML output: Some disabled options:

```
1 \DeclareOption{thref}{}
2
3
4 \newbool{LWR@ntheoremmarks}
5 \boolfalse{LWR@ntheoremmarks}
6
7 \DeclareOption{thmmarks}{
```



```

8 \booltrue{LWR@theoremmarks}
9 \newif\ifsetendmark\setendmarktrue
10 }
11
12
13 \newbool{LWR@theoremamsthm}
14 \boolfalse{LWR@theoremamsthm}
15
16 \DeclareOption{amsthm}{\booltrue{LWR@theoremamsthm}}
17
18
19 \DeclareOption{amsmath}{}
20 \DeclareOption{hyperref}{}
21
22
23 \LWR@ProvidesPackagePass{theorem}

```

Storage for the style being used for new theorems.

```
24 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```

25 \gdef\theoremstyle#1{%
26   \@ifundefined{th@#1}{\@warning
27     {Unknown theoremstyle ‘#1’. Using ‘plain’}%
28     \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\xnthm#1#2[#3]{%
38   \ifthm@tempif
39     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
40     \expandafter\@ifundefined{c@#1}%
41       {\@definecounter{#1}}{}%
42     \@newctr{#1}[#3]%
43     \expandafter\xdef\csname the#1\endcsname{%
44       \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
45       {\noexpand\csname\the\theoremnumbering\endcsname{#1}}}%
46     \expandafter\gdef\csname mkheader@#1\endcsname

```

```

47     {\csname setparms@#1\endcsname
48       \@thm{#1}{#1}{#2}
49 }%
50   \global\@namedef{end#1}{\@endtheorem}
51   \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
52   \fi
53 }
54
55 \gdef\@ynthm#1#2{%
56   \ifthm@tempif
57     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
58     \expandafter\@ifundefined{c@#1}%
59       {\@definecounter{#1}}{}%
60     \expandafter\xdef\csname the#1\endcsname
61       {\noexpand\csname\the\theoremnumbering\endcsname{#1}}%
62     \expandafter\gdef\csname mkheader@#1\endcsname
63       {\csname setparms@#1\endcsname
64         \@thm{#1}{#1}{#2}
65 }%
66     \global\@namedef{end#1}{\@endtheorem}
67     \AtBeginEnvironment{#1}{\edef\LWR@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@addtheoremeline#1\endcsname{%
77       \noexpand\@num@addtheoremeline{#1}{#3}}%
78     \expandafter\protected@xdef\csname nonum@addtheoremeline#1\endcsname{%
79       \noexpand\@nonum@addtheoremeline{#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@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
95 \LWR@startpars%
96 }

```

The following are patched for CSS.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc. They are gathered together here.

Each theorem is encased in a `BlockClass` environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader<style>`.

```

97 \gdef\newtheoremstyle#1#2#3{%
98   \expandafter\@ifundefined{th@#1}%
99   {\expandafter\gdef\csname th@#1\endcsname{%
100     \def\@begintheorem####1####2{%
101 \LWR@forcenewpage% new
102 \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
103 \LWR@inctheorem% new
104 #2}%
105   \def\@opargbegintheorem####1####2####3{%
106 \LWR@forcenewpage% new
107 \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
108 \LWR@inctheorem% new
109 #3}%
110 }%
111 }%
112 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
113 }
114
115 \renewtheoremstyle{plain}%
116   {\item[\hskip\labelsep \theorem@headerfont
117 \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]]}%
118   {\item[\hskip\labelsep \theorem@headerfont
119 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]]}
120
121 \renewtheoremstyle{break}%
122   {\item[
123 % \rlap{\vbox{\hbox{
124 \hskip\labelsep \theorem@headerfont
125 \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
126 % } \hbox{\strut}}}]
127 ]}%
128   {\item[
129 % \rlap{\vbox{\hbox{
130 \hskip\labelsep \theorem@headerfont
131 \InlineClass{theoremheaderbreak}{##1\ ##2\ (##3)\theorem@separator}\newline

```

```

132 % }\hbox{\strut}}
133 ]}
134
135 \renewtheoremstyle{change}%
136 {\item[\hskip\labelsep
137 \theorem@headerfont
138 \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]}%
139 {\item[\hskip\labelsep
140 \theorem@headerfont
141 \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]}
142
143 \renewtheoremstyle{changebreak}%
144 {\item[
145 % \rlap{\vbox{\hbox{
146 \hskip\labelsep \theorem@headerfont
147 \InlineClass{theoremheaderchangebreak}{##2\ ##1\theorem@separator}\newline
148 % }\hbox{\strut}}
149 ]}%
150 {\item[
151 % \rlap{\vbox{\hbox{
152 \hskip\labelsep \theorem@headerfont
153 \InlineClass{theoremheaderchangebreak}{##2\ ##1\ (##3)\theorem@separator}\newline
154 % }\hbox{\strut}}
155 ]}
156
157 \renewtheoremstyle{margin}%
158 {\item[\hskip\labelsep\theorem@headerfont
159 \InlineClass{theoremheadermargin}{##2 \quad ##1\theorem@separator}
160 ]}%
161 {\item[\hskip\labelsep\theorem@headerfont
162 \InlineClass{theoremheadermargin}{##2 \quad ##1\ (##3)\theorem@separator}
163 ]}
164
165 \renewtheoremstyle{marginbreak}%
166 {\item[\hskip\labelsep\theorem@headerfont
167 \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\theorem@separator}\newline
168 ]}%
169 {\item[\hskip\labelsep\theorem@headerfont
170 \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\ (##3)\theorem@separator}\newline
171 ]}
172
173 \renewtheoremstyle{nonumberplain}%
174 {\item[\theorem@headerfont\hskip\labelsep
175 \InlineClass{theoremheaderplain}{##1\theorem@separator}]}%
176 {\item[\theorem@headerfont\hskip\labelsep
177 \InlineClass{theoremheaderplain}{##1\ (##3)\theorem@separator}]}
178
179 \renewtheoremstyle{nonumberbreak}%
180 {\item[
181 % \rlap{\vbox{\hbox{

```

```

182 \hskip\labelsep \theorem@headerfont
183 \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline
184 % }\hbox{\strut}}
185 ]}%
186 {\item[
187 % \rlap{\vbox{\hbox{
188 \hskip\labelsep \theorem@headerfont
189 \InlineClass{theoremheaderbreak}{##1\ (##3)\theorem@separator}\newline
190 % }\hbox{\strut}}
191 ]}
192
193 \renewtheoremstyle{empty}%
194 {\item[]}%
195 {\item[\theorem@headerfont \hskip\labelsep\relax
196 \InlineClass{theoremheaderplain}{##3}]}
197
198 \renewtheoremstyle{emptybreak}%
199 {\item[]}%
200 {\item[\theorem@headerfont \hskip\labelsep\relax
201 \InlineClass{theoremheaderplain}{##3}] \ \newline}

```

The following manually adjust the CSS for the standard configuration objects which are not a purely plain style:

```

202 \ifbool{LWR@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@theoremamsthm}{
243
244 \gdef\th@plain{%
245   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
246   \def\@begintheorem##1##2{%
247     \LWR@forcenewpage% new
248     \BlockClass{theorembodyplain}% new
249     \LWR@inctheorem% new
250     \item[\hskip\labelsep
251 %   \theorem@headerfont
252     \InlineClass{theoremheaderplain}{##1\ ##2.}
253   ]}%
254   \def\@opargbegintheorem##1##2##3{%
255     \LWR@forcenewpage% new
256     \BlockClass{theorembodyplain}% new
257     \LWR@inctheorem% new
258     \item[\hskip\labelsep
259 %   \theorem@headerfont
260     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
261   ]}}
262
263 \gdef\th@nonumberplain{%
264   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
265   \def\@begintheorem##1##2{%
266     \LWR@forcenewpage% new
267     \BlockClass{theorembodyplain}% new
268     \LWR@inctheorem% new
269     \item[\hskip\labelsep
270 %   \theorem@headerfont

```

```

271 \InlineClass{theoremheaderplain}{##1.}
272 ]}%
273 \def\@opargbegintheorem##1##2##3{%
274 \LWR@forcenewpage% new
275 \BlockClass{theorembodyplain}% new
276 \LWR@inctheorem% new
277 \item[\hskip\labelsep
278 % \theorem@headerfont
279 \InlineClass{theoremheaderplain}{##1\ (##3).}
280 ]}}
281
282 \gdef\th@definition{%
283 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
284 \def\@begintheorem##1##2{%
285 \LWR@forcenewpage% new
286 \BlockClass{theorembodydefinition}% new
287 \LWR@inctheorem% new
288 \item[\hskip\labelsep
289 % \theorem@headerfont
290 \InlineClass{theoremheaderdefinition}{##1\ ##2.}
291 ]}%
292 \def\@opargbegintheorem##1##2##3{%
293 \LWR@forcenewpage% new
294 \BlockClass{theorembodydefinition}% new
295 \LWR@inctheorem% new
296 \item[\hskip\labelsep
297 % \theorem@headerfont
298 \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
299 ]}}
300
301 \gdef\th@nonumberdefinition{%
302 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
303 \def\@begintheorem##1##2{%
304 \LWR@forcenewpage% new
305 \BlockClass{theorembodydefinition}% new
306 \LWR@inctheorem% new
307 \item[\hskip\labelsep
308 % \theorem@headerfont
309 \InlineClass{theoremheaderdefinition}{##1.}
310 ]}%
311 \def\@opargbegintheorem##1##2##3{%
312 \LWR@forcenewpage% new
313 \BlockClass{theorembodydefinition}% new
314 \LWR@inctheorem% new
315 \item[\hskip\labelsep
316 % \theorem@headerfont
317 \InlineClass{theoremheaderdefinition}{##1\ (##3).}
318 ]}}
319
320 \gdef\th@remark{%

```

```

321 \def\theorem@headerfont{\itshape}\normalfont%
322 \def\@begintheorem##1##2{%
323 \LWR@forcenewpage% new
324 \BlockClass{theorembodyremark}% new
325 \LWR@inctheorem% new
326 \item[\hskip\labelsep
327 % \theorem@headerfont
328 \InlineClass{theoremheaderremark}{##1\ ##2.}
329 ]}%
330 \def\@opargbegintheorem##1##2##3{%
331 \LWR@forcenewpage% new
332 \BlockClass{theorembodyremark}% new
333 \LWR@inctheorem% new
334 \item[\hskip\labelsep
335 % \theorem@headerfont
336 \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
337 ]}}
338
339 \gdef\th@nonumberremark{%
340 \def\theorem@headerfont{\itshape}\normalfont%
341 \def\@begintheorem##1##2{%
342 \LWR@forcenewpage% new
343 \BlockClass{theorembodyremark}% new
344 \LWR@inctheorem% new
345 \item[\hskip\labelsep
346 % \theorem@headerfont
347 \InlineClass{theoremheaderremark}{##1.}
348 ]}%
349 \def\@opargbegintheorem##1##2##3{%
350 \LWR@forcenewpage% new
351 \BlockClass{theorembodyremark}% new
352 \LWR@inctheorem% new
353 \item[\hskip\labelsep
354 % \theorem@headerfont
355 \InlineClass{theoremheaderremark}{##1\ (##3).}
356 ]}}
357
358 \gdef\th@proof{%
359 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
360 \def\@begintheorem##1##2{%
361 \LWR@forcenewpage% new
362 \BlockClass{theorembodyproof}% new
363 \LWR@inctheorem% new
364 \item[\hskip\labelsep
365 % \theorem@headerfont
366 \InlineClass{theoremheaderproof}{##1.}
367 ]}%
368 \def\@opargbegintheorem##1##2##3{%
369 \LWR@forcenewpage% new
370 \BlockClass{theorembodyproof}% new

```

```

371 \LWR@inctheorem% new
372     \item[\hskip\labelsep
373 % \theorem@headerfont
374 \InlineClass{theoremheaderproof}{##1\ (##3).}
375 ]}}
376
377
378
379 \newcounter{proof}%
380 \if@thmmarks
381 \newcounter{currproofctr}%
382 \newcounter{endproofctr}%
383 \fi
384
385 \gdef\proofSymbol{\openbox}
386
387 \newcommand{\proofname}{Proof}
388
389 \newenvironment{proof}[1][\proofname]{
390 \th@proof
391 \def\theorem@headerfont{\itshape}%
392 \normalfont
393 \theoremsymbol{\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@theoremmarks}{%
402 \ifsetendmark%
403 \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
404 \setendmarkfalse%
405 \fi%
406 }{}%
407 \LWR@origendtheorem%
408 \ifbool{LWR@theoremmarks}{\global\setendmarktrue}{}%
409 \endBlockClass%
410 }

411 \gdef\NoEndMark{\global\setendmarkfalse}

```

Redefined to reuse the float mechanism to add list-of-theorem links:

`\thm@thmline {<1: printed type>} {<2: #>} {<3: optional>} {<4: page>}`

```
412 \renewcommand{\thm@@thmline@noname}[4]{%
413 \hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
414 }
415
416 \renewcommand{\thm@@thmline@name}[4]{%
417 \hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
418 }
```

This was redefined by `ntheorem` when loaded, so it is now redefined for `lwarp`:

```
419 \def\thm@@thmline{\thm@@thmline@name}
```

Patch for CSS:

```
420 \def\listtheorems#1{
421 \LWR@html@elementclass{nav}{lothm}%
422 \begingroup
423 \c@tocdepth=-2%
424 \def\thm@list{#1}\thm@processlist
425 \endgroup
426 \LWR@html@elementclassend{nav}{lothm}%
427 }
```

Proof QED symbol:

```
428
429 \newcommand{\qed}{\qquad\the\qedsymbol}
430
431 \AtBeginDocument{
432 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
433 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
434 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
435 }
```

`\thref {<label>}`

```
436 \newcommand*{\thref}[1]{\cref{#1}}
```

Package 64

lwarp-pagenote.sty

136 Pagenote

Pkg **pagenote** pagenote works as-is.

