

Lwarp

LATEX HTML5

The **l warp** package

LATEX to HTML

v0.65 — 2018/12/22

© 2016–2018 Brian Dunn
bd@BDTechConcepts.com

Abstract

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

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

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

A modular package-loading system uses the **l warp** version of a package for HTML when available. More than 300 LATEX packages are supported with these high-level source-compatibility replacements, and many others work as-is.

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

To update existing projects, see section 2: Updates.

For a list of supported features, see table 2: Supported packages and features.

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

License:

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

1 Supporting T_EX development

T_EX and related projects:

- are mostly open-sourced and a volunteer effort;
- benefit students, academics, scientists, engineers, and businesses;
- help drive education, public and private research, and commercial activity;
- are used in the fields of mathematics, science, engineering, and humanities;
- are international in reach;
- span decades of development;
- are enduring—many older packages are still actively used and maintained;
- are largely backwards compatible;
- are portable across all the major computing platforms;
- are usable even on older computers and away from internet access;
- are continuing to maintain relevance with modern improvements;
- require no yearly subscription fees;
- and are supported by an active community of knowledgeable volunteers.

Please consider helping by joining and/or contributing to the T_EX Users Group, a United States 501(c)(3) tax-exempt charitable organization. Contributions are accepted by credit card, check, or Pay Pal, via the United Way, or by USA or European bank transfer. Membership in TUG supports the development of T_EXLive, the major T_EX distribution.

Donations may be directed towards individual projects:

TUG Bursary Fund: Assistance for attending annual TUG meetings.

CTAN: The Comprehensive T_EX Archive Network—Central storage for T_EX.

TeX Development Fund: Support for specific projects.

EduTeX: Teaching and using T_EX in schools and universities.

GUST e-foundry fonts: Enhanced for math and additional language groups.

LaTeX Project: Modernizing the L^AT_EX core.

Libre Font Fund: Fonts, tools (FontForge), and distribution (the Open Font Library).

LuaTeX: Combining the pdfT_EX engine and the Lua language.

MetaPost: Postscript graphics.

MacTeX: T_EX for Mac.

PDF Accessibility: Modern PDF standards.

Other: Additional projects may be specified.

To make a contribution: <https://www.tug.org/donate.html>

For country-specific T_EX users groups: <http://tug.org/usergroups.html>

For users of MiK^TE_X: <https://miktex.org/donations.html>

Contents

1 Supporting T<small>EX</small> development	2
List of Figures	55
List of Tables	55
2 Updates	56
3 Introduction	76
3.1 Typesetting conventions	78
3.2 Supported packages and features	79
4 Alternatives	85
4.1 internet class	85
4.2 TeX4HT	85
4.3 Translators	85
4.4 ASCII <small>D<small>O<small>C<small>T<small>R</small></small></small></small></small> and ASCII <small>D<small>O<small>C<small>T<small>R</small></small></small></small>OR</small>	86
4.4.1 ASCII <small>D<small>O<small>C<small>T<small>R</small></small></small></small>-L<small>T<small>E<small>X</small></small></small></small>	86
4.5 PANDOC	86
4.6 Word processors	86
4.7 Commercial systems	86
4.8 Comparisons	86
5 Installation	88
5.1 Installing the <i>lwarpm</i> package	90
5.2 Installing the <i>lwarpmk</i> utility	92
5.2.1 Using a local copy of <i>lwarpmk</i>	93
5.3 Installing additional utilities	94
6 Tutorial	96
6.1 Starting a new project	96
6.2 Compiling the print version with <i>lwarpmk</i>	101
6.3 Compiling the HTML version with <i>lwarpmk</i>	102
6.4 Generating the SVG images	103
6.5 Using MATHJAX for math	105
6.6 Changing the css style	106
6.7 Customizing the HTML output	106
6.8 Using <i>latexmk</i>	107
6.9 Using X <small>L<small>T<small>E<small>X</small></small></small> or L<small>u<small>a<small>tex</small></small></small><small>M<small>X</small></small></small>	108
6.10 Using DVI L <small>T<small>E<small>X</small></small></small>	108
6.11 Using a glossary	109
6.12 Cleaning auxiliary files	110
6.13 Cleaning auxiliary and output files	110
6.14 Cleaning the images from the <i>lateximages</i> directory	110
6.15 Converting PDF or EPS images to SVG	110

6.16	Creating HTML from an incomplete compile	110
6.17	Processing multiple projects in the same directory	111
6.18	Using the <i>make</i> utility	111
7	Converting an existing document	112
8	Additional details	113
8.1	Shell escape	113
8.2	Font and UTF-8 support	113
8.2.1	Indexes, glossaries, and encoding	115
8.3	l warp package loading and options	116
8.4	Customizing the HTML output	120
8.4.1	Example HTML file naming	126
8.5	Customizing the css	126
8.6	Selecting the operating system	128
8.7	Selecting actions for print or HTML output	128
8.8	Commands to be placed into the warpprint environment	129
8.9	Title page	130
8.10	HTML page meta descriptions	131
8.11	HTML homepage meta title	131
8.12	HTML page meta author	131
9	Special cases and limitations	132
9.1	Things to avoid	132
9.2	Formatting	133
9.2.1	Text formatting	133
9.2.2	Horizontal space	133
9.2.3	Text alignment	133
9.2.4	Accents	134
9.2.5	textcomp package	134
9.2.6	Superscripts and other non-math uses of math mode	134
9.2.7	Empty \item followed by a new line of text or a nested list:	134
9.2.8	Filenames and URLs in lists or footnotes	134
9.2.9	relsize package	134
9.3	Boxes and minipages	135
9.3.1	Marginpars	135
9.3.2	Save Boxes	135
9.3.3	Minipages	135
9.3.4	Side-by-side minipages	136
9.3.5	Framed minipages and other environments	136
9.3.6	fancybox package	137
9.3.7	mdframed package	139
9.4	Cross-references	139
9.4.1	Page references	139
9.4.2	cleveref and varioref packages	140
9.4.3	Hyperlinks, hyperref, and url	140

9.4.4	Footnotes and page notes	141
9.5	Front and back matter	142
9.5.1	Custom classes with multiple authors and affiliations	142
9.5.2	Starred chapters and sections	142
9.5.3	abstract package	143
9.5.4	titling and authblk	143
9.5.5	tocloft package	143
9.5.6	appendix package	143
9.5.7	pagenote package	143
9.5.8	endnotes package	144
9.5.9	BIBTEX	144
9.5.10	glossaries package	144
9.5.11	Indexing overview	145
9.5.12	Indexing with basic LATEX and makeidx	146
9.5.13	Indexing with index	147
9.5.14	Indexing with splitidx	148
9.5.15	Indexing with imakeidx	150
9.5.16	Indexes with memoir	153
9.5.17	Using a custom makeindex style file	156
9.5.18	Using a custom xindy style file	156
9.5.19	Additional indexing limitations	157
9.5.20	Index positions, toc, tocbibind	158
9.6	Math	159
9.6.1	Rendering tradeoffs	159
9.6.2	svg option	159
9.6.3	MATHJAX option	160
9.6.4	Customizing MATHJAX	160
9.6.5	MATHJAX limitations	161
9.6.6	Catcode changes	161
9.6.7	Complicated inline math objects	162
9.6.8	Complicated display math objects	162
9.6.9	ntheorem package	162
9.6.10	siunitx package	163
9.6.11	units and nicefrac packages	163
9.6.12	newtxmath package	163
9.7	Graphics	163
9.7.1	tikz package	165
9.7.2	grffile package	165
9.7.3	color package	165
9.7.4	xcolor package	165
9.7.5	epstopdf package	166
9.7.6	pstricks package	166
9.7.7	pdftricks package	166
9.7.8	psfrag package	167
9.7.9	pstool package	167
9.7.10	asymptote package	167

9.7.11	overpic package	168
9.8	Tabbing	168
9.9	Tabular	168
9.9.1	longtable package	171
9.9.2	supertabular and xtab packages	171
9.9.3	bigdelim package	172
9.10	Floats	172
9.10.1	Float contents alignment	172
9.10.2	float, trivfloat, and/or algorithmicx together	172
9.10.3	caption and subcaption packages	173
9.10.4	subfig package	173
9.10.5	floatrow package	173
9.10.6	keyfloat package	174
9.11	KOMA-SCRIPT classes	174
9.12	MEMOIR class	174
9.13	Miscellaneous packages	175
9.13.1	verse and memoir	175
9.13.2	newclude package	176
9.13.3	babel package	176
9.13.4	polyglossia package	176
9.13.5	todonotes and luatodonotes packages	177
9.13.6	fixme	177
9.13.7	chemfig package	177
9.13.8	chemformula package	177
9.13.9	mhchem package	178
9.13.10	xparse	178
10	Compiling using custom shell commands	179
10.1	Command options	179
10.2	Literal character macros	179
10.3	<i>latexmk</i>	181
10.4	perltex package	181
10.5	pythontex package	181
10.6	Other packages	182
10.7	<i>make</i> program	182
10.8	UTF-8 locale	182
11	EPUB conversion	184
12	Word-processor conversion	186
12.1	Activating word-processor conversion	186
12.2	Additional modifications	188
12.3	Recommendations	189
12.4	Limitations	190
13	Modifying l warp	192

13.1	Creating a development system	192
13.2	Modifying a package for l warp	194
13.2.1	Adding a package to the l warp.dtx file	195
13.3	Modifying a class for l warp	195
13.4	Testing l warp	195
13.5	Modifying l warpmk	196
14	Troubleshooting	197
14.1	Using the l warp.sty package	197
14.1.1	Debug tracing output	201
14.2	Compiling the l warp .dtx file	201
15	Trademarks	202
1	l warp.sty	203
16	Implementation	203
17	Section depths and HTML headings	204
18	Source code	205
19	Detecting the TeX Engine — pdflatex, lualatex, xelatex	206
20	Early package requirements	206
21	Package load order	207
21.1	Tests of package load order	207
21.2	Error for disallowed packages and classes loaded before l warp	210
21.3	Enforcing package loading after l warp	210
22	MD5 hashing	218
23	pdflATEX T1 and UTF-8 encoding	219
24	Unicode input characters	220
25	Upright quotes	220
26	Miscellaneous tools	221
27	Operating-System portability	222
27.1	Literal characters	222
27.2	Common portability code	223
27.3	UNIX, LINUX, and MAC OS	223
27.4	MS-WINDOWS	223