It is only included as an `lwarp-pagenote.sty` file because past versions of `lwarp` used `pagenote` to emulate footnotes, and so the file may exist on current installations, and should be over-written by this newer version.

for HTML output: 1 \LWR@ProvidesPackagePass{pagenote}

Package 65

lwarp-parskip.sty

137 Parskip

Pkg `parskip` `parskip` is ignored.
for **HTML output**: Discard all options for `lwarp-parskip`.
1 `\LWR@ProvidesPackageDrop{parskip}`

Package 66

lwarp-placeins.sty

138 Placeins

Pkg `placeins` `placeins` is not used during HTML conversion.
Discard all options for `lwarp-placeins`:
for **HTML output**: 1 `\LWR@ProvidesPackageDrop{placeins}`
2 `\newcommand*{\FloatBarrier}{}`

Package 67

lwarp-ragged2e.sty

139 Ragged2e

Pkg ragged2e ragged2e is not used during HTML conversion.

Discard all options for lwarp-ragged2e:

```
for HTML output: 1 \LWR@ProvidesPackageDrop{ragged2e}

2 \newcommand*{\Centering}{\centering}
3 \newcommand*{\RaggedLeft}{\raggedleft}
4 \newcommand*{\RaggedRight}{\raggedright}
5 \newcommand*{\justifying}{\}
6 \newlength{\CenteringLeftskip}
7 \newlength{\RaggedLeftLeftskip}
8 \newlength{\RaggedRightLeftskip}
9 \newlength{\CenteringRightskip}
10 \newlength{\RaggedLeftRightskip}
11 \newlength{\RaggedRightRightskip}
12 \newlength{\CenteringParfillskip}
13 \newlength{\RaggedLeftParfillskip}
14 \newlength{\RaggedRightParfillskip}
15 \newlength{\JustifyingParfillskip}
16 \newlength{\CenteringParindent}
17 \newlength{\RaggedLeftParindent}
18 \newlength{\RaggedRightParindent}
19 \newlength{\JustifyingParindent}
20 \newenvironment*{Center}{\center}{\endcenter}
21 \newenvironment*{FlushLeft}{\flushleft}{\endflushleft}
22 \newenvironment*{FlushRight}{\flushright}{\endflushright}
23 \newenvironment*{justify}{\justifying}{\endjustifying}
```

Package 68

lwarp-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 69

lwarp-setspace.sty

141 Setspace

Pkg **setspace** setspace is not used during HTML conversion.

Discard all options for lwarp-setspace:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{setspace}
2
3 \newcommand*{\setstretch}[1]{}
4 \newcommand*{\SetSinglespace}[1]{}
5 \newcommand*{\singlespacing}{}
6 \newcommand*{\onehalfspacing}{}
7 \newcommand*{\doublespacing}{}
8
9 \newenvironment*{singlespace}
10 {
11 \LWR@forcenewpage
12 \BlockClass{singlespace}
13 }
14 {\endBlockClass}
15
16 \newenvironment*{singlespace*}
17 {
18 \LWR@forcenewpage
19 \BlockClass{singlespace}
20 }
21 {\endBlockClass}
22
23 \newenvironment*{spacing}[1]{
24
25 }{
26
27 }
28
29 \newenvironment*{onehalfspace}
30 {
31 \LWR@forcenewpage
32 \BlockClass{onehalfspace}
33 }
34 {\endBlockClass}
35
36 \newenvironment*{doublespace}
37 {
```

```
38 \LWR@forcenewpage
39 \BlockClass{doublespace}
40 }
41 {\endBlockClass}
```

clearpage

Package 70

lwarp-showidx.sty

142 Showidx

Pkg showidx showidx is ignored.

for HTML output: Discard all options for lwarp-showidx:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

Package 71

lwarp-showkeys.sty

143 Showkeys

Pkg showkeys showkeys is ignored.

for HTML output: Discard all options for lwarp-showkeys:

```
1 \LWR@ProvidesPackageDrop{showkeys}
```

```
2 \NewDocumentCommand{\showkeys}{s}{}
```


Package 72

lwarp-sidecap.sty

144 Sidecap

Pkg sidecap sidecap is nullified.

for HTML output: Discard all options for lwarp-sidecap.

```
1 \LWR@ProvidesPackageDrop{sidecap}
```

See:

<http://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment>
regarding the creation of starred environments with xparse.

```
2 \NewDocumentEnvironment{SCtable}{soo}
3 {\IfValueTF{#3}{\table[#3]}\table}}
4 {\endtable}
5
6 \ExplSyntaxOn
7 \cs_new:cpn {SCtable*} {\SCtable*}
8 \cs_new_eq:cN {endSCtable*} \endSCtable
9 \ExplSyntaxOff
10
11
12 \NewDocumentEnvironment{SCfigure}{soo}
13 {\IfValueTF{#3}{\figure[#3]}\figure}}
14 {\endfigure}
15
16 \ExplSyntaxOn
17 \cs_new:cpn {SCfigure*} {\SCfigure*}
18 \cs_new_eq:cN {endSCfigure*} \endSCfigure
19 \ExplSyntaxOff
20
21
22 \newenvironment*{wide}{}{}
```

Package 73

lwarp-sidenotes.sty

145 Sidenotes

(Based on original code by ANDY THOMAS, OLIVER SCHEBAUM.)

Pkg sidenotes Patched for lwarp.

for HTML output: Load the original package:

```
1 \LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with lwarp:

Stop paragraph handling while creating the caption:

```
2 \RenewDocumentCommand \sidecaption {s o o m}
3 {
4   \LWR@stoppars
5   \captionsetup{style=sidecaption}
6   \IfBooleanTF{#1}
7   { % starred
8     \IfNoValueOrEmptyTF{#2}
9     {\marginnote{\caption*{#4}}}
10    {\marginnote{\caption*{#4}}[#2]}
11  }
12  { % unstarred
13    \IfNoValueOrEmptyTF{#2}
14    {\def\@sidenotes@sidecaption@tof{#4}}
15    {\def\@sidenotes@sidecaption@tof{#2}}
16    \IfNoValueOrEmptyTF{#3}
17    {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}}
18    {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}[#3]}
19  }
20 \LWR@startpars
21 }
```

Borrowed from the lwarp version of keyfloat:

```
22 \NewDocumentEnvironment{KFLT sidenotes @marginfloat}{0{-1.2ex} m}
23 {% start
24 \LWR@maybeinthisfloat%
25 \LWR@forcenewpage
26 \LWR@stoppars%
27 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}}{}
```

```

28 \LWR@startpars%
29 \captionsetup{type=#2}%
30 }
31 {
32 \LWR@htmldivclassend{div}
33 }
34
35 \RenewDocumentEnvironment{marginfigure}{o}
36   {\begin{KFLTsidenotes@marginfloat}{figure}}
37   {\end{KFLTsidenotes@marginfloat}}
38
39 \RenewDocumentEnvironment{margintable}{o}
40   {\begin{KFLTsidenotes@marginfloat}{table}}
41   {\end{KFLTsidenotes@marginfloat}}

```

The following were changed by sidenotes, and now are reset back to their lwarp-supported originals:

Restoring the definition from the L^AT_EX 2_ε `article.cls` source:

```

42 \renewenvironment{figure*}
43     {\@dblfloat{figure}}
44     {\end@dblfloat}
45
46 \renewenvironment{table*}
47     {\@dblfloat{table}}
48     {\end@dblfloat}

```

Package 74

lwarp-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{}{}
```

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 \ifcsempy{#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}{uline}{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 75

lwarp-subfig.sty

147 Subfig

(Based on original code by STEVEN DOUGLAS COCHRAN.)

Pkg subfig subfig is supported and patched by lwarp.

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

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

for HTML output: Accept all options for lwarp-subfig:

```

1 \LWR@ProvidesPackagePass{subfig}

\sf@@@subfloat {<1 type>} [<2 lof entry>] [<3 caption>] {<4 contents>}

The outer minipage allows side-by-side subfloats with \hfill between.

2 \long\def\sf@@@subfloat#1[#2][#3]#4{%
3 \begin{minipage}{\linewidth}% new
4 \LWR@stoppars% new
5   \ifundefined{FBsc@max}{}%
6     {\FB@readaux{\let\FBsuboheight\relax}}%
7   \@tempcnta=\@ne
8   \if@minipage
9     \@tempcnta=\z@
10  \else\ifdim \lastskip=\z@ \else
11    \@tempcnta=\tw@
12  \fi\fi
13  \ifmaincaptiontop
14    \sf@top=\sf@nearskip
15    \sf@bottom=\sf@farskip
16  \else
17    \sf@top=\sf@farskip
18    \sf@bottom=\sf@nearskip
19  \fi
20  \leavevmode
21 \setbox\@tempboxa \hbox{#4}%
22   \@tempdima=\wd\@tempboxa

```

```

23 \ifundefined{FBsc@max}{}%
24 {\global\advance\Xhsize-\wd\@tempboxa
25 \dimen@=\ht\@tempboxa
26 \advance\dimen@\dp\@tempboxa
27 \ifdim\dimen@>\FBsc@max
28 \global\FBsc@max\dimen@
29 \fi}%
30 \vtop\bgroup
31 \vbox\bgroup
32 \ifcase\@tempcnta
33 \@minipagefalse
34 \or
35 \vskip\sf@top
36 \or
37 \ifdim \lastskip=\z@ \else
38 \@tempskipb\sf@top\relax\@xaddvskip
39 \fi
40 \fi
41 \sf@ifpositiontop{%
42 \ifx \@empty#3\relax \else
43 \sf@subcaption{#1}{#2}{#3}%
44 \vskip\sf@capskip
45 \vskip\sf@captopadj
46 \fi\egroup
47 \hrule 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\FBsuboheight\relax
60 % \box\@tempboxa
61 #4
62 \else
63 % \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
64 #4
65 \fi}%
66 \LWR@stoppars% new
67 \egroup
68 \ifx \@empty#3\relax \else
69 \vskip\sf@capskip
70 \hrule 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}% new
85 \LWR@startpars% new
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% new
90 \ifx \relax#2\relax \else
91 \bgroup
92 \let\label=\@gobble
93 \let\protect=\string
94 \def\@subcaplabel{%
95 \caption@lstfmt{\@nameuse{p#1}}{\@nameuse{the#1}}}%
96 \sf@updatecaptionlist{#1}{#2}{\the\value{\@capttype}}{\the\value{#1}}%
97 \egroup
98 \fi
99 \bgroup
100 \ifx \relax#3\relax
101 \let\captionlabelsep=\relax
102 \fi
103 % \setbox0\vbox{%
104 % \hb@xt@\the\@tempdima{%
105 %
106 % % \hss
107 % % \parbox[t]{\the\@tempdima}{%
108 % % \caption@make
109 % % {\@nameuse{sub\@capttype name}}%
110 % % {\@nameuse{thesub\@capttype}}%
111 % % {\#3}
112 % % }%
113 % % \hss
114 % % }
115 % %}%
116 \@ifundefined{FBsc@max}%
117 % {\box0}%
118 {
119 % \parbox[t]{\the\@tempdima}{%

```



```

120 \LWR@traceinfo{sfsubcap B1}% new
121 \LWR@htmlblocktag{figcaption}% new
122 \caption@make
123 {\@nameuse{sub\@capttype name}}}%
124 {\@nameuse{thesub\@capttype}}}%
125 {#3}
126 \LWR@htmlblocktag{/figcaption}% new
127 \LWR@traceinfo{sfsubcap B2}% new
128 % }%
129 }%
130     {\dimen@ \ht0%
131     \advance \dimen@ \dp0%
132     \ifdim \dimen@ > \FBsc@max
133     \global \FBsc@max \dimen@
134     \fi
135     \FB@readaux{\let \FBsubcheight \relax}%
136     \ifx \FBsubcheight \relax
137     \def \next{
138 % \parbox[t]{\the \@tempdima}
139 }%
140     \else
141     \def \next{
142 % \parbox[t][\FBsubcheight][t]{\the \@tempdima}
143 }%
144     \fi
145     \vbox{%
146 % \hb@xt@\the \@tempdima{%
147
148 % \hss
149 % \next{%
150 \LWR@traceinfo{sfsubcap C1}% new
151     \caption@make
152     {\@nameuse{sub\@capttype name}}}%
153     {\@nameuse{thesub\@capttype}}}%
154     {#3}
155 \LWR@traceinfo{sfsubcap C1}% new
156 % }%
157 % \hss
158
159 % }
160 }
161 }%
162 \egroup
163 \LWR@startpars% new
164 }