28 Package options	224
28.1 Conditional compilation	230
29 Required packages	231
30 Loading packages	235
31 Additional required packages	240
32 File handles	240
33 Include a file	241
34 Copying a file	242
35 Debugging messages	243
36 Defining print and HTML versions of macros and environments	244
37 HTML-conversion output modifications	249
37.1 User-level controls	249
37.2 Heading adjustments	251
38 Remembering original formatting macros	252
39 Accents	254
40 Configuration Files	256
40.1 Decide whether to generate configuration files	256
40.2 project_html.tex	256
40.3 l warpmk configuration files	257
40.3.1 Helper macros	257
40.3.2 l warpmk.conf	263
40.3.3 <project>.l warpmkconf	264
40.4 l warp.css	264
40.5 l warp_sagebrush.css	294
40.6 l warp_formal.css	299
40.7 sample_project.css	303
40.8 l warp.ist	304
40.9 l warp.xdy	304
40.10 l warp_one_limage.cmd	305
40.11 l warp_mathjax.txt	306
40.12 l warpmk.lua — l warpmk option	308
41 Stacks	326
41.1 Assigning depths	327
41.2 Closing actions	327
41.3 Closing depths	328

41.4	Pushing and popping the stack	328
42	Data arrays	330
43	Localizing catcodes	330
44	Localizing dynamic math	331
45	Sanitizing labels and filenames	333
46	HTML entities	333
47	HTML filename generation	334
48	Homepage link	337
49	\LWRPrintStack diagnostic tool	338
50	Closing stack levels	339
51	PDF pages and styles	339
52	HTML tags, spans, divs, elements	341
52.1	Mapping L ^A T _E X sections to HTML sections	341
52.2	Babel-French tag modifications	341
52.3	HTML tags	342
52.4	Block tags and comments	345
52.5	Div class and element class	346
52.6	Single-line elements	347
52.7	HTML5 semantic elements	347
52.8	High-level block and inline classes	348
52.9	Closing HTML tags	350
53	Paragraph handling	351
54	Paragraph start/stop handling	355
55	Page headers and footers	357
56	css	358
57	Title, HTML meta author, HTML meta description	359
58	Footnotes	360
58.1	Regular page footnotes	361
58.2	Minipage footnotes	361
58.3	Titlepage thanks	361
58.4	Regular page footnote implementation	361

58.5	Minipage footnote implementation	364
58.6	Printing pending footnotes	365
59	Marginpars	366
60	Splitting HTML files	367
60.1	Customizing MATHJAX	371
61	Sectioning	376
61.1	User-level starred section commands	377
61.2	Book class commands	377
61.3	Sectioning support macros	378
61.4	Pre- and post- sectioning names	386
61.5	\section and friends	386
62	Starting a new file	389
63	Starting HTML output	393
64	Ending HTML output	396
65	Title page	398
65.1	Setting the title, etc.	398
65.2	\if@titlepage	399
65.3	Changes for \affiliation	400
65.4	Printing the thanks	401
65.5	Printing the title, etc. in HTML	401
65.6	Printing the title, etc. in print form	403
65.7	\maketitle for HTML output	403
65.8	\published and \subtitle	406
66	Abstract	408
67	Quote and verse	408
67.1	Attributions	408
67.2	Quotes, quotations	409
67.3	Verse	409
67.3.1	LATEX core verse environment	410
67.3.2	verse and memoir	411
68	Verbatim and tabbing	411
69	Theorems	414
70	Lists	415
70.1	List environment	416
70.2	Itemize	419
70.3	Enumerate	420

70.4	Description	420
70.5	Patching the lists	421
71	Tabular	423
71.1	Limitations	423
71.2	Temporary package-related macros	426
71.2.1	arydshln	426
71.3	Token lookahead	426
71.4	Tabular variables	427
71.5	Handling &, @, !, and bar	430
71.5.1	Handling &	432
71.5.2	Filling an unfinished row	433
71.6	Handling \\	435
71.7	Looking ahead in the column specifications	436
71.8	Parsing @, >, <, !, bar columns	436
71.9	Parsing 'l', 'c', or 'r' columns	440
71.10	Parsing 'p', 'm' or 'b' columns	441
71.11	Parsing 'w' columns	441
71.12	Parsing 'D' columns	442
71.13	Parsing the column specifications	442
71.14	colortbl and xpars tabular color support	447
71.15	Starting a new row	449
71.16	Printing vertical bar tags	450
71.17	Printing at or bang tags	450
71.18	Data opening tag	451
71.19	Midrules	454
71.20	Cell colors	460
71.21	Multicolumns	464
71.21.1	Parsing multicolumns	464
71.21.2	Multicolumn factored code	468
71.21.3	Multicolumn	471
71.21.4	Longtable captions	471
71.21.5	Counting HTML tabular columns	474
71.22	Multirow if not loaded	476
71.23	Multicolumnrow	476
71.24	Utility macros inside a table	477
71.25	Special-case tabular markers	478
71.26	Checking for a new table cell	479
71.27	\mrowcell	482
71.28	\mcolrowcell	483
71.29	HTML tabular environment	483
72	Cross-references	490
72.1	Setup	490
72.2	zref setup	492
72.3	Labels	493

72.4	References	495
72.5	Hyper-references	498
73	Floats	502
73.1	Float environment	503
73.2	Float tracking	505
73.3	Caption inside a float environment	506
73.4	Caption and LOF linking and tracking	507
74	Table of Contents, LOF, LOT	510
74.1	Reading and printing the toc	510
74.2	High-level TOC commands	514
74.3	Side TOC	514
74.4	Low-level TOC line formatting	516
75	Index and glossary	519
76	Bibliography presentation	522
77	Restoring original formatting	523
78	Math	525
78.1	Limitations	525
78.1.1	Rendering tradeoffs	525
78.1.2	SVG option	526
78.1.3	MATHJAX option	527
78.1.4	Customizing MATHJAX	527
78.1.5	MATHJAX limitations	527
78.1.6	Catcode changes	528
78.1.7	Complicated inline math objects	528
78.1.8	Complicated display math objects	529
78.2	HTML alt tag names	529
78.3	Inline and display math	529
78.4	MATHJAX support	541
78.5	Equation environment	543
78.6	\displaymathnormal and \displaymathother	546
78.7	AMS Math environments	548
78.7.1	Support macros	548
78.7.2	Environment patches	549
79	Lateximages	552
79.1	Description	552
79.2	Support counters and macros	553
79.3	Font size	554
79.4	Sanitizing math expressions for HTML	554
79.5	Equation numbers	556
79.6	HTML alt tags	557

79.7	<code>lateximage</code> environment	558
80	<code>center, flushleft, flushright</code>	564
81	Preloaded packages	566
82	<code>siunitx</code>	567
83	Graphics print-mode modifications	568
83.1	General limitations	568
83.2	Print-mode modifications	570
84	<code>xcolor boxes</code>	570
85	<code>chemmacros environments</code>	573
86	<code>cleveref</code>	574
87	<code>picture environment</code>	578
88	Boxes and Minipages	578
88.1	Counters and lengths	579
88.2	Footnote handling	579
88.3	Minipage handling	579
88.4	<code>\parbox, \mbox, \makebox, \framebox, \fbox, \raisebox</code>	583
89	Direct formatting	588
90	Skips, spaces, font sizes	598
91	<code>\phantomsection</code>	607
92	\LaTeX{} and other logos	607
92.1	HTML logos	607
92.2	Print logos	610
93	<code>\AtBeginDocument, \AtEndDocument</code>	611
94	KOMA-SCRIPT classes	611
95	MEMOIR class	611
96	ut* classes	612
97	CTEXpatch	613
2	lwarf-2in1.sty	615

98	2in1	615
3	lwarf-2up.sty	615
99	2up	615
4	lwarf-a4.sty	615
100	a4	615
5	lwarf-a4wide.sty	616
101	a4wide	616
6	lwarf-a5comb.sty	616
102	a5comb	616
7	lwarf-abstract.sty	616
103	abstract	616
8	lwarf-accsupp.sty	619
104	accsupp	619
9	lwarf-acro.sty	619
105	acro	619
10	lwarf-acronym.sty	621
106	acronym	621
11	lwarf-adjmulticol.sty	622
107	adjmulticol	622

12 l warp-addlines.sty	623
108 addlines	623
13 l warp-afterpage.sty	623
109 afterpage	623
14 l warp-algorithm2e.sty	624
110 algorithm2e	624
15 l warp-algorithmicx.sty	628
111 algorithmicx	628
16 l warp-alltt.sty	629
112 alltt	629
17 l warp-amsmath.sty	629
113 amsmath	629
18 l warp-amsthm.sty	636
114 amsthm	636
19 l warp-anonchap.sty	641
115 anonchap	641
20 l warp-any size.sty	641
116 anysize	641
21 l warp-appendix.sty	642
117 appendix	642

22 l warp-arabicfront.sty	642
118 arabicfront	642
23 l warp-array.sty	643
119 array	643
24 l warp-arydshln.sty	643
120 arydshln	643
25 l warp-asymptote.sty	645
121 asymptote	645
26 l warp-atbegshi.sty	646
122 atbegshi	646
27 l warp-attachfile.sty	647
123 attachfile	647
28 l warp-attachfile2.sty	648
124 attachfile2	648
29 l warp-authblk.sty	650
125 authblk	650
30 l warp-axessibility.sty	651
126 axessibility	651
31 l warp-axodraw2.sty	652
127 axodraw2	652

32 lwarf-backref.sty	652
128 backref	652
33 lwarf-balance.sty	652
129 balance	652
34 lwarf-biblatex.sty	653
130 biblatex	653
35 lwarf-bibunits.sty	653
131 bibunits	653
36 lwarf-bigdelim.sty	654
132 bigdelim	654
37 lwarf-bigstrut.sty	655
133 bigstrut	655
38 lwarf-bitpattern.sty	655
134 bitpattern	655
39 lwarf-blowup.sty	656
135 blowup	656
40 lwarf-booklet.sty	656
136 booklet	656
41 lwarf-bookmark.sty	657
137 bookmark	657