\caption@@@make {\caption label} {\caption text}

165 \renewcommand \caption@@@make[2]{%
166 \LWR@startpars% new

```

```

167 \sbox\@tempboxa{#1}%
168 \ifdim\wd\@tempboxa=\z@
169 \let\caption@lsep\relax
170 \fi
171 \caption@ifempty{#2}{%
172 \let\caption@lsep\@empty
173 \let\caption@tfmt\@firstofone
174 }%
175 % \@setpar{\@par\caption@@par}\caption@@par
176 \renewcommand{\@par}{\LWR@closeparagraph\LWR@orig@@par}% new
177 \caption@applyfont
178 \caption@fmt
179 {\ifcaption@star\else
180 \begingroup
181 \caption@labelfont
182 #1%
183 \endgroup
184 \fi}%
185 {\ifcaption@star\else
186 \begingroup
187 \caption@iflf\caption@labelfont
188 \relax\caption@lsep
189 \endgroup
190 \fi}%
191 {{\caption@textfont
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@newsfloat\@newsfloat
217
218 \def\@newsfloat[#1]#2{%
219 \LWR@orig@newsfloat[#1]{#2}%
220 \renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}
221 }
```

Pre-defined for figures and tables:

```
222 \renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2}}
223 \renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}
224 % \def\subfigure{\subfloat}
225 % \def\subtable{\subfloat}
```

Package 76

lwarp-tabularx.sty

148 Tabularx

Pkg `tabularx` `tabularx` is emulated by `lwarp`.

for **HTML output**: Discard all options for `lwarp-tabularx`:

```
1 \LWR@ProvidesPackageDrop{tabularx}

2 \NewDocumentEnvironment{tabularx}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabularx*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}
```

Package 77

lwarp-tabulary.sty

149 Tabulary

Pkg tabulary tabulary is emulated by lwarp.

for HTML output: Discard all options for lwarp-tabulary.

Column types L, C, R, and J are emulated by lwarp core code.

```
1 \LWR@ProvidesPackageDrop{tabulary}

2 \NewDocumentEnvironment{tabulary}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabulary*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}
9
10 \newdimen\tymin
11 \newdimen\tymax
12 \def\tyformat{}
```

Package 78

lwarp-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 79

lwarp-theorem.sty

151 Theorem

(Based on original code by FRANK MITTELBAACH.)

Pkg theorem theorem is patched for use by lwarp.

CSS styling of theorems and proofs:

Theorem: <div> of class theorembody<theoremstyle>

Theorem Header: of class theoremheader

where <theoremstyle> is plain, break, etc.

for HTML output: 1 \LWR@ProvidesPackagePass{theorem}

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
3 \gdef\theoremstyle#1{%
4   \@ifundefined{th#1}{\@warning
5     {Unknown theoremstyle ‘#1’. Using ‘plain’}%
6     \theorem@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% new
8   }%
9   {%
10    \theorem@style{#1}%
11    \renewcommand{\LWR@newtheoremstyle}{#1}% new
12  }%
13  \begingroup
14    \csname th@the\theorem@style \endcsname
15  \endgroup}
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```
16 \gdef\xnthm#1#2[#3]{%
17   \expandafter\@ifdefinable\csname #1\endcsname
```

```

18  {%
19  \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
20  \@definecounter{#1}\@newctr{#1}[#3]%
21  \expandafter\xdef\csname the#1\endcsname
22    {\expandafter \noexpand \csname the#3\endcsname
23      \@thmcountersep \@thmcounter{#1}}%
24  \def\@tempa{\global\@namedef{#1}}%
25  \expandafter \@tempa \expandafter{%
26    \csname th@the \theoremstyle
27      \expandafter \endcsname \the \theorem@bodyfont
28      \@thm{#1}{#2}}%
29  \global \expandafter \let \csname end#1\endcsname \@endtheorem
30  \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
31  }}
32
33 \gdef\@ynthm#1#2{%
34 \expandafter\@ifdefinable\csname #1\endcsname
35   {
36 \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
37 \@definecounter{#1}%
38   \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
39   \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
40   \expandafter{\csname th@the \theoremstyle \expandafter
41     \endcsname \the\theorem@bodyfont \@thm{#1}{#2}}%
42   \global \expandafter \let \csname end#1\endcsname \@endtheorem
43   \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% 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 \theoremstyle \expandafter
57     \endcsname \the\theorem@bodyfont \@thm{#2}{#3}}%
58   \global \expandafter \let \csname end#1\endcsname \@endtheorem
59   \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
60 }%
61 \fi}

```

The following are patched for CSS.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc. They are gathered together here.

Each theorem is encased in a BlockClass environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader`.

```

62 \gdef\th@plain{%\normalfont\itshape
63   \def\@begintheorem##1##2{%
64     \LWR@forcenewpage% new
65     \BlockClass{theorembody\LWR@thisthmstyle}% new
66       \item[\hskip\labelsep
67 \InlineClass{theoremheader}{##1\ ##2}
68 ]}%
69 \def\@opargbegintheorem##1##2##3{%
70   \LWR@forcenewpage% new
71   \BlockClass{theorembody\LWR@thisthmstyle}% new
72     \item[\hskip\labelsep
73 \InlineClass{theoremheader}{##1\ ##2\ (##3)}
74 ]}
75 }
76
77 \gdef\th@break{%\normalfont\slshape
78   \def\@begintheorem##1##2{%
79     \LWR@forcenewpage% new
80     \BlockClass{theorembody\LWR@thisthmstyle}% new
81     \item[\hskip \labelsep
82 \InlineClass{theoremheader}{##1\ ##2}\newline%
83 ]}%
84 \def\@opargbegintheorem##1##2##3{%
85   \LWR@forcenewpage% new
86   \BlockClass{theorembody\LWR@thisthmstyle}% new
87     \item[\hskip \labelsep
88 \InlineClass{theoremheader}{##1\ ##2\ (##3)}\newline
89 ]}
90 }
91
92 \gdef\th@marginbreak{%\normalfont\slshape
93   \def\@begintheorem##1##2{
94     \LWR@forcenewpage% new
95     \BlockClass{theorembody\LWR@thisthmstyle}% new
96     \item[\hskip\labelsep %
97 \InlineClass{theoremheader}{##2 \quad ##1}\newline
98 ]}%
99 \def\@opargbegintheorem##1##2##3{%
100   \LWR@forcenewpage% new
101   \BlockClass{theorembody\LWR@thisthmstyle}% new
102   \item[\hskip\labelsep %
103 \InlineClass{theoremheader}{##2 \quad ##1\ %
104 (##3)}\newline
105 ]}
106 }
107
```

```

108 \gdef\th@changebreak{%\normalfont\slshape
109 \def\@begintheorem##1##2{
110 \LWR@forcenewpage% new
111 \BlockClass{theorembody\LWR@thisthmstyle}% new
112 \item[\hskip\labelsep
113 \InlineClass{theoremheader}{##2\ ##1}\newline
114 ]}%
115 \def\@opargbegintheorem##1##2##3{%
116 \LWR@forcenewpage% new
117 \BlockClass{theorembody\LWR@thisthmstyle}% new
118 \item[\hskip\labelsep
119 \InlineClass{theoremheader}{ ##2\ ##1\ %
120 (##3)}\newline
121 ]}
122 }
123
124 \gdef\th@change{%\normalfont\slshape
125 \def\@begintheorem##1##2{
126 \LWR@forcenewpage% new
127 \BlockClass{theorembody\LWR@thisthmstyle}% new
128 \item[\hskip\labelsep
129 \InlineClass{theoremheader}{##2\ ##1}
130 ]}%
131 \def\@opargbegintheorem##1##2##3{%
132 \LWR@forcenewpage% new
133 \BlockClass{theorembody\LWR@thisthmstyle}% new
134 \item[\hskip\labelsep
135 \InlineClass{theoremheader}{##2\ ##1\ (##3)}
136 ]}
137 }
138
139 \gdef\th@margin{%\normalfont\slshape
140 \def\@begintheorem##1##2{
141 \LWR@forcenewpage% new
142 \BlockClass{theorembody\LWR@thisthmstyle}% new
143 \item[\hskip\labelsep
144 \InlineClass{theoremheader}{##2 \quad ##1}
145 ]}%
146 \def\@opargbegintheorem##1##2##3{%
147 \LWR@forcenewpage% new
148 \BlockClass{theorembody\LWR@thisthmstyle}% new
149 \item[\hskip\labelsep
150 \InlineClass{theoremheader}{##2 \quad ##1\ (##3)}
151 ]}
152 }

```

Patched for CSS:

```

153 \gdef\@endtheorem{\endBlockClass\endtrivlist}

```

Package 80

lwarp-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]{\textsuperscript{#1}}
```

Package 81

lwarp-tikz.sty

153 Tikz

Pkg **tikz** tikz is supported.

Accept all options for lwarp-tikz:

```
1 \LWR@ProvidesPackagePass{tikz}
```

catcodes lwarp changes the catcode of `$` for its own use. The Tikz `babel` library temporarily changes catcodes back to normal for Tikz's use. `tikz` v3.0.0 introduced the `babel` library which handles catcode changes. For older versions, lwarp must change `$`'s catcode itself.

for HTML output: `2 \begin{warpHTML}`

```
3 \newboolean{LWR@tikzbabel}
```

```
4
```

```
5 \@ifpackagelater{tikz}{2013/12/20}% Test for Tikz version v3.0.0
```

```
6 {\usetikzlibrary{babel}\booltrue{LWR@tikzbabel}}
```

```
7 {\boolfalse{LWR@tikzbabel}}
```

Env **tikzpicture** `tikzpicture` environment is enclosed inside a `\lateximage`. May be used as-is, and its contents will be converted to an image.

```
8 \BeforeBeginEnvironment{tikzpicture}{%
```

```
9 \lateximage%
```

```
10 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
```

```
11 {}%
```

```
12 {\catcode'\$=3} % dollar sign is math shift
```

```
13 }
```

```
14
```

```
15 \AfterEndEnvironment{tikzpicture}{%
```

```
16 \endlateximage%
```

```
17 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
```

```
18 {}%
```

```
19 {\catcode'\$=\active}%
```

```
20 }
```

```
21 \end{warpHTML}
```

Package 82

lwarp-titles.sty

154 Titles

Pkg titles titles is loaded and used by lwarp during HTML output. All user options and macros are ignored and disabled.

Discard all options for lwarp-titles:

for HTML output: 1 \LWR@ProvidesPackageDrop{titles}

\pagestyle and \thispagestyle are already disabled in the lwarp code.

```

2 \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{\settitledmarks}{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*{\bottitledmarks}{}
16 \renewcommand*{\toptitledmarks}{}
17 \renewcommand*{\firsttitledmarks}{}
18 \renewcommand*{\nexttoptitledmarks}{}
19 \renewcommand*{\outertitledmarks}{}
20 \renewcommand*{\innertitledmarks}{}

21 \RenewDocumentCommand{\newtitledmark}{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{\nexttoextramarks}[1]{}
34 \newcommand{\outerextramarks}[1]{}
35 \newcommand{\innerextramarks}[1]{}

```

Package 83

lwarp-titlesec.sty

155 Titlesec

Pkg titlesec titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titlesec:

```
for HTML output: 1 \LWR@ProvidesPackageDrop{titlesec}

2 \newcommand*{\titlelabel}[1]{}

3 \newcommand\titleformat{%
4   \@ifstar{\ttl@format@s}%
5     {\ttl@format@i}}
6 \newcommand{\ttl@format@s}[1]{}
7 \NewDocumentCommand{\ttl@format@i}{m o m m m o}{}

8 \@ifundefined{chapapp}{\let\@chapapp\chaptername}{}
9 \newcommand\chaptertitlename{\@chapapp}

10 \NewDocumentCommand{\titlespacing}{s m m m m o}{}

11 \newcommand*{\filright}{}
12 \newcommand*{\filcenter}{}
13 \newcommand*{\filleft}{}
14 \newcommand*{\fillast}{}
15 \newcommand*{\filinner}{}
16 \newcommand*{\filouter}{}

17 \newcommand\wordsep{\fontdimen\tw@\font \@plus
18   \fontdimen\thr@\font \@minus \fontdimen4\font}

19 \NewDocumentCommand{\titleline}{s o m}{}

20 \providecommand*{\titlerule}{\@ifstar{\ttl@row}{\ttl@rule}}
21 \newcommand*{\ttl@rule}[1] [] {}
22 \newcommand*{\ttl@row}[2] [] {}