42 l warp-booktabs.sty	657
138 booktabs	657
43 l warp-bophook.sty	659
139 bophook	659
44 l warp-bounddvi.sty	659
140 bounddvi	659
45 l warp-boxedminipage2e.sty	659
141 boxedminipage2e	659
46 l warp-breakurl.sty	660
142 breakurl	660
47 l warp-breqn.sty	660
143 breqn	660
48 l warp-bsheaders.sty	662
144 bsheaders	662
49 l warp-bxpaper size.sty	662
145 bxpaper size	662
50 l warp-bytefield.sty	663
146 bytefield	663
51 l warp-cancel.sty	663
147 cancel	663

52 lwarf-canoniclayout.sty	664
148 canoniclayout	664
53 lwarf-caption.sty	664
149 caption	664
54 lwarf-cases.sty	667
150 cases	667
55 lwarf-changebar.sty	668
151 changebar	668
56 lwarf-changepage.sty	669
152 changepage	669
57 lwarf-chngpage.sty	669
153 chngpage	669
58 lwarf-chappg.sty	670
154 chappg	670
59 lwarf-chapterbib.sty	670
155 chapterbib	670
60 lwarf-chemfig.sty	670
156 chemfig	670
61 lwarf-chemformula.sty	672

157 chemformula	672
62 lwarf-chemgreek.sty	677
158 chemgreek	677
63 lwarf-chemmacros.sty	677
159 chemmacros	677
159.1 Changes to the user's document	678
159.2 Code	678
159.3 Loading modules	678
159.4 New environments	679
159.5 Acid-base	679
159.6 Charges	681
159.7 Nomenclature	682
159.8 Particles	684
159.9 Phases	685
159.10 Mechanisms	685
159.11 Newman	688
159.12 Orbital	688
159.13 Reactions	689
159.14 Redox	689
159.15 Scheme	690
159.16 Spectroscopy	691
159.17 Thermodynamics	695
64 lwarf-chemnum.sty	697
160 chemnum	697
65 lwarf-chkfloat.sty	698
161 chkfloat	698
66 lwarf-cite.sty	698
162 cite	698
67 lwarf-CJK.sty	699

163 CJK	699
68 l warp-CJKutf8.sty	699
164 CJKutf8	699
69 l warp-clrdblp g.sty	700
165 clrdblp g	700
70 l warp-cmdtrack.sty	700
166 cmdtrack	700
71 l warp-color.sty	700
167 color	700
72 l warp-colortbl.sty	700
168 colortbl	700
73 l warp-continue.sty	702
169 continue	702
74 l warp-copyrightbox.sty	703
170 copyrightbox	703
75 l warp-crop.sty	703
171 crop	703
76 l warp-cuted.sty	704
172 cuted	704

77 l warp-cutwin.sty	704
173 cutwin	704
78 l warp-dblfloatfix.sty	705
174 dblfloatfix	705
79 l warp-dblfnote.sty	705
175 dblfnote	705
80 l warp-dcolumn.sty	706
176 dcolumn	706
81 l warp-diagbox.sty	706
177 diagbox	706
82 l warp-dprogress.sty	708
178 dprogress	708
83 l warp-draftcopy.sty	708
179 draftcopy	708
84 l warp-draftfigure.sty	708
180 draftfigure	708
85 l warp-draftwatermark.sty	709
181 draftwatermark	709
86 l warp-easy-todo.sty	709
182 easy-todo	709

87 lwarf-ebook.sty	710
183 ebook	710
88 lwarf-ellipsis.sty	711
184 ellipsis	711
89 lwarf-embrac.sty	711
185 embrac	711
90 lwarf-emptypage.sty	712
186 emptypage	712
91 lwarf-endfloat.sty	712
187 endfloat	712
92 lwarf-endheads.sty	713
188 endheads	713
93 lwarf-endnotes.sty	713
189 endnotes	713
94 lwarf-enumerate.sty	714
190 enumerate	714
95 lwarf-enumitem.sty	714
191 enumitem	714
96 lwarf-epigraph.sty	715
192 epigraph	715

97 l warp-epsfig.sty	716
193 epsfig	716
98 l warp-epstopdf.sty	717
194 epstopdf	717
99 l warp-epstopdf-base.sty	717
195 epstopdf-base	717
100 l warp-errata.sty	719
196 errata	719
101 l warp-eso-pic.sty	720
197 eso-pic	720
102 l warp-everypage.sty	721
198 everypage	721
103 l warp-everyshi.sty	721
199 everyshi	721
104 l warp-extramarks.sty	722
200 extramarks	722
105 l warp-fancybox.sty	722
201 fancybox	722
106 l warp-fancyhdr.sty	728

202	fancyhdr	728
107 l warp-fancyref.sty		729
203	fancyref	729
108 l warp-fancytabs.sty		730
204	fancytabs	730
109 l warp-fancyvrb.sty		730
205	fancyvrb	730
110 l warp-figcaps.sty		737
206	figcaps	737
111 l warp-figsize.sty		737
207	figsize	737
112 l warp-fix2col.sty		738
208	fix2col	738
113 l warp-fixme.sty		738
209	fixme	738
114 l warp-fixmetodonotes.sty		739
210	fixmetodonotes	739
115 l warp-flafter.sty		740
211	flafter	740

116 l warp-flippdf.sty	740
212 flippdf	740
117 l warp-float.sty	740
213 float	740
118 l warp-floatflt.sty	742
214 floatflt	742
119 l warp-floatpag.sty	743
215 floatpag	743
120 l warp-floatrow.sty	743
216 floatrow	743
121 l warp-fltrace.sty	749
217 fltrace	749
122 l warp-flushend.sty	749
218 flushend	749
123 l warp-fnbreak.sty	750
219 fnbreak	750
124 l warp-fncychap.sty	750
220 fncychap	750
125 l warp-fnlineno.sty	751
221 fnlineno	751

126 l warp-fnpos.sty	751
222 fnpos	751
127 l warp-fontenc.sty	751
223 fontenc	751
128 l warp-footmisc.sty	752
224 footmisc	752
129 l warp-footnote.sty	753
225 footnote	753
130 l warp-footnotehyper.sty	754
226 footnotehyper	754
131 l warp-footnoterange.sty	754
227 footnoterange	754
132 l warp-footnpag.sty	755
228 footnpag	755
133 l warp-forest.sty	755
229 forest	755
134 l warp-framed.sty	756
230 framed	756
135 l warp-ftnright.sty	758
231 ftnright	758

136 l warp-fullminipage.sty	758
232 fullminipage	758
137 l warp-fullpage.sty	759
233 fullpage	759
138 l warp-fullwidth.sty	759
234 fullwidth	759
139 l warp-fwlw.sty	759
235 fwlw	759
140 l warp-gentombow.sty	760
236 gentombow	760
141 l warp-geometry.sty	760
237 geometry	760
142 l warp-glossaries.sty	761
238 glossaries	761
143 l warp-gmeometric.sty	763
239 gmeometric	763
144 l warp-graphics.sty	763
240 graphics	763
240.1 Graphics extensions	764
240.2 Length conversions and graphics options	764
240.3 Printing HTML styles	767
240.4 \includegraphics	768

240.5 Boxes	773
145 l warp-graphicx.sty	777
241 graphicx	777
146 l warp-grffile.sty	777
242 grffile	777
147 l warp-grid.sty	777
243 grid	777
148 l warp-grid-system.sty	778
244 grid-system	778
149 l warp-gridset.sty	778
245 gridset	778
150 l warp-hang.sty	779
246 hang	779
151 l warp-hanging.sty	780
247 hanging	780
152 l warp-hypcap.sty	781
248 hypcap	781
153 l warp-hypdestopt.sty	782
249 hypdestopt	782

154 l warp-hypernat.sty	782
250 hypernat	782
155 l warp-hyperref.sty	782
251 hyperref	782
156 l warp-hyperxmp.sty	788
252 hyperxmp	788
157 l warp-hyphenat.sty	788
253 hyphenat	788
158 l warp-idxlayout.sty	789
254 idxlayout	789
159 l warp-ifoddpage.sty	790
255 ifoddpage	790
160 l warp-imakeidx.sty	790
256 imakeidx	790
161 l warp-indentfirst.sty	795
257 indentfirst	795
162 l warp-index.sty	795
258 index	795
163 l warp-intopdf.sty	797
259 intopdf	797

164 l warp-keyfloat.sty	797
260 keyfloat	797
165 l warp-layaureo.sty	799
261 layaureo	799
166 l warp-layout.sty	800
262 layout	800
167 l warp-leading.sty	800
263 leading	800
168 l warp-letterspace.sty	800
264 letterspace	800
169 l warp-lettrine.sty	801
265 lettrine	801
170 l warp-lineno.sty	802
266 lineno	802
171 l warp-lips.sty	804
267 lips	804
172 l warp-listings.sty	805
268 listings	805
173 l warp-lltjext.sty	809

269 l t jext	809
174 l warp-longtable.sty	810
270 longtable	810
175 l warp-lscape.sty	812
271 l scape	812
176 l warp-ltablex.sty	812
272 l tablex	812
177 l warp-ltcaption.sty	813
273 l tcaption	813
178 l warp-ltxgrid.sty	813
274 l txgrid	813
179 l warp-ltxtable.sty	814
275 l txtable	814
180 l warp-lua-check-hyphen.sty	814
276 lua-check-hyphen	814
181 l warp-lua-visual-debug.sty	814
277 lua-visual-debug	814
182 l warp-luacolor.sty	815
278 luacolor	815

183 l warp-luatodonotes.sty	815
279 luatodonotes	815
 184 l warp-magaz.sty	817
280 magaz	817
 185 l warp-makeidx.sty	818
281 makeidx	818
 186 l warp-marginal.sty	818
282 marginal	818
 187 l warp-marginfit.sty	818
283 marginfit	818
 188 l warp-marginfix.sty	819
284 marginfix	819
 189 l warp-marginnote.sty	819
285 marginnote	819
 190 l warp-mcaption.sty	820
286 mcaption	820
 191 l warp-mdframed.sty	820
287 mdframed	820
287.1 Limitations	820
287.2 Package loading	821
287.3 Patches	821
287.4 Initial setup	822

287.5 Color and length HTML conversion	822
287.6 Environment encapsulation	823
287.7 Mdframed environment	824
287.8 Titles and subtitles	825
287.9 New environments	827
192 l warp-memhfixc.sty	830
288 memhfixc	830
193 l warp-metalogo.sty	830
289 metalogo	830
194 l warp-mhchem.sty	831
290 mhchem	831
195 l warp-microtype.sty	833
291 microtype	833
196 l warp-midfloat.sty	834
292 midfloat	834
197 l warp-midpage.sty	834
293 midpage	834
198 l warp-morefloats.sty	835
294 morefloats	835
199 l warp-moreverb.sty	835
295 moreverb	835
200 l warp-mparhack.sty	836