23 \newcommand{\iftitlemeasuring}[2] {#2}

24 \newcommand{\assignpagestyle}[2] {#2}

25 \NewDocumentCommand{\titleclass}{m o m o}
```

Package 84

lwarp-titletoc.sty

156 Titletoc

Pkg titletoc titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titletoc:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{titletoc}

2 \NewDocumentCommand{\dottedcontents}{m o m m m}{\}

3 \newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
4 \NewDocumentCommand{\ttl@tcstar}{m o m m m o o}{\}
5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m 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 85

lwarp-titling.sty

157 Titling

`Pkg titling` titling is used by lwarp. The following patches are not needed by lwarp, but are required if the user requests titling.

lwarp uses page notes for footnotes, so the various titling footnote restyling commands have no effect.

Pass all options to lwarp-titling:

for HTML output: `1 \LWR@ProvidesPackagePass{titling}`

Patch `\@bsmtitleempty`:

```
2 \let\LWR@orig@bsmtitleempty\@bsmtitleempty
3 \renewcommand*{\@bsmtitleempty}{%
4 \LWR@orig@bsmtitleempty%
5 \global\let\published\relax%
6 \global\let\subtitle\relax%
7 }
```

Patch `\keepthetitle`:

```
8 \let\LWR@origkeepthetitle\keepthetitle
9 \renewcommand*{\keepthetitle}{%
10 \LWR@orig@keepthetitle%
11 \global\let\@published\@empty%
12 \global\let\@subtitle\@empty%
13 }
```

Patch `\killtitle`:

```
14 \let\LWR@origkilltitle\killtitle
15 \renewcommand*{\killtitle}{%
16 \LWR@orig@killtitle%
17 \global\let\thepublished\relax%
18 \global\let\thesubtitle\relax%
19 }
```

Package 86

lwarp-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 `lwarp-tocloft`:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{tocloft}

2 \newcommand{\tocloftpagestyle}[1]{}

3 \newcommand*{\cftmarktoc}{}
4 \newcommand*{\cfttoctitlefont}{}
5 \newcommand*{\cftaftertoctitle}{}

6 \newlength{\cftbeforetoctitleskip}
7 \newlength{\cftaftertoctitleskip}

8 \newcommand*{\cftmarklof}{}
9 \newcommand*{\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{\cftpartincent}
29 \newlength{\cftpnumwidth}
30 \newcommand*{\cftpfont}{}
31 \newcommand*{\cftpapresnum}{}
32 \newcommand*{\cftpataftersnum}{}
33 \newcommand*{\cftpataftersnumb}{}
34 \newcommand*{\cftpleader}{}
35 \newcommand*{\cftpdotsep}{1}
36 \newcommand*{\cftpfont}{}
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*{\cftchapfont}{}
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*{\cftsecfont}{}
59 \newcommand*{\cftsecafterpnum}{}

60 \newlength{\cftbeforesubsecskip}
61 \newlength{\cftsubsecindent}
62 \newlength{\cftsubsecnumwidth}
63 \newcommand*{\cftsubsecfont}{}
64 \newcommand*{\cftsubsecpresnum}{}
65 \newcommand*{\cftsubsecaftersnum}{}
66 \newcommand*{\cftsubsecaftersnumb}{}
67 \newcommand*{\cftsubsecleader}{}
68 \newcommand*{\cftsubsecdotsep}{1}
```

```

69 \newcommand*{\cftsubsecpagefont}{}
70 \newcommand*{\cftsubsecafterpnum}{}

71 \newlength{\cftbeforesubsubsecskip}
72 \newlength{\cftsubsubsecindent}
73 \newlength{\cftsubsubsecnumwidth}
74 \newcommand*{\cftsubsubsecfont}{}
75 \newcommand*{\cftsubsubsecpresnum}{}
76 \newcommand*{\cftsubsubsecaftersnum}{}
77 \newcommand*{\cftsubsubsecaftersnumb}{}
78 \newcommand*{\cftsubsubsecleader}{}
79 \newcommand*{\cftsubsubsecdotsep}{1}
80 \newcommand*{\cftsubsubsecpagefont}{}
81 \newcommand*{\cftsubsubsecafterpnum}{}

82 \newlength{\cftbeforeparaskip}
83 \newlength{\cftparaindent}
84 \newlength{\cftparanumwidth}
85 \newcommand*{\cftparafont}{}
86 \newcommand*{\cftparapresnum}{}
87 \newcommand*{\cftparaftersnum}{}
88 \newcommand*{\cftparaftersnumb}{}
89 \newcommand*{\cftparaleader}{}
90 \newcommand*{\cftparadotsep}{1}
91 \newcommand*{\cftparapagefont}{}
92 \newcommand*{\cftparaafterpnum}{}

93 \newlength{\cftbeforesubparaskip}
94 \newlength{\cftsubparaindent}
95 \newlength{\cftsubparanumwidth}
96 \newcommand*{\cftsubparafont}{}
97 \newcommand*{\cftsubparapresnum}{}
98 \newcommand*{\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 \@namedef{listof#2}{\listof{#2}{#4}}
157 \@namedef{#2depth}{1}
158 \expandafter\newlength\csuse{cftbefore#2skip}
159 \expandafter\newlength\csuse{cft#2indent}
160 \expandafter\newlength\csuse{cft#2numwidth}
161 \@namedef{cft#2font}{}
162 \@namedef{cft#2presnum}{}
163 \@namedef{cft#2aftersnum}{}
164 \@namedef{cft#2aftersnumb}{}
165 \@namedef{cft#2leader}{}
166 \@namedef{cft#2dotsep}{1}
167 \@namedef{cft#2pagefont}{}
168 \@namedef{cft#2afterpnum}{}
169 }

```

\cftchapterprecis from tocloft:

```

170 \newcommand{\cftchapterprecis}[1]{%
171   \cftchapterprecishere{#1}
172   \cftchapterprecistoc{#1}}
173 \newcommand{\cftchapterprecishere}[1]{%
174   \begin{quote}\textit{#1}\end{quote}}
175 \newcommand{\cftchapterprecistoc}[1]{
176   \addtocontents{toc}{%
177     {
178       \protect\begin{quote}#1\protect\end{quote}}
179   }
180 }

```

Package 87

lwarp-trivfloat.sty

159 Trivfloat

Pkg `trivfloat` `trivfloat` is forced to use the built-in `lwarp` emulation for floats.

Discard all options for `lwarp-trivfloat`. This tells `trivfloat` not to use `floatrow` or `memoir`.

for HTML output: `1 \LWR@ProvidesPackageDrop{trivfloat}`
`2 \LWR@origRequirePackage{trivfloat}`

for HTML & PRINT: `3 \begin{warpall}`

To create a new float type and change its name:

```
\trivfloat{example}
\renewcommand{\exemplename}{Example Name}
\crefname{example}{example}{examples}
\Crefname{example}{Example}{Examples}
```

`4 \end{warpall}`

`\tfl@chapter@fix` Nullified at the beginning of the document. Is used by `trivfloat` to correct float chapter numbers, but is not needed for `lwarp`.

for HTML output: `5 \begin{warpHTML}`
`6 \AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}`
`7 \end{warpHTML}`

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 88

lwarp-ulem.sty

160 Ulem

(Based on original code by DONALD ARSENEAU.)

Pkg ulem Emulated.

for HTML output: Original lwarp definitions:

```
1 \let\LWR@ulemorigemph\emph
2 \let\LWR@ulemorigtextbf\textbf
```

Basic markup commands, using CSS:

```
3 \NewDocumentCommand{\uline}{+m}{%
4 \InlineClass{uline}{#1}%
5 }
6
7 \NewDocumentCommand{\uuline}{+m}{%
8 \InlineClass{uuline}{#1}%
9 }
10
11 \NewDocumentCommand{\uwave}{+m}{%
12 \InlineClass{uwave}{#1}%
13 }
14
15 \NewDocumentCommand{\sout}{+m}{%
16 \InlineClass{sout}{#1}%
17 }
18
19 \NewDocumentCommand{\xout}{+m}{%
20 \InlineClass{xout}{#1}%
21 }
22
23 \NewDocumentCommand{\dashuline}{+m}{%
24 \InlineClass{dashuline}{#1}%
25 }
26
27 \NewDocumentCommand{\dotuline}{+m}{%
28 \InlineClass{dotuline}{#1}%
29 }
```

Nullified parameters:

```
30 \NewDocumentCommand{\ULthickness}{-}{-}
31 \newlength{\ULdepth}
```

Nullified/emulated macros:

```
32 \NewDocumentCommand{\markoverwith}{m}{-}
33 \NewDocumentCommand{\ULon}{+m}{\uline{#1}\egroup}
```

`\useunder` only works with `\textbf`, etc, but not `\bfseries`, etc.

```
34 \NewDocumentCommand{\useunder}{m m m}{%
35 \relax%
36 \ifx\relax#3\relax\else % argumentative command
37   \def#3{#1}\MakeRobust{#3}\fi
38 }
```

Triggered by package options, also available for the users:

```
39 \newcommand*{\normalem}{\let\emph\LWR@ulemorigemph}
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 89

lwarp-verse.sty

161 Verse

(Based on original code by PETER WILSON.)

Pkg verse verse is supported and patched by lwarp.

for HTML output: Pass all options for lwarp-verse:

```
1 \LWR@ProvidesPackagePass{verse}
```

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

```
\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}
```

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

Len \leftmargini

Len \TMLvleftskip

Len \TMLleftmargini

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

Env verse The `verse` environment will be placed inside a HTML `pre`.

```
2 \AfterEndPreamble{
```

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{\@vstypeptitle}{m}{%
19   \vspace{\beforepoemtitleskip}%
20   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
21   \vspace{\afterpoemtitleskip}%
22 }
23 }{}
24
25 }
```

Package 90

lwarp-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*{\LRCornerWallPaper}[2]{}
15 \newcommand*{\ThisLRCornerWallPaper}[2]{}
16 \newcommand*{\ClearWallPaper}{}
17 \newlength{\wpXoffset}
18 \newlength{\wpYoffset}
```

Package 91

lwarp-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@htmltag{div class="marginblock" id="autofloat-\arabic{\LWR@thisfloat}"
21 style="width:\rndprintlength{\LWR@wrapwidth} ; %
22 \LWR@wrapposition"%
23 }
24 \LWR@startpars
25 }
26
27
28 \NewDocumentEnvironment{wrapfigure}{o m o m}
29 {%
30 \LWR@subwrapfigure{#2}{#4}%
31 \captionsetup{type=figure}%
32 }
33 {
34 \LWR@htmldivclassend{div}
```

```
35 }
36
37
38 \NewDocumentEnvironment{wraptable}{o m o m}
39 {%
40 \LWR@subwrapfigure{#2}{#4}%
41 \captionsetup{type=table}%
42 }
43 {
44 \LWR@htmldivclassend{div}
45 }
46
47
48 \NewDocumentEnvironment{wrapfloat}{m o m o m}
49 {%
50 \LWR@subwrapfigure{#3}{#5}%
51 \captionsetup{type=#1}%
52 }
53 {
54 \LWR@htmldivclassend{div}
55 }
56
57 \newlength{\wrapoverhang}
```

Package 92

lwarp-xcolor.sty

164 Xcolor

Pkg	xcolor	xcolor is supported by lwarp.
support		Color definitions, models, and mixing are fully supported without any changes required.
tables		Colored tables are ignored so far. Use CSS to style tables.
colored text and boxes		<code>\textcolor</code> , <code>\colorbox</code> , and <code>\fcolorbox</code> are supported.
<code>\color</code> and <code>\pagecolor</code>		<code>\color</code> and <code>\pagecolor</code> are ignored. Use CSS or <code>\textcolor</code> where possible.
for HTML output:		<pre> 1 \LWR@ProvidesPackagePass{xcolor} 2 \newcommand*{\LWR@tempcolor}{} defaulting to black. 3 \newcommand*{\LWR@currenttextcolor}{black} \LWR@colorstyle {\langle 1: styletext \rangle} {\langle 2: model \rangle} {\langle 3: color \rangle} {\langle 4: spancontents \rangle} 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: </pre>


```

8 \LWR@htmltagc{span style="#1\#\LWR@tempcolor"{} }%
9 \begin{LWR@nestspan}%

```

Prevent additional paragraph tags inside this span:

Print the contents then close the span:

```

10 #4%
11 \LWR@htmltagc{/span}%
12 \end{LWR@nestspan}%

```

For paragraph-tag handling:

```

13 \LWR@ensuredoingapar%
14 }

```

`\color` appears in the L^AT_EX PDF output, but is ignored by `pdftotext` and thus is ignored in the HTML file. Text styling by local group is not yet supported.

Each of the following macros is given a temporary name, and is `\let` to the final name once the HTML conversion starts.

`\textcolor` [*model*] {*color*} {*text*} is converted into an HTML hex color span.

```

15 \NewDocumentCommand{\LWR@textcolor}{0{named} m m}{%
16 \begingroup%
17 \LWR@colorstyle{color:}{#1}{#2}{#3}%
18 \endgroup%
19 }

```

`\pagecolor` [*model*] {*color*} is ignored. Use `\CSSFilename` instead.

```

20 \newcommand*{\LWR@pagecolor}[2][named]{ }

```

`\colorbox` [*model*] {*color*} {*text*} is converted into an HTML hex background color span.

```

21 \NewDocumentCommand{\LWR@colorbox}{0{named} m m}{%
22 \begingroup%
23 \LWR@colorstyle{background:}{#1}{#2}{#3}%
24 \endgroup%
25 }

```

`\fcolorbox` [*framemodel*] {*framecolor*} [*boxmodel*] {*boxcolor*} {*text*} is converted into a framed HTML hex background color span.

A background color of "none" creates a colored frame without a background color.

```
26 \NewDocumentCommand{\LWR@fcolorbox}{O{named} m O{named} m m}{%
27 \begingroup%
28 \ifthenelse{\equal{#4}{none}}{% no background color
29 \LWR@colorstyle{border:1px solid }{#1}{#2}{#5}%
30 }{% yes background color
31 \LWR@colorstyle{border:1px solid }{#1}{#2}%
32 {\LWR@colorstyle{background:}{#3}{#4}{#5}}%
33 }%
34 \endgroup%
35 }
```

Redirect to new definitions:

```
36 \let\textcolor\LWR@textcolor
37 \let\pagecolor\LWR@pagecolor
38 \let\colorbox\LWR@colorbox
39 \let\fcolorbox\LWR@fcolorbox
```

Package 93

lwarp-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, **lwarp** should be loaded after font-related setup. During HTML conversion, this font is used by **lwarp** to generate its initial PDF output containing HTML tags, later to be converted by **pdftotext** to a plain text file. While the text may be in any font which **pdftotext** can read, the math is directly converted into SVG images using this same user-selected font. **xfrac** below is set for the Latin Modern (lmr) font. If another font is used, it may be desirable to redefine **\xfracHTMLfontsize** with a different em size.

\sfrac [*instance*] {*num*} [*sep*] {*denom*}

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. **\scalebox** is made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a **lateximage**, no adjustments are necessary.

for HTML & PRINT: 2 \begin{warpall}

User-redefinable macro which controls the font size of the fraction.

3 \newcommand*{\xfracHTMLfontsize}{.6em}

4 \end{warpall}

for HTML output: 5 \begin{warpHTML}

font size A span for a small font, used in the numerator and denominator:

```
6 \newcommand*{\LWR@htmlsmallfontstart}{%
7 \LWR@htmltagc{span style="font-size:\xfracHTMLfontsize"}{}}%
8 \LWR@nestspan%
9 %
10 }
```

```

11
12 \newcommand*{\LWR@htmlsmallfontend}{%
13 \LWR@htmltagc{/span}%
14 \endLWR@nestspan%
15 }

```

`\scalebox` A nullified `\scalebox` command, to avoid introducing HTML scaling tags:

```

16 \NewDocumentCommand{\LWR@noscalebox}{m o m}{#3}

```

instances Instances of `xfrac` for various font choices:

Produce HTML tags for a small superscript numerator and a small (non-subscript) denominator.