296 mparhack	836
201 l warp-multicol.sty	837
297 multicol	837
202 l warp-multicolrule.sty	838
298 multicolrule	838
203 l warp-multirow.sty	838
299 multirow	838
299.1 Multirow	839
299.2 Combined multicolumn and multirow	840
204 l warp-multitoc.sty	842
300 multitoc	842
205 l warp-musicography.sty	842
301 musicography	842
206 l warp-nameref.sty	844
302 nameref	844
207 l warp-natbib.sty	844
303 natbib	844
208 l warp-nccfancyhdr.sty	845
304 nccfancyhdr	845
209 l warp-needspace.sty	846
305 needspace	846

210 l warp-nextpage.sty	846
306 nextpage	846
211 l warp-nicefrac.sty	847
307 nicefrac	847
212 l warp-nonfloat.sty	847
308 nonfloat	847
213 l warp-nonumonpart.sty	848
309 numonpart	848
214 l warp-nopageno.sty	848
310 pageno	848
215 l warp-notespages.sty	848
311 notespages	848
216 l warp-nowidow.sty	849
312 widow	849
217 l warp-ntheorem.sty	849
313 ntheorem	849
313.1 Limitations	850
313.2 Options	850
313.3 Remembering the theorem style	851
313.4 HTML cross-referencing	854
313.5 \newtheoremstyle	854
313.6 Standard styles	855
313.7 Additional objects	856
313.8 Renewed standard configuration	857
313.9 amsthm option	858

313.10 Ending a theorem	860
313.11 \NoEndMark	861
313.12 List-of	861
313.13 Symbols	861
313.14 Cross-referencing	862
218 l warp-octave.sty	862
314 octave	862
219 l warp-overpic.sty	863
315 overpic	863
220 l warp-pagegrid.sty	864
316 pagegrid	864
221 l warp-pagenote.sty	864
317 pagenote	864
222 l warp-pagesel.sty	865
318 pagesel	865
223 l warp-paralist.sty	865
319 paralist	865
224 l warp-parnotes.sty	866
320 parnotes	866
225 l warp-parskip.sty	867
321 parskip	867

226 l warp-pbox.sty	867
322 pbox	867
227 l warp-pdfcomment.sty	868
323 pdfcomment	868
228 l warp-pdflandscape.sty	868
324 pdflscape	868
229 l warp-pdfmarginpar.sty	868
325 pdfmarginpar	868
230 l warp-pdfpages.sty	869
326 pdfpages	869
231 l warp-pdfprivacy.sty	872
327 pdfprivacy	872
232 l warp-pdfrender.sty	872
328 pdfrender	872
233 l warp-pdfsync.sty	872
329 pdfsync	872
234 l warp-pdftricks.sty	873
330 pdftricks	873
235 l warp-pdfx.sty	873

331 pdfx	873
236 lwarf-pfnote.sty	874
332 pfnote	874
237 lwarf-phfqit.sty	874
333 phfqit	874
238 lwarf-placeins.sty	874
334 placeins	874
239 lwarf-plarydshln.sty	875
335 plarydshln	875
240 lwarf-plexit.sty	875
336 plexit	875
241 lwarf-plexitarydshln.sty	876
337 plexitarydshln	876
242 lwarf-plextblcolortbl.sty	876
338 plextblcolortbl	876
243 lwarf-prelim2e.sty	877
339 prelim2e	877
244 lwarf-prettyref.sty	877
340 prettyref	877

245 l warp-preview.sty	877
341 preview	877
246 l warp-psfrag.sty	878
342 psfrag	878
247 l warp-psfragx.sty	878
343 psfragx	878
248 l warp-pst-eps.sty	879
344 pst-eps	879
249 l warp-pstool.sty	879
345 pstool	879
250 l warp-pstricks.sty	880
346 pstricks	880
251 l warp-pxatbegshi.sty	880
347 pxatbegshi	880
252 l warp-pxeveryshi.sty	881
348 pxeveryshi	881
253 l warp-pxftnright.sty	881
349 pxftnright	881
254 l warp-pxjahyper.sty	881

350 pxjahyper	881
255 l warp-quotchap.sty	881
351 quotchap	881
256 l warp-quoting.sty	883
352 quoting	883
257 l warp-ragged2e.sty	883
353 ragged2e	883
258 l warp-realscripts.sty	884
354 realscripts	884
259 l warp-refcheck.sty	885
355 refcheck	885
260 l warp-register.sty	885
356 register	885
261 l warp-relsize.sty	886
357 relsize	886
262 l warp-repeatindex.sty	887
358 repeatindex	887
263 l warp-resizegather.sty	888
359 resizegather	888

264 l warp-rmpage.sty	889
360 rmpage	889
265 l warp-romanbar.sty	889
361 romanbar	889
266 l warp-romanbarpagenumber.sty	889
362 romanbarpagenumber	889
267 l warp-rotating.sty	890
363 rotating	890
268 l warp-rotfloat.sty	890
364 rotfloat	890
269 l warp-rviewport.sty	891
365 rviewport	891
270 l warp-savetrees.sty	892
366 savetrees	892
271 l warp-scalefnt.sty	892
367 scalefnt	892
272 l warp-schemata.sty	892
368 schemata	892
273 l warp-scrextend.sty	893
369 scrextend	893

274 l warp-scrhack.sty	896
370 scrhack	896
275 l warp-scrlayer.sty	896
371 scrlayer	896
276 l warp-scrlayer-notecolumn.sty	898
372 scrlayer-notecolumn	898
277 l warp-scrlayer-scrpage.sty	898
373 scrlayer-scrpage	898
278 l warp-scrpage2.sty	899
374 scrpage2	899
279 l warp-section.sty	901
375 section	901
280 l warp-sectionbreak.sty	902
376 sectionbreak	902
281 l warp-sectsty.sty	902
377 sectsty	902
282 l warp-semantic-markup.sty	903
378 semantic-markup	903
283 l warp-setspace.sty	904
379 setspace	904

284 l warp-shadow.sty	905
380 shadow	905
285 l warp-showidx.sty	906
381 showidx	906
286 l warp-showkeys.sty	906
382 showkeys	906
287 l warp-sidecap.sty	906
383 sidecap	906
288 l warp-sidenotes.sty	907
384 sidenotes	907
289 l warp-SIunits.sty	909
385 SIunits	909
290 l warp-siunitx.sty	909
386 siunitx	909
291 l warp-soul.sty	915
387 soul	915
292 l warp-soulpos.sty	917
388 soulpos	917
293 l warp-soulutf8.sty	917
389 soulutf8	917

294 l warp-splitidx.sty	918
390 splitidx	918
295 l warp-srcltx.sty	919
391 srcltx	919
296 l warp-srctex.sty	920
392 srctex	920
297 l warp-stabular.sty	920
393 stabular	920
298 l warp-stffloats.sty	921
394 stffloats	921
299 l warp-subfig.sty	921
395 subfig	921
300 l warp-subfigure.sty	926
396 subfigure	926
301 l warp-supertabular.sty	927
397 supertabular	927
302 l warp-syntonly.sty	929
398 syntonly	929
303 l warp-tables.sty	929
399 tabs	929

304 l warp-tabularx.sty	930
400 tabularx	930
305 l warp-tabulary.sty	930
401 tabulary	930
306 l warp-tascmac.sty	931
402 tascmac	931
307 l warp-textarea.sty	932
403 textarea	932
308 l warp-textcomp.sty	933
404 textcomp	933
404.1 Limitations	933
404.2 Package loading	933
404.3 HTML symbols	933
404.3.1 pdfLATEX symbols	933
404.3.2 XE LATEX and LualATEX symbols	934
404.4 HTML diacritics	935
309 l warp-textfit.sty	936
405 textfit	936
310 l warp-textpos.sty	936
406 textpos	936
311 l warp-theorem.sty	937
407 theorem	937
407.1 Remembering the theorem style	938
407.2 css patches	939

312 l warp-thinsp.sty	941
408 thinsp	941
313 l warp-threadcol.sty	942
409 threadcol	942
314 l warp-threeparttable.sty	942
410 threeparttable	942
315 l warp-thumb.sty	943
411 thumb	943
316 l warp-thumbs.sty	943
412 thumbs	943
317 l warp-tikz.sty	944
413 tikz	944
318 l warp-titleps.sty	945
414 titleps	945
319 l warp-titleref.sty	948
415 titleref	948
320 l warp-titlesec.sty	949
416 titlesec	949
321 l warp-toc.sty	951
417 toctitle	951

322 l warp-titling.sty	953
418 titling	953
323 l warp-tochbasic.sty	958
419 tochbasic	958
324 l warp-toccbibind.sty	958
420 toccbibind	958
325 l warp-tocenter.sty	960
421 tocenter	960
326 l warp-tocloft.sty	961
422 tocloft	961
327 l warp-tocstring.sty	967
423 tocstring	967
328 l warp-todo.sty	968
424 todo	968
329 l warp-todonotes.sty	969
425 todonotes	969
330 l warp-tram.sty	971
426 tram	971
331 l warp-transparent.sty	971
427 transparent	971

332 l warp-trimclip.sty	972
428 trimclip	972
333 l warp-trivfloat.sty	972
429 trivfloat	972
429.1 Combining \newfloat, \trivfloat, and algorithmicx	973
334 l warp-turnthepage.sty	974
430 turnthepage	974
335 l warp-twoup.sty	974
431 twoup	974
336 l warp-typearea.sty	974
432 typearea	974
337 l warp-ulem.sty	975
433 ulem	975
338 l warp-underscore.sty	977
434 underscore	977
339 l warp-units.sty	977
435 units	977
340 l warp-unitsdef.sty	978
436 unitsdef	978
341 l warp-upref.sty	979

437 upref	979
342 lwarf-url.sty	979
438 url	979
343 lwarf-uspace.sty	979
439 uspace	979
344 lwarf-verse.sty	980
440 verse	980
345 lwarf-versonotes.sty	982
441 versonotes	982
346 lwarf-vertbars.sty	982
442 vertbars	982
347 lwarf-vmargin.sty	983
443 vmargin	983
348 lwarf-vowel.sty	983
444 vowel	983
349 lwarf-vpe.sty	984
445 vpe	984
350 lwarf-vwcol.sty	984
446 vwcol	984