Scaling is turned off so that `pdftotext` correctly reads the result.

```

17 \DeclareInstance{xfrac}{default}{text}{
18 numerator-format = {%
19 \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		Docs: Commands for a successful HTML conversion. 65
General: 2016/03/08 Initial version	1	
v0.11		Docs: Commands into a warpprint environment. 65
General: 2016/03/11	1	Docs: Newclude limitations. . . 78
Added section:		Docs: Table: Cross-referencing data structures. 264
Operating-System portability.	94	Docs: Table: Float data structures. 274
Added section: Selecting the operating system.	64	Docs: Trademarks section. . . 342
Test Suite: limages and index in README.txt	1	Docs: Troubleshooting cross-references. 86
Test Suite: MS-Windows in README.txt	1	Test Suite: Assigned cleveref name for Test Float. 1
v0.12		Test Suite: Floatrow 1
\LWR@newhtmlfile: Bugfix: TOC with numbered files.	192	v0.15
General: 2016/03/14	1	\printpublished: No div if empty. 214
Global: Uses \p@(type) in float captions.	1	\printsubtitle: No div if empty. 215
Test Suite: Sub-figures	1	General: 2016/04/06 1
v0.13		Added 372
\CaptionSeparator: Fix for newer babel package.	276	Ampersand (&): Fixed handling when passed as an argument. 239
\LWR@LwarpStart: \up and \fup .	205	Docs: Added warning icons for items needing special attention. 91
\published: Default to empty published.	211	Docs: Clarify print/HTML output. 64
\subtitle: Default to empty subtitle.	212	Docs: Moved the supported functions table to the introduction. 27
General: 2016/03/24	1	Files: lwarp_formal.css added. . 1
Removed package: subfig	1	Fix: steps counter 371
Test Suite: Ordinals, Subcaption	1	Fixed & handling. 369
tikzpicture: Fix dollar-redefined bug for newer package.	452	Test Suite: test_suite_formal.css file added. 1
v0.14		v0.16
\LWR@htmlsectionfilename: Fix: Links to home page.	167	General: 2016/04/11 1
General: 2016/03/31	1	
floatrow: Added.	367	

<code>\tittingpage</code> : Improved print-output spacing.	216	<code>\LWR@subhyperrefclass</code> : Improved HTML output linebreaks.	271
Added XeLaTeX, LuaLaTeX support.	101	<code>\LWR@subinlineimage</code> : Surpress extra space.	272
Docs: Font and UTF-8 support.	61	General: 2016/05/19	1
Docs: Moved location of <code>\usepackage{lwarp}</code>	62	File: lwarp.css: Improved TOC outline display.	1
Docs: Text not converting.	86	Files: lwarp.css and lwarp_formal.css: Improved responsive design.	1
Fix: amsmath options clash	105	Microtype disabled during HTML generation	102
Fix: newtxmath compatibility.	105	PDF Unicode input characters.	92
Lwarp no longer selects fonts.	61, 101	Test Suite: Verse package	1
Removed package: suffix	1	<code>lateximage</code> : pdfcrop: --hires added.	305
Test Suite: Improved tittingpage.	216	Reorganize <code>\HomeHTMLFilename</code> logic.	305
Test Suite: Lwarp no longer selects fonts.	1	Surpress extra space.	305
Test Suite: Supports XeLaTeX, LuaLaTeX.	1	<code>verse</code> : Supports verse, memoir packages.	467
<code>xfrac</code> : Adjusted for the use of any font:	476	<code>minipage</code> : Fix: <code>\linewidth</code> , <code>\textwidth</code> , <code>\textheight</code> inside a minipage.	326
v0.17		v0.19	
<code>\LWR@htmlsectionfilename</code> : Fix: Links when entire doc is one HTML page.	167	<code>\HTMLFilename</code> : Docs: Escape filename underscores.	166
General: 2016/04/14	1	<code>\HomeHTMLFilename</code> : Docs: Escape filename underscores.	166
<code>mdframed</code> : Added.	401	<code>\LWR@LwarpStart</code> : Enabled <code>\\</code> equal to <code>\newline</code>	204
Test Suite: Fix: Print-version front-matter page numbers.	1	<code>\LWR@doubledollar</code> : MathJax support.	292
Test Suite: Mdfamed	1	<code>\LWR@filestart</code> : lwarp_mathjax.txt loaded.	202
v0.18		<code>\LWR@hspace</code> : Fix: <code>\hspace</code> length computations.	335
<code>\LWR@hspace</code> : <code>\hspace</code> supported.	335	<code>\LWR@minipagestartpars</code> : Surpresses paragraph tags between minipages.	333
<code>\LWR@includegraphicsb</code> : Add: svgz file extension.	315	<code>\LWR@singledollar</code> : MathJax support.	292
em, ex, %, px dimensions preserved.	315	<code>\LWR@tabledatamultirowtag</code> : Added optional args.	259
Fix: <code>\linewidth</code> , <code>\textwidth</code> , <code>\textheight</code> inside a minipage.	315	<code>\LateximageFontSizeName</code> : Add: User-adjustable math/lateximage font size.	304
Improved HTML output linebreaks.	315	<code>\minipagefullwidth</code> : Added: No width tag for the next minipage in HTML.	325
<code>\LWR@myshorttoc</code> : Reorganize <code>\HomeHTMLFilename</code> logic.	280		
<code>\LWR@newhtmlfile</code> : sideTOC after title, improving responsive design.	191		
<code>\LWR@requesttoc</code> : Reorganize <code>\HomeHTMLFilename</code> logic.	207		
<code>\LWR@subhyperref</code> : Improved HTML output linebreaks.	271		

<code>\rule</code> : Added	338	<code>\LWR@hspace</code> : Add: Supports HTML thin breakable space.	335
<code>\warpHTMLonly</code> : Added.	95	<code>\LWR@htmldivclass</code> : Added optional style.	174
<code>\warpprintonly</code> : Replaces <code>\rowprintedonly</code> .	95	<code>\LWR@htmlclass</code> : Added optional style.	174
General: 2016/06/08	1	<code>\LWR@htmlsectionfilename</code> : HTMLFilename: removed additional trailing '-', and may be empty.	167
Added.	475	Sections called "Index" or "index" have an underscore prepended to their filenames if no prefix.	167
Avoids MathJax.	291	<code>\LWR@includegraphicsb</code> : Fix: <code>\linewidth</code> in a floatrow.	315
<code>cleveref</code> : Loaded		Fix: Expands filename.	315
<code>\AtEndPreamble</code> .	321	<code>\LWR@longtabledatacaptiontag</code> : Fix: Pars in captions.	256
CSS for table note item.	451	<code>\LWR@section</code> : Combined higher-level sections together into files.	195
Docs: <code>multirow</code> browser bug workaround.	74, 236	<code>\LWR@setOSWindows</code> : Auto-detects operating system.	94
Docs: Math options.	62	<code>\LWR@subhtmlclass</code> : Factored code.	173
Docs: Table: Cross-referencing data structures, updated.	264	<code>\SetHTMLFileNumber</code> : Add: Control file numbers.	166
File: <code>lwarp_mathjax.txt</code> added.	1	<code>\cpagerefFor</code> : User-redefinable word for page references.	322
File: <code>lwarp.css</code> : <code>tnoteitemheader</code> added	1	<code>\dotfill</code> : Inserts an ellipsis.	332
Introduction: MathJax support mentioned.	26	<code>\hfill</code> : Inserts a <code>\qqquad</code> .	332
MathJax support added.	293, 297, 299	<code>\hrulefill</code> : Inserts a short rule.	332
Options: <code>mathsvg</code> and <code>mathjax</code>	96	<code>\hyperindexref</code> : Print mode provided in case <code>hyperref</code> not used.	288
Supports colored <code>\rule</code> .	472	<code>\pageref</code> : Added.	270
titles: null <code>\pagestyle</code> and <code>\thispagestyle</code> for HTML.	105	<code>\tracinglwarp</code> : Added.	109
<code>tikzpicture</code> : Adapts to <code>tikz</code> version.	452	General: 2017/02/09	1
<code>equation</code> : MathJax support.	296	<code>afterpage</code> : Added.	345
v0.20		<code>alltt</code> : Added.	347
<code>\BlockClassSingle</code> : Renamed from "LWR@htmldivclassline".	176	<code>bookmark</code> : Added.	350
<code>\HTMLDescription</code> : Added <code>\NewHTMLdescription</code> . (Renamed in v0.30.)	185	caption and subcaption supported.	1
<code>\HTMLFilename</code> : No longer escape underscores.	166	<code>cleveref</code> and referencing patches: Applied <code>\AfterEndPreamble</code> .	321
<code>\HomeHTMLFilename</code> : No longer escape underscores.	166	<code>draftwatermark</code> : Added.	356
<code>\InlineClass</code> : Renamed from "inlineclass".	176	<code>eso-pic</code> : Added.	361
<code>\LWR@LwarpStart</code> : Fix: math cross references.	205	<code>everypage</code> : Added.	362
<code>\LWR@closeparagraph</code> : <code>\unskip</code> extra spaces.	179	<code>extramarks</code> : Added.	362
No break tags in the start/end of a tabular.	179	<code>fancyhdr</code> : Added.	363
<code>\LWR@endoffline</code> : Fix: <code>\\</code>	333		
<code>\LWR@filestart</code> : Adds meta description.	202		

hyperref: Additional user macros.	383	Test Suite: HTML meta descriptions.	1
keyfloat: Added.	387	Verbatim: Added.	227
letterspace: User-interface emulated.	389	verbatim: Added.	225
listings: Added.	392	BlockClass: Added optional style.	175
lrcaption: Added.	398	Renamed from "blockclass". ..	175
lwarp-newproject: Added.	112	LWR@nestspan: Fix: Minipages inside a span.	171
microtype: User-interface emulated.	408	v0.21	
needspace: Added.	413	\LWR@LwarpStart: Changed lateximages to a .txt file. .	204
nowidow: Added.	415	\LWR@filestart: Skip title if not given.	202
placeins: Added.	428	\LWR@newhtmlfile: Skip title if not given.	191
ragged2e: Added.	429	\marginpar: Fixed source listing.	188
setspace: Improved support. ..	431	General: 2017/02/23	1
textpos: Added.	446	fontenc: Added.	373
titles: Added.	453	fontspec: Added.	373
titlesec: Added.	455	inputenc: Added.	386
titletoc: Added.	456	newclude: Added.	413
titling: Improved compatibility.	457	newunicodechar: Added.	414
tocloft: Added.	458	lwarpmk: Fix: lwarpmk again for Windows.	150
wallpaper: Added.	469	lwarpmk: Fix: lwarpmk limages for Windows.	150
wrapfig: Added.	470	lwarpmk: Fix: lwarpmk uses lateximages text file instead of shell script.	150
Added @, <, > columns.	236	Add: Errors for misplaced packages.	98
Added single-expansion data arrays.	165	Docs: Added internet class. .	32
Code factored into independent lwarp_html files.	342	Docs: Added TeX2page, GladTeX.	32
Docs: Examples for generating HTML file names.	54	Docs: Installing on Windows. .	38
Docs: Improved index.	1	File lwarp_tutorial.txt added.	43
Enhanced titling support.	213	v0.22	
File: lwarp.css: Minor fixes for validation.	1	\LWR@parseDcolumn: Added tabular D column.	245
File: lwarpmk used to compile print, HTML, indexes, and lateximages.	1	\LWR@parsebangcolumn: Added tabular ! column.	242
Fix: \linewidth in a floatrow.	370	\LWR@parsetablecols: Unknown table column types become l. Added tabular D, !, X columns.	246
Improved float caption type handling.	364	\LWR@printmccoldata: Added tabular D, !, and X columns.	253
Moved sidebar and example code to test suite.	1	General: 2017/03/02	1
Page geometry set to 6in wide with large margins.	103	abstract: Added.	343
Parallel versions of aux files for print/HTML.	1	changepage: Added.	353
Removed reliance on make, grep, gawk.	1	dcolumn: Added.	355
Tabular: \unskip extra spaces.	236		

enumerate : Added.	359	emptypage : Added.	357
ftnright : Added.	380	framed : Added.	377
geometry : Nullified commands.	380	lips : Added.	391
indentfirst : Added.	386	mdframed : Help avoid	
layout : Added.	389	hyphenation.	402
lscape : Added.	398	ntheorem : Added.	416
mcaption : Added.	400	showidx : Added.	432
nameref : Added.	412	theorem : Added.	447
nextpage : Added.	414	Basic L ^A T _E X theorems: improved	
parskip : Added.	428	CSS.	231
showkeys : Added.	432	Docs: Adds credits for patched	
sidecap : Added.	433	code.	1
tabularx : Added.	444	Docs: Testing lwarp	85
varioref : Supported.	70	Fix: Allows XeL ^A T _E X and	
verse : Added.	467	LuaL ^A T _E X to preload graphics	
v0.23		and graphicx.	99
\LWR@parsetablecols : Fix for vert		v0.26	
bar column type.	246	General: 2017/03/31	1
\LWR@printmccoldata : Fix for vert		cutwin : Added.	354
bar column type.	253	endnotes : Added.	358
General: 2017/03/02	1	floatflt : Added.	366
v0.24		footmisc : Added.	373
\LWR@hspace : Add: \hspace \fill		footnotehyper : Added.	376
converts to 2em	335	footnote : Added.	375
\LWR@htmlfileref : Fix: Index		marginfix : Added.	399
links while \tracinglwarp	266	marginnote : Added.	400
\hypertocfloat : List of floats		mparhack : Added.	409
responds to lofdepth ,		pagenote : Supported as-is.	427
lotdepth	286	sidenotes : Added.	434
General: 2017/03/15	1	lwarp.css : Improved responsive	
floatrow : Support for subfig	367	marginpar and marginblock	114
subfig : Added.	438	Docs: Improved MiK _T E _X install	
tikz : For tikz v3.0.0 or later,		instructions.	35, 38
auto-loads tikz babel library if		Dollar span avoided in a	
necessary.	452	lateximage	291
Docs: Filename underscore.	52, 66	Footnotes now are L ^A T _E X boxes	
No longer preloads subcaption ;		instead of pagenotes	185
conflicted with subfig	104	lateximage : Labels track page	
picture : Fix for inline images.	323	numbers of lateximages	305
tikzpicture : Fix for inline		Print mode now uses a minipage	
images.	452	of \linewidth	305
v0.25		picture : Fix for \makebox in	
\LWR@loadnever : Added the ability		picture	323
to prevent conflicting packages.	99	v0.27	
\addcontentsline : Handles		General: 2017/04/04	1
theorems.	279	lettrine : Added.	390
General: 2016/03/22	1	microtype : Fix with XeL ^A T _E X,	
amsthm : Added.	348	LuaL ^A T _E X.	408
ccaption : Prevented.	352	soul : Added.	436
ellipsis : Added.	356	ulem : Added.	465

Docs: Installing utilities for MacOS.	40	Docs: Modifying lwarpmk and index processing.	85
Docs: Limitations of saveboxes.	75	File <code>lwarp_mathjax.txt</code> : Updated CDN repository.	148
Fix for table footnote par tags.	186	Forced oneside to maintain large right margin.	103
Page geometry modified to reduce line overflow.	103		
v0.28		v0.29	
\@wrindex: Improved indexing.	288	\LWR@includegraphicsb: Fix: Error when no optional arguments.	315
\HTMLAuthor: Added \HTMLauthor. (Renamed in v0.30.)	185	General: 2017/04/15	1
\LWR@LwarpEnd: If FormatEPUB or FormatWordProcessor, no bottom nav.	207	lwarpmk: Add: language option for config files.	150
\LWR@LwarpStart: FormatWordProcessor forces single-file output.	204	Add: lwarpmklang option for lwarp.	96
\LWR@filestart: Adds HTML meta author.	202	Docs: Using a glossary	59
\LWR@forcenewpage: Forces new PDF page before major environments.	170	File <code>*.lwarpmkconf</code> : Add: language option for config files.	113
\LWR@htmlcomment: Breaks ligatures in HTML comments.	173	File <code>lwarpmk.conf</code> : Add: language option for config files.	112
\LWR@includegraphicsb: Adapts to graphics syntax.	315	v0.30	
\LWR@newhtmlfile: If FormatEPUB or FormatWordProcessor: skips headers, footers, nav.	191	\CSSFilename: Renamed from \NewCSS.	184
\LWR@parsetablecols: Added L, C, R, J column types.	246	\HTMLAuthor: Renamed from \HTMLauthor.	185
\LWR@startref: Removed space.	268	\HTMLDescription: Renamed from \NewHTMLdescription.	185
\chapter: If EPUB, prints footnotes before each section.	200	\HTMLFirstPageTop: Renamed from \SetFirstPageTop.	183
\hyperindexref: Improved indexing.	288	\HTMLLanguage: Renamed from \MetaLanguage.	201
\textup: Fixed span class.	330	\HTMLPageBottom: Renamed from \SetPageBottom.	184
General: 2017/04/14	1	\HTMLPageTop: Renamed from \SetPageTop.	183
glossaries: Added.	381	General: 2017/04/29	1
graphics: Added.	382	lwarp-newproject removed, and combined with lwarp.	112
tabularx: Fix for optional pos.	444	lwarpmk: Add: xdyfile configuration option.	150
tabulary: Added.	445	lwarpmk: Fix: xindy and texindy adjusted for pdflatex, xelatex and lualatex.	150
lwarpmk: Add: printglossary and htmlglossary commands.	150	lwarpmk: Fix: xindy now used for print index generation with latexmk.	150
Added boolean FormatEPUB.	110		
Added boolean FormatWordProcessor.	110		
Added boolean HTMLDebugComments.	110		
Added boolean HTMLMarkFloats	110		

lwarpmk: language now used for both index and glossary generation.	150	Options HomeHTMLFilename and HTMLFilename replace macros \HomeHTMLFilename and \HTMLFilename.	97
File: lwarp_html.xdy renamed to lwarp.xdy.	148	v0.31	
Fix: *.css files only written in print mode.	114	General: 2017/05/15	1
Fix: lwarp.xdy only written in print mode.	148	keyfloat: Improved compatibility.	387
Fix: lwarp_mathjax.txt only written in print mode.	148	v0.32	
Option OSWindows replaces macro \warpOSWindows.	97	\RequirePackage: Fix: Ignores blanks in package list.	107
Option latexmk replaces macro \UseLatexmk.	97	General: 2016/06/09	1
Option lwarpmklang changed to IndexLanguage.	96	glossaries: Prevent error with \glo@name not defined.	289
Option xdyFilename added. ..	97	lwarpmk: Fix: io.lines() changed to file:lines() due to luatex changes.	150

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>291</i>	@title	<i>210</i>
\&	<i>165</i> , <u>4590</u>	\@wrglossary	<u>5575</u>
\(.....	<u>5630</u>	\@wrindex	<u>5568</u>
\)	<u>5630</u>	\\	<i>333</i>
\@@setcpageref	<u>6349</u>	\$	<i>291</i>
\@@setcref	<u>6345</u>	\$\$	<i>291</i>
\@@setcrefrange	<u>6346</u>	A	
@author	<i>210</i>	abstract (environment)	<u>4209</u>
\@begintheorem	<u>4447</u>	abstract (package)	<i>343</i>
\@capttype	<u>5324</u>	\abstractname	<i>54</i> , <u>4208</u>
\@currentlabelname	<u>5107</u>	accents	<i>194</i>
@date	<i>210</i>	\addcontentsline	<u>5378</u>
\@dlbfloat	<u>5302</u>	Adobe (program)	<i>34</i>
\@endtheorem	<u>4457</u>	\affiliation	<u>3875</u>
\@float	<u>5302</u>	afterpage (package)	<i>345</i>
\@fnsymbol	<u>4169</u>	algorithmic	
\@makecaption	<u>5326</u>	with newfloat, trivfloat	<i>463</i>
\@maketitle	<u>4180</u>	algorithmicx (package)	<i>346</i>
\@opargbegintheorem	<u>4452</u>	align (environment)	<u>5846</u>
@published	<i>210</i>	align* (environment)	<u>5868</u>
@subtitle	<i>210</i>	alltt (package)	<i>347</i>

<code>\AmS</code>	6698	<code>LWR@validtablecol</code>	246
<code>amsmath</code> (package)	105	<code>LWR@verbtags</code>	226
<code>amsthm</code> (package)	348	<code>mathjax</code>	95
<code>\and</code>	210	<code>usingOSWindows</code>	94
<code>array</code> (package)	263	<code>warpingHTML</code>	95
<code>AsciiDoc</code> (program)	33	<code>warpingprint</code>	95
<code>AsciiDoctor</code> (program)	33	bugs	86
<code>Asciidoctor-LaTeX</code> (program)	33	<code>BVerbatim</code> (environment)	4432
<code>\attrib</code>	77, 224, 467		
<code>\attribution</code>	4219		
author			
HTML meta tag	68, 185		
<code>\author</code>	67, 210, 3882		
<code>autosec</code>	194		
B		C	
<code>babel</code> (package)	79	<code>calc</code> (package)	104
<code>\backmatter</code>	3536	<code>Calibre</code>	81
<code>\BaseJobname</code>	2890	<code>caption</code> (package)	104
<code>BaseJobname</code> (option)	96	<code>\caption@begin</code>	5357
<code>baseline</code>		<code>\caption@end</code>	5357
tabular	246	<code>\captionlistentry</code>	5361
<code>\bfseries</code>	6509	<code>\captionof</code>	5391
<code>\BibTeX</code>	6693	<code>\CaptionSeparator</code>	5325
<code>BlockClass</code> (environment)	3107	<code>ccaption</code> (package)	352
<code>\BlockClassSingle</code>	3119	<code>center</code> (environment)	6068
<code>bookmark</code> (package)	350	<code>\centering</code>	205
<code>booktabs</code> (package)	352	<code>changepage</code> (package)	353
booleans:		<code>\chapter</code>	3661
<code>CombineHigherDepths</code>	53, 188	<code>\citetitle</code>	4226
<code>FileSectionNames</code>	54, 167	class:	
<code>FormatEPUB</code>	81, 110	internet	32
<code>FormatWordProcessor</code>	83, 110	<code>cleveref</code> (package)	321
<code>HTMLDebugComments</code>	110	<code>cmap</code> (package)	62
<code>HTMLMarkFloats</code>	83, 110	<code>CombineHigherDepths</code> (boolean)	53, 188
<code>LWR@amsmultline</code>	297	<code>comment</code> (package)	93
<code>LWR@doingapar</code>	178	<code>Computer Modern</code>	61
<code>LWR@doinghline</code>	237	<code>\ConTeXt</code>	6690
<code>LWR@doingstartpars</code>	177	counters:	
<code>LWR@doingtbrule</code>	238	<code>FileDepth</code>	53, 188
<code>LWR@freezethisfloat</code>	276	<code>lofdepth</code>	285
<code>LWR@infloatrow</code>	314	<code>lotdepth</code>	285
<code>LWR@intabularmetadata</code>	238	<code>LWR@externalfilecnt</code>	291
<code>LWR@minipagefullwidth</code>	325	<code>LWR@htmlfilenumber</code>	167
<code>LWR@minipagethispar</code>	325	<code>LWR@latestautopage</code>	277
<code>LWR@skippingmrowcell</code>	238	<code>LWR@lateximagedepth</code>	304
<code>LWR@starredlongtable</code>	256	<code>LWR@lateximagenumber</code>	304
<code>LWR@startedrow</code>	237	<code>LWR@LIpage</code>	304
<code>LWR@tableparcell</code>	238	<code>LWR@midrulecounter</code>	251
<code>LWR@tracinglwarp</code>	109	<code>LWR@minipagedepth</code>	324
		<code>LWR@nextautofloat</code>	277
		<code>LWR@nextautopage</code>	277
		<code>LWR@nextequation</code>	293
		<code>LWR@prevFileDepth</code>	195
		<code>LWR@spandepth</code>	177
		<code>LWR@tablecolspan</code>	241

LWR@tablecolswidth	241	BlockClass	3107
LWR@tabletotalcols	241	BVerbatim	4432
LWR@tabletotalcolsnext	241	center	6068
LWR@thisfloat	276	description	4519
SideTOCDepth	53, 283	displaymath	5639
tocdepth	52	enumerate	4488
\cpagerefFor	6348	equation	5706
cross-references		flalign	5890
missing or incorrect	86	flalign*	5912
CSS		flushleft	6080
file selection	69	flushright	6074
lwrap.css	68	gather	5802
per HTML page	69	gather*	5824
project-specific changes	68	itemize	4470
\CSSFilename	51, 54, 69, 3296	lateximage	5982
cutwin (package)	354	longtable	2
		LVerbatim	4444
D		LWR@nestspan	3004
danger icon	91	LWR@tabular	5064
\date	67	math	5638
dcolumn (package)	355	minipage	6416
debugging	86	multline	5755
HTML debug comments	110	multline*	5778
\DeclareGraphicsExtensions	6098	picture	322, 6366
\DeclareGraphicsRule	6098	quote	4234
Deja Vu	61	theindex	5551
description		tikzpicture	8
HTML meta tag	68, 185	titlepage	67, 3916
description (environment)	4519	titlingpage	67, 3922, 3935
displaymath (environment)	5639	Verbatim	4299
\dotfill	6525	verbatim	4255
draftwatermark (package)	356	VerbatimClass	4291
		verse	2
E		warpall	45, 64
ellipsis (package)	356	warpHTML	53, 56, 64
\emph	6498	warpprint	46, 55, 64
emptypage (package)	357	epigraph (package)	360
\end@dblfloat	5316	Epub	
\end@float	5316	conversion software	81
endnotes (package)	358	HTML conversion settings	81, 110
\enlargethispage	6628	equation (environment)	5706
\enskip	334, 6592	error messages	86
enumerate (environment)	4488	eso-pic (package)	361
enumerate (package)	80, 359	etoolbox (package)	93
enumitem (package)	79, 104	everyhook (package)	103
environ (package)	105	everypage (package)	362
environments:		expl3 (package)	103
abstract	4209	extramarks (package)	362
align	5846		
align*	5868		

F	
<code>fancyhdr</code> (package)	363
<code>fancyvrb</code> (package)	103
FAQ	86
<code>fbox</code>	
around a minipage	66
<code>\fbox</code>	6488
file	
inaccessible	53
underscore	52, 66
<code>FileDepth</code> (counter)	53, 188
files:	
<code>glyptounicode</code>	61
<code>lwrap.css</code>	68, 114
<code>lwrap.xdy</code>	69, 148
<code>lwrap_formal.css</code>	143
<code>lwrap_mathjax.txt</code>	148
<code>lwrap_sagebrush.css</code>	138
<code>lwrap_tutorial.txt</code>	43
<code>lwarpmk.conf</code>	112
<code>lwarpmk.lua</code>	85
<code>project.css</code>	68
<code>project.lwarpmkconf</code>	113
<code>project_html.tex</code>	112
<code>sample_project.css</code>	68, 147
<code>tutorial.tex</code>	43
<code>FileSectionNames</code> (boolean)	54, 167
<code>\flagverse</code>	468
<code>flalign</code> (environment)	5890
<code>flalign*</code> (environment)	5912
Flare (program)	34
<code>float</code> (package)	364
<code>floatflt</code> (package)	366
<code>\floatname</code>	11
<code>\floatplacement</code>	14
<code>floatrow</code> (package)	367
<code>\floatstyle</code>	17
<code>flushleft</code> (environment)	6080
<code>flushright</code> (environment)	6074
font	
bitmapped	61
Computer Modern	61
Deja Vu	61
ligatures	61, 62
packages	61
selection	61
size — <code>lateximage</code>	70, 289, 304
size — math, SVG	70, 289, 304
size — <code>xfrac</code>	475
<code>fontenc</code> (package)	61, 373
<code>fontspec</code> (package)	61, 373
<code>footmisc</code> (package)	374
<code>footnote</code> (package)	375
<code>footnotehyper</code> (package)	376
<code>footnotes</code>	185
for	
HTML & PRINT	91
HTML output	91
PRINT output	91
<code>FormatEpub</code> (boolean)	81, 110
<code>FormatWordProcessor</code> (boolean)	83, 110
<code>\framebox</code>	6485
<code>framed</code> (package)	377
FrameMaker (program)	34
Frequently Asked Questions	86
<code>\frontmatter</code>	3533
<code>ftnright</code> (package)	380
<code>\fup</code>	6522
G	
<code>gather</code> (environment)	5802
<code>gather*</code> (environment)	5824
GELLMU (program)	32
generator	
HTML meta tag	203
<code>geometry</code> (package)	102, 380
<code>getttitlestring</code> (package)	103
GladTeX (program)	33
<code>glossaries</code> (package)	79, 381
glossary	
language	59
processing	59
<code>glyptounicode</code> (file)	61
<code>graphics</code> (package)	310, 382
<code>\graphicspath</code>	6096
<code>graphicx</code> (package)	310, 382
H	
Hevea (program)	32
<code>\hfill</code>	6523
<code>\HomeHTMLFilename</code>	2892
<code>HomeHTMLFilename</code> (option)	52, 97
horizontal space	
between minipages	333
<code>\href</code>	5249
<code>\hrulefill</code>	6524
<code>\hspace</code>	334, 6621
HTML	
commands for a successful conver-	
sion	65

conversion settings	52	item, list, empty	66
debug comments	110	itemize (environment)	4470
EPUB	81, 110	\itshape	6514
word processor	83, 110		
filename generation	54	J	
headings	89	JavaScript	
meta tag		MathJax	70, 289
author	68, 185		
description	68, 185	K	
generator	203	keyfloat (package)	387
viewport	203	kvoptions (package)	95
tabular column conversion	247		
\HTMLAuthor	55, 68, 3310	L	
HTMLDebugComments (boolean)	110	label	
\HTMLDescription	55, 68, 3315	in HTML	205
\HTMLEntity	2880	math environment	298
\HTMLFilename	2891	language	
HTMLFilename (option)	52, 97	glossary	59
\HTMLFirstPageTop	55, 3273	index	59
\HTMLLanguage	55, 3718	language HTML metadata	201
HTMLleftmargin (length)	78, 224, 225, 467	\LaTeX	6676
HTMLMarkFloats (boolean)	83, 110	LaTeX2HTML (program)	32
\HTMLPageBottom	55, 210, 3289	\LaTeXe	6676
\HTMLPageTop	55, 3281	lateximage (environment)	5982
\HTMLUnicode	2881	\LateximageFontSizeName	5946
HTMLvleftskip (length)	78, 224, 467	lateximages	
\hyperindexref	5582	font size	70, 289, 304
hyperref (package)	271, 383	latexmk (option)	52, 97
\hypertoc	5493	LaTeXML (program)	32
\hypertocfloat	5506	layout (package)	389
		lengths:	
I		HTMLleftmargin	78, 224, 225, 467
icon		HTMLvleftskip	78, 224, 467
warning	91	LWR@minipageheight	324
ifplatform (package)	93	LWR@minipagewidth	324
images		LWR@tempheight	335
graphicx package	310	LWR@tempraise	335
in strange places	87	LWR@tempwidth	335
\includegraphics	6281	\VerbatimHTMLWidth	225
indentfirst (package)	386	vleftmargin	78, 224, 467
InDesign (program)	34	vleftskip	78, 224, 467, 468
index		letterspace (package)	389
language	59	lettrine (package)	390
processing	47, 48	LibreOffice (program)	33
UTF-8	62	ligatures	61, 62, 102