351 l warp-wallpaper.sty	986
447 wallpaper	986
352 l warp-watermark.sty	987
448 watermark	987
353 l warp-widows-and-orphans.sty	987
449 widows-and-orphans	987
354 l warp-wrapfig.sty	988
450 wrapfig	988
355 l warp-xbmks.sty	989
451 xbmks	989
356 l warp-xcolor.sty	989
452 xcolor	989
452.1 Limitations	990
452.2 xcolor definitions: location and timing	990
452.3 Package loading	992
452.4 Remembering and restoring original definitions	992
452.5 HTML color style	993
452.6 HTML border	994
452.7 High-level macros	994
452.8 Row colors	999
357 l warp-xexchangebar.sty	1001
453 xexchangebar	1001
358 l warp-xellipsis.sty	1001
454 xellipsis	1001

359 l warp-xetexko-vertical.sty	1002
455 xetexko-vertical	1002
 360 l warp-xfrac.sty	1002
456 xfrac	1002
 361 l warp-xltabular.sty	1005
457 xltabular	1005
 362 l warp-xltxtra.sty	1005
458 xltxtra	1005
 363 l warp-xmpincl.sty	1006
459 xmpincl	1006
 364 l warp-xpiano.sty	1006
460 xpiano	1006
 365 l warp-xpinyin.sty	1007
461 xpinyin	1007
 366 l warp-xtab.sty	1007
462 xtab	1007
 367 l warp-xunicode.sty	1009
463 xunicode	1009
 368 l warp-xurl.sty	1010
464 xurl	1010

369 l warp-xy.sty	1010
465 xy	1010
370 l warp-zhlineskip.sty	1011
466 zhlineskip	1011
371 l warp-zwpagelayout.sty	1011
467 zwpagelayout	1011
372 l warp-patch-komascript.sty	1013
468 patch-komascript	1013
373 l warp-patch-memoir.sty	1015
469 patch-memoir	1015
469.1 Packages	1016
469.2 Preliminary setup	1017
469.3 Laying out the page	1017
469.4 Text and fonts	1020
469.5 Titles	1020
469.6 Abstracts	1021
469.7 Document divisions	1021
469.8 Pagination and headers	1024
469.9 Paragraphs and lists	1025
469.10 Contents lists	1026
469.11 Floats and captions	1030
469.12 Page notes	1034
469.13 Decorative text	1035
469.14 Poetry	1036
469.15 Boxes, verbatims and files	1036
469.16 Cross referencing	1037
469.17 Back matter	1038
469.18 Miscellaneous	1039
469.19 ccaption emulation	1040
469.20 Final patchwork	1042
Change History	1043

470 Chg Hist **1043**

Index of Objects **1070**

General Index **1091**

Troubleshooting Index **1095**

Index of Indexes **1100**

List of Figures

1	tutorial.tex listing	98
---	----------------------	----

List of Tables

1	Typesetting conventions	78
2	L ^A T _E X–HTML generation — l warp package — Supported features	79
3	Required software programs	89
4	Files created along with the print version	100
5	l warp package options	117
6	HTML settings	121
7	Literal character macros	180
8	Section HTML headings for word-processor conversion	190
9	Section depths and HTML headings	204
10	Tabular baseline	442
11	Tabular HTML column conversions	443
12	Cross-referencing data structures	491
13	Float data structures	502
14	amsthm package — css styling of theorems and proofs	637
15	Ntheorem package — css styling of theorems and proofs	850
16	Theorem package — css styling of theorems and proofs	937

2 Updates

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

For a detailed list of the most recent changes, see the end of the Change History on page [1069](#).

v0.65: css layout, alt tags, Japanese. PENDING

- | | |
|--------------------------------------|--|
| page layout | <ul style="list-style-type: none">• Moved the sidetoc to the left side, allowing improved css for margin notes.• Improved page layout css. |
| image alt tags | <ul style="list-style-type: none">• graphicx \includegraphics: Added the alt key to assign an alt tag to an image. Default is “(image)”, assigned to pass validation. |
| duplicate HTML files | <ul style="list-style-type: none">• Detects and causes an error if duplicate HTML file names are generated, caused by identical or similar sectioning names. |
| fixes | <ul style="list-style-type: none">• Fix for tabular*.• Fix for tabular border colors.• Fixes \quad, \enskip, and figure captions to pass validation. |
| Japanese | <ul style="list-style-type: none">• Added ltj* classes, bounddvi, gentombow, lltjext, plarydshln, plex, plextarydshln, plextcolortbl, pxatbegshi, pxeveryshi, pxfnright, pxjahyper, tascmac.• Verified to work with plarray, plautopatch, plextarray, plextdelarray, pxgentombow, plsiunitx, pxpdfpages, pxpgfrcs, pxpgfmark. |
| packages | <ul style="list-style-type: none">• Added support for fontspec \texttsi and \sishape.• Added multicol's \docolaction.• Added embrac, footnoterange, multicolrule, versonotes. |

v0.64: Koma-Script, Japanese, Chinese.

- | | |
|-----------------------------|---|
| Japanese | <ul style="list-style-type: none">• Added utarticle and related classes.• Improved ujarticle and related classes. |
| Chinese | <ul style="list-style-type: none">• Fix for biblatex with CTEX and other classes. |
| Koma-Script | <ul style="list-style-type: none">• Fixes for scrlayer, scrlayer-scrpage. |
| packages | <ul style="list-style-type: none">• addlines: Updated to v0.3.• Added bsheaders, gmeometric, marginal, rmpage, scrpage2. |

v0.63: mdframed, Chinese, Japanese, Korean

- localization**
 - Added `\linkhomename`: A user-definable name for the **Home** link.
 - Documented `\sidetocname`: A user-definable name for the sidetoc.
- fixes**
 - Fix: `\LinkHome` for print output.
- optimizations**
 - Moved package load checks to the l warp core to reduce the number of `l warp-*` files.
- packages**
 - mdframed: Fix with amsthm, improved titles and font control. Improved rule widths.
- Chinese**
 - Fixes for xeCJK.
 - Added `xpinyin`, `zhlineskip`.
 - Verified to work with `cjkpunct`, `upzhkinsoku`, `zhspacing`.
- Japanese**
 - Verified to work with `zxjatype`, `luatexja`, `luatexja-fontspec`.
 - Added `bxjsarticle` and related classes.
 - Added `ltjsarticle` and related classes.
 - Added `pLATEX`, `upLATEX`, `ujarticle` and related classes.
 - Prevented `utarticle` and related classes.
 - Prevented `bxcjkatype`.
- Korean**
 - Verified to work with `kotex`, `xetexko`, `luatexko`.

v0.62: MiK_TE_X docs, HTML title, C_TE_X, xeCJK, bitpattern.

- docs**
 - Docs: Setting a UTF-8 locale. See section [10.8](#).
- MiK_TE_X**
 - MiK_TE_X: Docs for *MiK_TE_X Console* and *miktex-poppler-bin*.
- HTML <title>**
 - HTML subplot titles: Added `\HTMLTitleBeforeSection` and `\HTMLTitleAfterSection` to select whether the HTML `<title>` displays the website name before or after the section name. See section [8.4](#).
- fixes**
 - Fix for package options handling.
 - Fixes for horizontal white space between `fminipage`, `fcolorminipage`, `colorboxBlock`, `fcolorboxBlock`.
 - Logos: Fix for X_HT_EX logo, improved css, made robust, improved search-engine optimization.
 - `\\\[$1]`: Additional HTML `
` if $$1 > 0 \text{ pt}$.
 - Fixes for `\includgraphics` filename, and with `FormatWP`.
 - Fix: css for `\textup`.
 - Fix: Added `\slshape`.
- Chinese**
 - Added `ctex` package and related classes, xeCJK.
 - Prevented CJK, CJKutf8 unless xeCJK, ctex are used.

- packages**
- chemfig: Docs for new macro `\polymerdelim`.
 - asymptote: Docs for compilation.
 - chngpage: Fix to load l warp-changepage.
 - algorithm2e: Fix with non-book classes.
 - register: Updated to v1.8.
 - nicefrac: Improved font control and css, honors `nice` and `ugly`.
 - units: Improved font control and css, honors `tight` and `loose`.
 - xfrac: Improved css.
 - textcomp and xunicode: Fix conflicts with `\textcircled`.
 - ulem: Improved compatibility with CJKulem, lateximage.
 - MathJax and siunitx: Removed inoperable extension.
 - Added bitpattern, pdfcomment, pdfmarginpar, tram, unitsdef, xchangebar.
 - Added musicography, octave, semantic-markup.
 - Added 2in1, flippdf, notespages, rviewport, twoup.
- v0.61:** Custom compilation, EPS-related packages, documentation, indexes.
- docs**
- Split index into multiple indexes. See page 1100.
 - Improved documentation regarding font selection. See section 8.2.
 - Added documentation regarding debugging options. See section 35.
 - Added documentation regarding HTML entities inside program listings. See section 9.2.1.
- custom compiling**
- Added options to specify the shell commands to execute for `l warpmk print` and `l warpmk html`, allowing the use of l warp with perltex, pythontex, etc. If not specified, these are set automatically depending on the LATEX engine, `--shell-escape`, and l warp options. See section 10.
- ⚠ changed names**
- | Old | New |
|--------------------------------|--------------------------------|
| <code>\StartDynamicMath</code> | <code>\inlinemathother</code> |
| <code>\StopDynamicMath</code> | <code>\inlinemathnormal</code> |
- fixes**
- Fix: Paragraph tags in a tabular.
 - Fix: supertabular and xtab captions.
 - Fix: DVI LATEX `\includegraphics` EPS images.
 - Fix: newfloat lists.
 - Fix: css footnotes text align, minipage tabular and footnote margins.
- packages**
- Added epsfig, psfrag, psfragx, pstool.
 - Added copyrightbox, pdfprivacy, thinspace, threadcol, uspace.

- Added `chkfloat`, `cmdtrack`, `dprogress`, `lua-visual-debug`, `refcheck`, `srcltx`, `srctex`, `vpe`, `xbmks`.

v0.60: Fixes for `longtable`, `listings`.