IndexLanguage (option)	59, 79, 96, 381	line numbers	91
\InlineClass	3127	\linebreak	6622
inputenc (package)	61, 386	\LinkHome	2942
internet (class)	32	Linux (program)	64, 94
		lips (package)	391

list item, empty	66	LWR@colbeforesec	242
listings (package)	392	\LWR@copyfile	353
\listof	5459	\LWR@createautosec	3542
lmodern (package)	61	\LWR@currentcss	3295
lofdepth (counter)	285	\LWR@descitem	4507
longtable (environment)	2	\LWR@docmidrule	4840
longtable (package)	396	LWR@doingapar (boolean)	178
lotdepth (counter)	285	LWR@doinghline (boolean)	237
lscope (package)	398	LWR@doingstartpars (boolean)	177
ltcaption (package)	398	LWR@doingtbrule (boolean)	238
LuaLaTeX		\LWR@domulticolumn	4913
detection	101	\LWR@doubledollar	5608
file & section names	194	\LWR@endhideamsmath	5748
\LuaLaTeX	6684	\LWR@endminipage	6380
LuaTeX (program) [requirement]	36	\LWR@endoffline	6569
\LuaTeX	6684	\LWR@endsubminipage	6401
LVerbatim (environment)	4444	\LWR@ensuredoingapar	3168
lwarp		LWR@externalfilecnt (counter)	291
loading	62	\LWR@filenamenoblanks	3387
options	62	\LWR@filestart	3725
lwarp (package)	62	\LWR@findword	288
lwarp.css (file)	68, 114	\LWR@floatbegin	5286
lwarp.xdy		\LWR@floatend	5304
customizing	69	\LWR@forcenewpage	2976
lwarp.xdy (file)	69, 148	LWR@freezethisfloat (boolean)	276
lwarp_formal.css (file)	143	\LWR@futurenonospacelet	4547
lwarp_mathjax.txt (file)	148	\LWR@getexparray	2876
lwarp_sagebrush.css (file)	138	\LWR@getmynexttoken	4554
lwarp_tutorial.txt (file)	43	\LWR@hidelatexequation	5681
lwarpmk		\LWR@hspace	6600
customizing	85	\LWR@htmlblockcomment	3049
lwarpmk (option)	97	\LWR@htmlblocktag	3057
lwarpmk (program)	85, 150	\LWR@htmlclosecomment	3035
lwarpmk.conf (file)	112	\LWR@htmlcomment	3042
lwarpmk.lua (file)	85	\LWR@htmldivclass	3084
\LWR@addmathjax	5695	\LWR@htmldivclassend	3087
\LWR@afterendverbatim	4283	\LWR@htmlmlelement	3097
LWR@amsmultline (boolean)	297	\LWR@htmlmlelementclass	3071
\LWR@atbeginverbatim	4275	\LWR@htmlmlelementclassend	3076
\LWR@beginhideamsmath	5740	\LWR@htmlmlelementclassline	3090
\LWR@botnavigation	2950	\LWR@htmlmlelementend	3100
\LWR@caption@begin	5338	LWR@htmlfilenumber (counter)	167
\LWR@caption@end	5348	\LWR@htmlfileref	5133
\LWR@clearmidrules	4821	\LWR@htmlmathlabel	5726
\LWR@closeparagraph	3194	\LWR@htmlopencomment	3035
\LWR@closeprevious	2970	\LWR@htmlrefsectionfilename	2935
\LWR@closetabledatacell	4568	\LWR@HTMLsanitize	5949
LWR@colafterspec	242	\LWR@HTMLsanitizeexpand	5967
LWR@colatspec	242	\LWR@htmlsectionfilename	2902
LWR@colbangspec	242	\LWR@htmlspan	3010

<code>\LWR@htmlspanclass</code>	3018	<code>\LWR@originnames</code>	6300
<code>\LWR@htmltag</code>	3026	<code>\LWR@parseaftercolumn</code>	4679
<code>\LWR@htmltagc</code>	2997	<code>\LWR@parseatcolumn</code>	4627
<code>\LWR@includegraphicsb</code>	6189	<code>\LWR@parsebangcolumn</code>	4649
<code>\LWR@indexitem</code>	5556, 5560, 5564	<code>\LWR@parsebeforecolumn</code>	4670
<code>\LWR@indexsection</code>	5540	<code>\LWR@parseDcolumn</code>	4706
<code>LWR@infloatrow</code> (boolean)	314	<code>\LWR@parsenormalcolumn</code>	4692
<code>LWR@intabularmetadata</code> (boolean) .	238	<code>\LWR@parsepcolumn</code>	4702
<code>\LWR@itemizeitem</code>	4463	<code>\LWR@parseskipcolumn</code>	4689
<code>LWR@latestautopage</code> (counter) ...	277	<code>\LWR@parsetablecols</code>	4711
<code>LWR@lateximagedepth</code> (counter) ...	304	<code>LWR@prevFileDepth</code> (counter)	195
<code>\LWR@lateximagedepthref</code>	5137	<code>\LWR@printmccoldata</code>	4874
<code>LWR@lateximagenumber</code> (counter) ...	304	<code>\LWR@printmccoltype</code>	4844
<code>\LWR@lateximagenumberref</code>	5140	<code>\LWR@printheadtitle</code>	4011
<code>LWR@LPage</code> (counter)	304	<code>\LWR@ProvidesPackageDrop</code>	343
<code>\LWR@loadafter</code>	97	<code>\LWR@ProvidesPackagePass</code>	331
<code>\LWR@loadbefore</code>	108	<code>\LWR@pushoneclose</code>	3545
<code>\LWR@loadnever</code>	113	<code>\LWR@requesttoc</code>	3845
<code>\LWR@longtabledatacaptiontag</code> ..	4934	<code>\LWR@requirepackagenames</code>	287
<code>\LWR@lookforpackagename</code>	291	<code>\LWR@restoremathlatexformatting</code>	5663
<code>\LWR@LwarpEnd</code>	3852, 6720	<code>\LWR@rotboxorigin</code>	6292
<code>\LWR@LwarpStart</code>	3782, 6720	<code>\LWR@rotstyle</code>	6179
<code>\LWR@maketitlesetup</code>	4160	<code>\LWR@rule</code>	6629
<code>\LWR@maybeinthisfloat</code>	5321	<code>\LWR@scalestyle</code>	6182
<code>\LWR@maybenewtablerow</code>	4770	<code>\LWR@section</code>	3552
<code>LWR@midrulecounter</code> (counter)	251	<code>\LWR@sectionnumber</code>	3539
<code>LWR@midrules</code>	251	<code>\LWR@setexparray</code>	2873
<code>LWR@minipagedepth</code> (counter)	324	<code>\LWR@setlatestname</code>	5109
<code>LWR@minipagefullwidth</code> (boolean) .	325	<code>\LWR@setOSWindows</code>	30
<code>LWR@minipageheight</code> (length)	324	<code>\LWR@sidetoc</code>	5472
<code>\LWR@minipagestartpars</code>	6574	<code>\LWR@singledollar</code>	5620
<code>\LWR@minipagestoppars</code>	6580	<code>LWR@skippingmrowcell</code> (boolean) .	238
<code>LWR@minipagethispar</code> (boolean) ..	325	<code>LWR@spandepth</code> (counter)	177
<code>LWR@minipagewidth</code> (length)	324	<code>\LWR@splabel</code>	5143
<code>\LWR@multicolother</code>	4865	<code>LWR@starredlongtable</code> (boolean) .	256
<code>\LWR@multicolpartext</code>	4860	<code>LWR@startedrow</code> (boolean)	237
<code>\LWR@multicolskip</code>	4871	<code>\LWR@startnewdepth</code>	3546
<code>\LWR@mynexttoken</code>	4546	<code>\LWR@startpars</code>	3232
<code>\LWR@myshorttoc</code>	5403	<code>\LWR@startref</code>	5180
<code>\LWR@nameref</code>	5130	<code>\LWR@stoppars</code>	3244
<code>LWR@nestspan</code> (environment)	3004	<code>\LWR@stripperperiod</code>	5108
<code>\LWR@newhtmlfile</code>	3465	<code>\LWR@strresult</code>	4620
<code>\LWR@newlabel</code>	5167	<code>\LWR@subcmidrule</code>	4832
<code>\LWR@newref</code>	5216	<code>\LWR@subhtml-elementclass</code>	3062
<code>LWR@nextautofloat</code> (counter)	277	<code>\LWR@subhyperref</code>	5240
<code>LWR@nextautopage</code> (counter)	277	<code>\LWR@subhyperrefclass</code>	5244
<code>LWR@nextequation</code> (counter)	293	<code>\LWR@subinlineimage</code>	5279
<code>\LWR@openparagraph</code>	3173	<code>\LWR@subminipage</code>	6395
<code>\LWR@origcolspec</code>	4622	<code>\LWR@subnewref</code>	5210
<code>\LWR@originname</code>	6293	<code>\LWR@subsublabel</code>	5146

newfloat (package)	104	ccaption	352
\newline	6567	change page	353
\newlist	4537	cleveref	321
\newtheorem	231	cmap	62
newtxmath (package)	78	comment	93
newunicodechar (package)	61, 414	cutwin	354
nextpage (package)	414	dcolumn	355
\nolinebreak	6623	draftwatermark	356
\nolinkurl	5261	ellipsis	356
\nopagebreak	6627	emptypage	357
nowidow (package)	415	endnotes	358
ntheorem (package)	416	enumerate	80, 359
\numberline	5490	enumitem	79, 104
numbers		environ	105
left margin	91	epigraph	360
O			
OpenOffice (program)	33	eso-pic	361
options:		etoolbox	93
BaseJobname	96	everyhook	103
HomeHTMLFilename	52, 97	everypage	362
HTMLFilename	52, 97	expl3	103
IndexLanguage	59, 79, 96, 381	extramarks	362
latexmk	52, 97	fancyhdr	363
lwarpmk	97	fancyvrb	103
mathjax	52, 62, 96	float	364
mathsvg	52, 62, 96	floatflt	366
OSWindows	64, 94, 97	floatrow	367
warpHTML	62, 96	fontenc	61, 373
warpprint	62, 95	fontspec	61, 373
xdyFilename	69, 97	footmisc	374
\OSPathSymbol	29	footnote	375
OSWindows (option)	64, 94, 97	footnotehyper	376
P			
packages		framed	377
required	101	ftnright	380
packages:		geometry	102, 380
abstract	343	getttitlestring	103
afterpage	345	glossaries	79, 381
algorithmicx	346	graphics	310, 382
alltt	347	graphicx	310, 382
amsmath	105	hyperref	271, 383
amsthm	348	ifplatform	93
array	263	indentfirst	386
babel	79	inputenc	61, 386
bookmark	350	keyfloat	387
booktabs	352	kvoptions	95
calc	104	layout	389
caption	104	letterspace	389
		lettrine	390
		lips	391
		listings	392
		lmodern	61

longtable	396	ulem	465
lscap	398	verse	467, 468
lrcaption	398	wallpaper	469
lwrap	62	wrapfig	470
makeidx	104	xcolor	472
marginfix	399	xfrac	106, 475
marginnote	400	xifthen	104
mcaption	400	xparse	103
mdframed	401	xstring	104
memoir	468	zref	105
microtype	62, 102, 408	page	
mparhack	409	inaccessible	53
multicol	410	\pagebreak	6624
multitrow	259, 412	pagenote (package)	427
nameref	412	\pagenumbering	280
needspace	413	\pageref	5223
newclude	78, 413	\pagerefPageFor	5222
newfloat	104	\pagestyle	276
newtxmath	78	Pandoc (program)	33
newunicodechar	61, 414	\paragraph	3700
nextpage	414	\parbox	6473
nowidow	415	\parsemulticolumnalignment	4901
ntheorem	416	parskip (package)	428
pagenote	427	\part	3649
parskip	428	pdfcrop (program) [requirement]	36
placeins	428	pdfLaTeX (program) [requirement]	36
ragged2e	429	pdfseparate (program) [requirement]	
refcount	104		36, 41
rotating	430	pdftocairo (program) [requirement]	
setspace	431		36, 41
showidx	432	pdftotext (program) [requirement]	36, 41
showkeys	432	Perl	41
sidecap	433	perl (program) [requirement]	41
sidenotes	434	\phantomsection	6670
siunitx	78, 310	picture (environment)	322, 6366
soul	436	placeins (package)	428
subfig	438	Plastex (program)	32
tabularx	444	\popclose	2846
tabulary	445	Poppler	36, 41
textcomp	62	\postpublished	209
textpos	446	\postsuffix	209
theorem	447	\prepublished	209
threeparttable	451	\presuffix	209, 219
tikz	452	\printauthor	210, 4028, 4043
titles	105, 453	\printdate	210, 4033, 4045
titlesec	455	\printindex	5543
titletoc	456	\printpublished	210, 3994, 4040
titling	104, 457	\PrintStack	2955
tocloft	458	\printsuffix	210, 4019, 4042
trivfloat	463	\printthanks	209, 213

<code>\printtitle</code>	210, 4003, 4041		
problems	86		
programs			
utility	35		
programs:			
[requirement]:			
LuaLaTeX	36		
MathJax	36		
pdfcrop	36		
pdfLaTeX	36		
pdfseparate	36, 41		
pdftocairo	36, 41		
pdftotext	36, 41		
perl	41		
XeLaTeX	36		
Adobe	34		
AsciiDoc	33		
AsciiDoctor	33		
AsciiDoctor-LaTeX	33		
Flare	34		
FrameMaker	34		
GELLMU	32		
GladTeX	33		
Hevea	32		
InDesign	34		
LaTeX2HTML	32		
LaTeXML	32		
LibreOffice	33		
Linux	64, 94		
lwarpmk	85, 150		
Mac OS	64, 94		
Madcap	34		
MathJax	71, 290		
MS-Windows	64, 94		
OpenOffice	33		
Pandoc	33		
Plastex	32		
TeX2page	32		
TeX4ht	32		
TtH	32		
Unix	64, 94		
Windows	64, 94		
Word	33		
xindy	69		
project.css (file)	68		
project.lwarpmkconf (file)	113		
project_html.tex (file)	112		
\published	67, 3892		
\pushclose	2819		
		Q	
		<code>\qqquad</code>	334, 6591
		<code>\quad</code>	334, 6586
		<code>quote</code> (environment)	4234
		R	
		<code>ragged2e</code> (package)	429
		<code>\raggedleft</code>	205
		<code>\raggedright</code>	205
		<code>\raisebox</code>	6493
		<code>\ref</code>	5216
		<code>refcount</code> (package)	104
		<code>\reflectbox</code>	6338
		<code>\renewlist</code>	4537
		[requirement]:	
		LuaLaTeX (program)	36
		MathJax (program)	36
		pdfcrop (program)	36
		pdfLaTeX (program)	36
		pdfseparate (program)	36, 41
		pdftocairo (program)	36, 41
		pdftotext (program)	36, 41
		perl (program)	41
		XeLaTeX (program)	36
		<code>\RequirePackage</code>	307
		<code>\resizebox</code>	6341
		<code>\restylefloat</code>	19
		<code>\rmfamily</code>	6510
		<code>\rotatebox</code>	6312
		<code>rotating</code> (package)	430
		<code>\rule</code>	6667
		S	
		<code>sample_project.css</code> (file)	68, 147
		<code>\sb</code>	6518
		<code>\scalebox</code>	6327
		<code>\scshape</code>	6515, 6516
		section	
		depths	89
		missing	53
		<code>\section</code>	3675
		<code>\SetHTMLFileNumber</code>	2893
		<code>setspace</code> (package)	431
		settings	
		css project-specific	68
		CSS selection	69
		HTML conversion	52
		lwrap package options	62
		selecting output	64
		title page	67

<code>\url</code>	5268	<code>warpingHTML</code> (boolean)	95
<code>\UseVerbatim</code>	4445	<code>warpingprint</code> (boolean)	95
<code>usingOSWindows</code> (boolean)	94	<code>warpprint</code> (environment) ...	46, 55, 64
UTF-8		<code>warpprint</code> (option)	62, 95
enhanced coverage	61	<code>\warpprintonly</code>	43, 56, 65
file & section names	194	Windows (program)	64, 94
index	62	Word (program)	33
selection	61	word processor	
utility		HTML conversion settings ..	83, 110
programs	35	<code>wrapfig</code> (package)	470
V		X	
<code>Verbatim</code> (environment)	4299	<code>xcolor</code> (package)	472
<code>verbatim</code> (environment)	4255	<code>xdyFilename</code> (option)	69, 97
<code>VerbatimClass</code> (environment)	4291	XeLaTeX	
<code>\VerbatimHTMLWidth</code> (length)	225	detection	101
<code>verse</code> (environment)	2	file & section names	194
<code>verse</code> (package)	467, 468	<code>\XeLaTeX</code>	6686
viewport		<code>XeLaTeX</code> (program) [requirement] ...	36
HTML meta tag	203	<code>\XeTeX</code>	6686
<code>vleftmargini</code> (length) ...	78, 224, 467	<code>xfrac</code> (package)	106, 475
<code>vleftskip</code> (length) ..	78, 224, 467, 468	<code>xifthen</code> (package)	104
W		xindy	
<code>wallpaper</code> (package)	469	customizing	69
warning icon	91	<code>xindy</code> (program)	69
<code>warpall</code> (environment)	45, 64	<code>xparse</code> (package)	103
<code>warpHTML</code> (environment)	53, 56, 64	<code>xstring</code> (package)	104
<code>warpHTML</code> (option)	62, 96	Z	
<code>\warpHTMLonly</code>	44, 56, 65	<code>zref</code> (package)	105