- | | |
|----------|--|
| fixes | <ul style="list-style-type: none"> • <code>longtable</code>, etc.: Fixes for slowdown and memory management for very long tables. • <code>listings</code>: Fix for HTML entities, and also when used inside a list. • <code>diagbox</code>: Fix for incorrect HTML <code>par</code> tags. |
| packages | <ul style="list-style-type: none"> • Added <code>2up</code>, <code>booklet</code>. • Added <code>bophook</code>, <code>draftfigure</code>, <code>fullminipage</code>, <code>grid-system</code>, <code>layaureo</code>. • Added <code>leading</code>, <code>widows-and-orphans</code>. • Added <code>fancytabs</code>, <code>thumb</code>, <code>thumbs</code>. |

v0.59: DVI `latex`, MATHJAX, `asymptote`, `pdfricks` and `pstricks`, `epstopdf`, `breqn`.

⚠ Reset the configuration

- | | |
|------------------|--|
| <i>lwarpmk</i> | <ul style="list-style-type: none"> • Due to changes in <code>lwarpmk</code>, recompile any existing project a single time using <code>pdflatex filename.tex</code> or similar, after which <code>lwarpmk</code> may then be used with the new configuration files. |
| <i>dvi latex</i> | <ul style="list-style-type: none"> • Added an error if <code>lwarpmk.conf</code>'s format has changed and the document must be recompiled. |
| <i>latexmk</i> | <ul style="list-style-type: none"> • Added a warning if the <code>lwarpmk.conf</code> configuration file appears to be for the wrong operating system, in case files are transferred between systems. |
| <i>math</i> | <ul style="list-style-type: none"> • Added <pre>lwarpmk epstopdf <list-of-EPS-files></pre> to quickly convert a document's EPS images to PDF or SVG. See section 9.7. |
| <i>images</i> | <ul style="list-style-type: none"> • Added support for DVI <code>latex</code>. See section 8.3. • Fix for <code>--shell-escape</code> with <code>latexmk</code>. • Updated MathJax script to v2.7.4. • Fix: Mathjax chapter number removed from non-numeric tagged equations. • Added MathJax support for <code>nicefrac</code>, <code>units</code>. • Fix for <code>\[</code> and <code>\]</code> with <code>\displaymathnormal</code>. • Fix for <code>\includegraphics</code> filename expansion. • <code>\includegraphics</code> now works with <code>.pdf</code> and <code>.eps</code> filename extensions. |
| <i>packages</i> | <ul style="list-style-type: none"> • Moved <code>amsmath</code> out of the <code>lwarp</code> core. • Fix for <code>chemformula \NMR</code>. • Added <code>asymptote</code>, <code>pdfricks</code>, <code>pstricks</code>, <code>pst-eps</code>. • Added <code>breqn</code>, <code>Slunits</code>. • Added <code>bxpapersize</code>, <code>canoniclayout</code>, <code>draftcopy</code>, <code>fnbreak</code>, <code>nccfancyhdr</code>. |

- Added accsupp, axessibility.
 - Added xunicode.
 - Improved and now supports epstopdf.
 - Tested to work as-is: eepic, sepfootnotes.
- docs • Added information about setting up a development version of lwarf.

v0.58: Extensive improvements in indexing, glossaries. Adds PDF-inclusion packages.

⚠ Reset the configuration

- Due to changes in *lwarpmk*, **recompile any existing project a single time** using *pdflatex filename.tex* or similar, after which *lwarpmk* may then be used with the new configuration files.

lwarpmk

glossaries

- *lwarpmk*: Added the -p option to specify the project name.
- *lwarpmk*: Now uses *makeglossaries* for glossary generation, allowing the processing of multiple glossaries at once.
- Added lwarf option *GlossaryCmd* to specify the shell command used by *lwarpmk printglossary* and *lwarpmk htmlglossary*. Defaults to *makeglossaries*.

index and glossary

- Docs: Extra indexing options. See section 9.5.11.
- Added support for *makeindex*. (Previously supported only *xindy*.) Also added indexing packages listed below.
- Added lwarf options *PrintIndexCmd*, *HTMLIndexCmd*, and *LatexmkIndexCmd* to specify shell commands used by *lwarpmk printindex*, *lwarpmk htmlindex*, and *latexmk*. May be preset with the *makeindex* or *xindy* lwarf options. See section 8.3.

• Added lwarf options *makeindex* and *xindy* to set *PrintIndexCmd*, *HTMLIndexCmd*, and *LatexmkIndexCmd* to sensible values for a typical single index. See section 8.3.

• Added lwarf option *makeindexStyle* to tell *lwarpmk* to use a custom style instead of *lwarf.ist*. See section 9.5.17.

• Fix for index entries with \see, \seealso, \emph, \textbf, etc.

• Replaced each \csuse with \cnameuse for improved error detection.

• Additional internal print/HTML macro selection improvements.

• Fix: \printindex finishes pending \index writes first.

• Fixes for memoir: makeidx, ccaption, multiple indexes, \specialindex.

• Fixes for komascript: Indexing improvements.

• Added imakeidx, index, repeatindex, splitidx.

• Added attachfile, attachfile2, intopdf, pdfpages, pdfx.

• Added cases.

• Tested to work as-is: notes2bib, hvindex.

misc. fixes

packages

v0.57: algorithm2e, float styles, tabular packages, internal improvements.

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

$$(-xy- \text{ diagram})$$
 - See section 8.4.
 - misc. fixes**
 - Fix: Improved print/HTML macro selection.
 - Fix: `\href` text catcodes.
 - Fix: `\subref` text.
 - Fixes: Colored `\rule` and `\boxframe`.
 - packages**
 - `float`, `rotfloat`: Adds support for float styles `ruled` and `boxed`.
 - `float`: Fix: Do not create `\l@<type>` until `\listof` is used.
 - `marginnote`: Fix: Long optional argument.
 - `ellipsis`: Adds `\midwordellipsis`.
 - `breakurl`: Fix for text catcodes.
 - Added `algorithm2e`, `register`, `ltablex`, `xltabular`, `xellipsis`, `trimclip`, `errata`, `vowel`, `xpiano`.
 - Prevents glossary.
 - Tested to work as-is with `gauss`, `phonrule`, `piano`, `Slunits`, `tikzcodeblocks`.
- v0.56:** Shell escape, tabular packages.
- lwarfmk**
 - Added

$$\text{lwarfmk pdftosvg } <\text{list-of-PDF-files}>$$
 - to quickly convert a document's PDF images to SVG, for use with HTML.
 - See section 9.7.
 - tabular**
 - Added support for `--shell-escape`. See section 8.1.
 - Added support for array w and W columns.
 - Fix: `\multicolumn` parameter handling.

- Added support for double `\hlines`, `\midrules`, and vertical rules.
- Added support for `arydshln` dashed lines with `HTML tabular`, but reverts to plain rules for `lateximage` and `SVG math array`.

misc. fixes

- Fix: `\thinspace`.

- Fix: `paralist compact environments`.

packages

- Added `parnotes`, `quoting`, `lua-check-hyphen`, `tocenter`, `underscore`.

- Tested to work as-is with `babelbib`, `bibunits`, `bodegraph`, `fast-diagram`, `nicematrix`, `structmech`.

v0.55: Various fixes.**misc fixes**

- Fix: Extraneous space in file links, which also prevented *Calibre EPUB* conversions.

- Fix: Float optional argument regression.

- Fix: `\ForceHTMLTOC` with `\phantomsection`.

- Fix: Overfull boxes in `lateximages`.

- Fix: QED symbols in `lateximage`.

packages

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

- Added `clrdblpg`.

v0.54: Float `\centering`, improved image checks.**⚠ Reset the configuration**

- Due to changes in `lwarfpmk`, **recompile any existing project a single time** using `pdflatex filename.tex` or similar, after which `lwarfpmk` may then be used with the new configuration files.

lwarfpmk

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

- `lwarfpmk`: Improved error message if configuration file does not exist.

BIBTEX

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

polyglossia

- Added documentation regarding polyglossia. See section 9.13.4.

macros in section names

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

document encoding

- Renamed and added package options:

Old Package Option	New Package Option
<code>xdyFilename</code>	<code>xindyStyle</code>
<code>IndexLanguage</code>	<code>xindyLanguage</code>
--	<code>xindyCodepage</code>
--	<code>pdftotextEnc</code>

⚠ New and revised encoding options

floats with \centering, etc.

- Use these options along with inputenc or inputenx to process documents in an encoding other than UTF-8. See section 8.2.
- Floats now honor \centering, \raggedright, \raggedleft, and their ragged2e equivalents, when placed directly after:

```
\begin{floattype}
\centering
```

misc. fixes

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



packages

- memoir: Minor update for v3.7g.
- Added inputenx, bibunits, chngpage, forest, magaz, gridset.
- Prevents loading ae, aecc, t1enc, and wasysym.

v0.53: Improved image checks.

l warpmk

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

v0.52: Improved footnotes, SVG math.

documentation

- Improved install instructions regarding *l warp_baseline_marker.png*.
- Added documentation regarding footnotes in section headings, and footnotes with \VerbatimFootnotes from fancybox, fancyvrb. See section 9.4.4.
- Added documentation regarding font selection when using XeL^AT_EX or L^Au^LT_EX with fontspec and traditional font packages. See section 8.2.

SVG math

- Fix: Limit the number of background tasks when generating *lateximages*.
- Added user-adjustable SVG math font scaling. See section 79.3.
- Added warnings if *l warp_baseline_marker.png* is not present, or if graphicx or graphics is not loaded.
- Improved \ensuremath hashing expansion.
- Fix: *equation** with *split*.

- tabbing now works inside a `\teximage`. Use for math in `\tabbing`.
 - MathJax
 - Fix: MathJax script was not executing in some conditions.
 - Added `\CustomizeMathJax` to add custom functions. See section [9.6](#).
 - footnotes
 - Fix: Footnote numbering when using `\HTMLDebugComments`.
 - Fix: Footnote paragraph tags.
 - Fix: `\FootnoteDepth` defaults to `\subsubsection`.
 - misc. fixes
 - Fix: `\kill` in a `\teximage`.
 - Fix: `\FileDepth`, misc. others, when input encoding is not `utf8`.
 - Fix: `\texorpdfstring` in a section name.
 - packages
 - hyperref emulation: Fix for #, %, &, ~, _ characters in URLs.
 - fancybox, fancyvrb: Initial support for `\VerbatimFootnotes`.
 - nicefrac: Added with fix for `\ensuremath`.
 - graphicx: Fix for option defaults. Added v1.1a/b options.
 - endfloat: Updated for v2.6.
 - url: Fixes for active characters.
- v0.51:** Improved svg math, added numerous chemistry packages.
- documentation
 - Docs: Added [Things to avoid](#).
 - Docs: Added to [Converting an existing document](#).
 - Docs: Multiple authors and affiliations with custom classes. See section [9.5.1](#).
 - Docs: tikz with matrices. See section [9.7.1](#).
 - SVG math
 - Improved svg math baseline.
 - Improved svg math font and color.
 - Faster svg math rendering.
 - Improved support for display math containing complicated math objects, such as tikz-cd. See section [9.6.8](#).
 - Fix: `\addcontentsline` inside SVG math.
 - Fix: SVG math containing an embedded `\teximage`.
 - MathJax
 - MathJax now handles `\ensuremath` in expressions.
 - misc. fixes
 - Fix: Added `\alignat` environment.
 - Fix: `\afterpackage` no longer required, which conflicted with `\scrfile`.
 - Fix: titling `\thanks` mark.
 - Fix: fancybox improvements.
 - Fix: tikz `\tikz` macro. (Previously only the `tikzpicture` environment worked.)

- Fix: tikz with optional argument.
- packages**
- Added mhchem, chemfig, chemformula, chemmacros, chemnum, chemgreek, epstopdf-base, grid, ltxgrid.

v0.50: Improved SVG math.

- svg math**
- SVG math and other `lateximages` now are converted to SVG using parallel background tasks, utilizing all available CPU cores.
 - Inline SVG math image file names now are MD5 hashes made from their source L^AT_EX code. Identical inline math expressions, such as multiple instance of $\$ \$$, now share a single image file. This reduces the number of images to store, transmit, process, and display. Each image file is only converted to SVG a single time, and reused if it already exists. Display math and other forms of SVG image such as `picture` and `Tikz` still use individual image files which are recreated each time `l warpmk l images` is run.
 - Fixes: SVG math and/or `\underline` in a sectioning file name.
 - Improved SVG display math and tags.
 - Improved SVG math and `siunitx` alt tags.
 - Improved `siunitx` units.
 - Fix: `\ensuremath` with MathJax now creates a `lateximage`.
 - Fix: `\centering`, etc. in SVG math, `lateximage`, `Tikz`.
- misc. fixes**
- Fix: Made various macros robust, additionally fixing `authblk`.
 - Fix: `ntheorem` if neither `standard` nor `amsthm` selected.
 - Fix: `listings`: Improved column alignment.
 - Fix: Load `fontspec` if necessary.
- packages**
- Added `xy`, `epstopdf`, `diagbox`, `pbox`, `bytefield`, `axodraw2`, `phfqt`, `schemata`, `dblfloatfix`, `nonfloat`, `morefloats`.

v0.49:

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

v0.48:

- documentation**
 - Added some documentation regarding converting an existing document. See section [7](#).
- cleveref**
 - Updated compatibility for new cleveref v0.21.
- tabular**
 - Fix: Ignores optional tabular column arguments.
- minor updates**
 - Added `\leftline`, `\centerline`, `\rightline`.
 - Lists have improved font control via `\makelabel`.
 - Print-mode `lateximage` now boxed to the natural width of its multiline contents.
 - `abstract` now allows an optional name, as required by some classes.
- math**
 - Fix: Improved spacing, `\mbox`, and font sizes with SVG math, Tikz.
 - `siunitx`: Improved SVG math, fraction compatibility, color output.
- misc. fixes**
 - Fix: LOF/LOT links.
 - Fix: Virtual page size grouping caused excessive PDF page breaks.
 - Fix: Parsing similar package names in a single `\usepackage`.
 - Fix: Adapts to classes without `\part`.
 - Fix: `\newline` in `\title` was causing `
` in window title.
 - Fix: `\maketitle` with `\cr`, `\crcr`, `\noalign`, for IEEETran class.
 - Fix: `xfrac` neutralized `BlockClass` and others.
 - Fix: `todonotes` and `luatodonotes`: Improved `\todototoc`.
- packages**
 - Added `colortbl`, `chapterbib`, `acro`, `acronym`, `hypernat`, `hypcap`, `stfloats`, `vmargin`, `fancyheadings`.
 - `fancyref`: Now directly supported.

v0.47:

- math**
 - Improved SVG math baseline and sizing.
 - Fixes: `svgmath` in captions, subcaptions, `\nameref`.
 - Fixes: Line wrap at hyphen in HTML output.
- packages**
 - Added `endheads`, `multitoc`, `sectionbreak`, `blowup`, `xurl`.

v0.46:

- ⚠ name change**
 - `\PrintStack` changed to `\LWRPrintStack`.
- misc. fixes**
 - Fix: Empty lines between `tabular` rows.
 - Fix: Stack unnesting.
 - Fix: SVG math and `lateximages` in numerous situations.
 - Fix: Spaces in `\usepackage`.
 - Fix: Now allows MATHJAX inside `verse`.

v0.45:

- documentation**
 - Improved *MiKTeX* install instructions.
 - Improved graphics and epstopdf instructions.
 - Updates to the [Introduction](#).
- memoir**
 - Added memoir, memhfixc. See section [9.12](#).
- cross-references**
 - Fix: Now allows underscores in labels.
 - Fix: _ and \<blank> in section/file names.
- math**
 - Fix: Now allows MATHJAX inside tabbing.
- bibliography**
 - Fix: Bibliography \em names.
 - Added cite, natbib, backref. (Also works as-is with biblatex.)
- misc. fixes**
 - Fix: Empty lines between tabular rows.
 - Fix: “Improper \prevdepth” with minipages, lists.
 - Fix: Incorrect svg math and lateximages with subfig.
 - Fix: Lateximages from incorrect pages with Mathjax.
 - Fix: Missing sidetoc if using listings.
 - Fix: Added an array emulation package.
- packages**
 - Added subfigure, prettyref, hanging, midpage, flafter, fltrace, changebar, endfloat, continue, fwlw, turnthepage, footnpag, pagesel, textfit, titleref.

v0.44:

- koma-script**
 - Added koma-script classes (except scrlltr2, scrjura).
 - Added scrextend, scrlayer, scrlayer-notecolumn, scrlayer-scrpage, scrhack, tocstyle, tocbasic.
- HTML title and author**
 - Added \HTMLTitle. Fixed web page title if \HTMLTitle empty and no \title given and not using titling package.
 - Fixed web page author if \HTMLAuthor is empty and \author is not given.
- encodings**
 - If using pdflatex, automatically loads T1 and UTF-8 encodings. (Additional fontenc encodings may be loaded after l warp.)
- lists**
 - Added list and trivlist environments, hang.
- tabular**
 - Fix: \multicolumn alignment if formatting for a word processor.
 - Added ltxtable.
- math**
 - Fix: MATHJAX combined with lateximages.
 - algorithmicx: Improved comment symbol and floating.
- packages**
 - Completed todonotes and luatodonotes.
 - Added todo, easy-todo, fixmetodonotes, fixme.
 - Added soulutf8, soulpos, cancel.

- Added section, fancyref, ifoddpage.
- Added preview, atbegshi, watermark.
- Improved tocloft \newlistof and \newlistentry.

v0.43:

- footnotes** • Added FootnoteDepth to control the placement of pending footnotes before section breaks. By default, pending footnotes are printed before each \subparagraph or higher.
- sectioning** • Fix: Expansion in section name.
- tabular** • Fix: Ignore spaces in tabular column specification.
• Fix: Tabular rules at bottom or when finishing incomplete rows.
• Fix: \multicolumn at/bang/before/after specifications, trim, and vertical rules.
• Fix: supertabular and xtab column misalignment.
- math** • Fix: equation*.
• Fix: SVG math in a section name.
• Fix: \ref and \eqref in SVG math.
- packages** • Added todonotes and luatodonotes (but only disabled).
• Added breakurl.
• hyperref: Fix: Several macros were made robust, \Gauge added.

v0.42:

- Support TeX!** • Added TeX development support page, [Supporting TeX development](#).
- word-processor conversion** • Improved assistance for word-processor conversions when boolean FormatWP is set true. See section 12.
 - The boolean FormatWordProcessor has been renamed FormatWP.
 - The boolean HTMLMarkFloats has been renamed WPMarkFloats.
 - New booleans control whether to place additional marks around mini-pages, at the table of contents, at the LOF and LOT, and whether to print math as LATEX source for copy/paste into the LibreOffice Writer TeXMaths extension.
 - Improved formatting for numerous objects. See section 12.
- tabbing** • Add: tabbing environment.
- overpic** • Add: overpic package. See section 315.
- math** • Fix: Text copy/paste of AMS math environment numbers and names.
• Improved \ensuremath.

- MATHJAX with siunitx: Updated script and documentation.
- symbols textcomp: Improved \interrobangdown.
- realscripts: Fix for subscripts in a lateximage.
- load order morewrites: Enforces loading before l warp.

v0.41:

- tabular • Added tabular vertical rules, subject to some limitations. See the rules section of section 9.9.
- Improved booktabs: Width and trim are honored.
- Added \mcolrowcell for empty cells inside a \multicolumnrow. Use \mcolrowcell instead of \mrowcell for two-dimensional cells created by \multicolumnrow. Continue to use \mrowcell for empty cells in a \multirow. See section 299.2 on page 840.
- Fix: Unfinished tabular rows are automatically filled.
- Fix for tabular column specifiers while using babel-french. (\NoAutoSpacing is activated then nullified inside the tabular, due to a conflict with the tabular column parsing code.)

v0.40:

- graphics, graphicx • graphics and graphicx have been moved from the l warp core, and are only loaded if requested with \usepackage.
- \includegraphics path • Improved graphics \graphicspath support. Multiple image directories may now be used. Refer to .pdf files without a file extension to allow the HTML version to use a .svg, .png, .jpg, or .gif version instead. See section 9.7.
- △ image file extensions
- bigdelim • grffile is now directly supported instead of emulated.
- symbols • Fix for bigdelim, and improved documentation. See section 132.
- fixes • Improved LATEX and textcomp symbols.
- Fix for LATEX logos and \InlineClass, etc. inside a lateximage.
- Fix for xltextra with XE LATEX.
- Fixes for tocbibind with \simplechapter, etc.
- Fixes for \multicolumnrow and \nullfonts with older versions of multirow and xparse.
- Added \underline.
- margins • Added adjmulticol.
- columns • Added cuted, midfloat.
- footnotes • Added pfnote, fnpos, dblfnote.
- tabular • Added stabular, tabls.
- sectioning • Added sectsty, anonchap, quotchap.

v0.39:[title pages](#)

[`\published` and `\subtitle`](#)
[`\loadorder`](#)
[tabular](#)

[multi column/row cell](#)

[macros inside tabular](#)

[tabular defined inside another environment](#)

[tabular](#)[margins](#)[page layout](#)

- Improved the `titlepage` HTML code, `\thanks` notes, and `\maketitle`. `titling` is no longer required, but is still supported. The `\published` and `\subtitle` fields are no longer provided, but `\AddSubTitlePublished` replicates them using `titling`. See section 65.8. `authblk` is added, and should be loaded before `titling`. See section 65.
- `\multirow` now supports the new optional `vpos` argument.
- Added `\multicolumnrow` for combined `\multicolumn` and `\multirow`. See section 299.2.
- Tabular special cases:
 - Added `\TabularMacro` to mark custom macros inside tabular data cells, avoiding row corruption. See section 9.9.
 - Added `\ResumeTabular` for use when a `tabular` environment is defined inside another environment. See section 9.9.
- Added `supertabular`, `xtab`, `bigstrut`, `bigdelim`.
- Added `fullwidth`.
- Added `addlines`, `any size`, `a4`, `a4wide`, `a5comb`, `textarea`, `zwpagelayout`, `typearea`, `ebook`.

v0.38:[forced single-pass compile](#)

- Added `lwarpmk print1` and `lwarpmk html1` actions to force a compile of the project a single time. Useful when multiple passes are not needed, or changes were not detected.
- Added `\ForceHTMLPage` and `\ForceHTMLTOC` to force a starred sectional unit onto its own HTML page and with its own TOC entry. See section 9.5.2.
- Modified the tutorial to use the new `\ForceHTMLPage` and `\ForceHTMLTOC` macros.
- Added `appendix`, `tocbibind`, `fncychap`, `fix2col`.
- Added `relsize`, `scalefnt`.
- Added `realscripts`, `metalogo`, `xltxtra`.
- Added `grffile`, `romanbar`.
- Added `arabicfront`, `chappg`, `nonumonpart`, `nopageno`, `romanbarpagenumber`.
- Docs: Improved description of the use of front/back matter. See section 9.5.
- Fix: `color` requests `xcolor`.
- Fix: `\part` for `article` class.

[page numbering](#)[front & back matter](#)[packages](#)[font size](#)

v0.37:

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

v0.36:

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

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

v0.34:

- ⚠ Optional arguments**
 - BlockClass's optional argument has been moved in front of the mandatory argument:
`BlockClass[style]{class}` (NEW)
instead of:
`BlockClass{class}[style]` (OLD)
 - This change makes it more consistent with L^AT_EX standards, and avoids problems with space between arguments.
 - Likewise, \InlineClass's optional argument now comes before the mandatory arguments:
`\InlineClass[style]{class}{text}`

- spans with minipages
 - Improved compatibility between spans, minipages, lists, frames, and math. Handles minipages and lists inside an `HTML span`, such as an `\fbox` containing a minipage, although with minimal `HTML` fomatting. See section 9.3.3. `\fboxBlock` is added to frame minipages, tables, and lists with full `HTML` formatting but no longer inline, and behaves as `\fbox` for print output. The `fminipage` environment is added for framed minipages, as an environment with full `HTML` formatting, and draws a framed minipage in print output. See section 9.3.5. `\fbox` and minipages now often work in `SVG` math and `lateXimages`. MATHJAX supports `\fbox`, but not `\fboxBlock` nor `fminipage`.
- framing minipages
 - Improved compatibility between `lateXimage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, `booktabs`.
 - Improved font control for `lateXimagees` and `SVG` math.
 - Added the `eqnarray` environments.
 - `fancyvrb` is no longer required (preloaded), but is still supported.
 - Added `verbatim` and `moreverb`.
 - Added `fancybox`, `boxedminipage2e` and `shadow`.
 - `enumitem` is no longer required, but is still supported.
 - Added `enumerate` and `paralist`.
 - `titleps` is no longer required, but is still supported.
 - Added `crop`.
 - Added `rotfloat`, `marginfit`, and several minor packages; see the change log.
 - Adds fixed-width `HTML` spaces around punctuation when using `babel-french`. `LuATEX` does not yet use the extra punctuation spacing.
- `lateXimage`, `SVG` math,
`tabular`
 - Improved compatibility between `lateXimage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, `booktabs`.
 - Improved font control for `lateXimagees` and `SVG` math.
 - Added the `eqnarray` environments.
 - `fancyvrb` is no longer required (preloaded), but is still supported.
 - Added `verbatim` and `moreverb`.
 - Added `fancybox`, `boxedminipage2e` and `shadow`.
 - `enumitem` is no longer required, but is still supported.
 - Added `enumerate` and `paralist`.
 - `titleps` is no longer required, but is still supported.
 - Added `crop`.
 - Added `rotfloat`, `marginfit`, and several minor packages; see the change log.
 - Adds fixed-width `HTML` spaces around punctuation when using `babel-french`. `LuATEX` does not yet use the extra punctuation spacing.
- `eqnarray`
 - Improved compatibility between `lateXimage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, `booktabs`.
 - Improved font control for `lateXimagees` and `SVG` math.
 - Added the `eqnarray` environments.
 - `fancyvrb` is no longer required (preloaded), but is still supported.
 - Added `verbatim` and `moreverb`.
 - Added `fancybox`, `boxedminipage2e` and `shadow`.
 - `enumitem` is no longer required, but is still supported.
 - Added `enumerate` and `paralist`.
 - `titleps` is no longer required, but is still supported.
 - Added `crop`.
 - Added `rotfloat`, `marginfit`, and several minor packages; see the change log.
 - Adds fixed-width `HTML` spaces around punctuation when using `babel-french`. `LuATEX` does not yet use the extra punctuation spacing.
- verbatim packages
 - Improved compatibility between `lateXimage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, `booktabs`.
 - Improved font control for `lateXimagees` and `SVG` math.
 - Added the `eqnarray` environments.
 - `fancyvrb` is no longer required (preloaded), but is still supported.
 - Added `verbatim` and `moreverb`.
 - Added `fancybox`, `boxedminipage2e` and `shadow`.
 - `enumitem` is no longer required, but is still supported.
 - Added `enumerate` and `paralist`.
 - `titleps` is no longer required, but is still supported.
 - Added `crop`.
 - Added `rotfloat`, `marginfit`, and several minor packages; see the change log.
 - Adds fixed-width `HTML` spaces around punctuation when using `babel-french`. `LuATEX` does not yet use the extra punctuation spacing.
- framing packages
 - Improved compatibility between `lateXimage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, `booktabs`.
 - Improved font control for `lateXimagees` and `SVG` math.
 - Added the `eqnarray` environments.
 - `fancyvrb` is no longer required (preloaded), but is still supported.
 - Added `verbatim` and `moreverb`.
 - Added `fancybox`, `boxedminipage2e` and `shadow`.
 - `enumitem` is no longer required, but is still supported.
 - Added `enumerate` and `paralist`.
 - `titleps` is no longer required, but is still supported.
 - Added `crop`.
 - Added `rotfloat`, `marginfit`, and several minor packages; see the change log.
 - Adds fixed-width `HTML` spaces around punctuation when using `babel-french`. `LuATEX` does not yet use the extra punctuation spacing.
- list packages
 - Improved compatibility between `lateXimage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, `booktabs`.
 - Improved font control for `lateXimagees` and `SVG` math.
 - Added the `eqnarray` environments.
 - `fancyvrb` is no longer required (preloaded), but is still supported.
 - Added `verbatim` and `moreverb`.
 - Added `fancybox`, `boxedminipage2e` and `shadow`.
 - `enumitem` is no longer required, but is still supported.
 - Added `enumerate` and `paralist`.
 - `titleps` is no longer required, but is still supported.
 - Added `crop`.
 - Added `rotfloat`, `marginfit`, and several minor packages; see the change log.
 - Adds fixed-width `HTML` spaces around punctuation when using `babel-french`. `LuATEX` does not yet use the extra punctuation spacing.
- `babel-french`
 - Improved compatibility between `lateXimage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, `booktabs`.
 - Improved font control for `lateXimagees` and `SVG` math.
 - Added the `eqnarray` environments.
 - `fancyvrb` is no longer required (preloaded), but is still supported.
 - Added `verbatim` and `moreverb`.
 - Added `fancybox`, `boxedminipage2e` and `shadow`.
 - `enumitem` is no longer required, but is still supported.
 - Added `enumerate` and `paralist`.
 - `titleps` is no longer required, but is still supported.
 - Added `crop`.
 - Added `rotfloat`, `marginfit`, and several minor packages; see the change log.
 - Adds fixed-width `HTML` spaces around punctuation when using `babel-french`. `LuATEX` does not yet use the extra punctuation spacing.

v0.33:

- Tabular @ and ! columns now have their own `HTML` columns.
- & catcode changes are localized, perhaps causing errors about the tab alignment character &, so any definitions of macros or environments which themselves contain `tabular` and & must be enclosed within `\StartDefiningTabulars` and `\StopDefiningTabulars` (previously called `\EndDefiningTabulars`). See section 43. This change is not required for the routine use of tables, but only when a table is defined inside another macro or environment, and while also using the & character inside the definition. This may include the use inside conditional expressions.
- Several math environments were incorrectly placed inline. Also, for `amsmath` with `SVG` math, the `fleqn` option has been removed, resulting in improved spacing for aligned equations.
- Bug fixes; see the changelog.

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

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

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

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

v0.30:**⚠ lwarf-newproject**

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

To modify existing documents, remove from the document source:

```
\usepackage{lwarf-newproject}
```

The *lwarf* 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
\HomeHTMLFileName	HomeHTMLfilename
\HTMLFileName	HTMLfilename
\useLatexmk	latexmk
\warpOSwindows	OSWindows
Old Package Option	New Package Option
lwarpmklang (new)	xindyLanguage xindyStyle
Old Macro	New Macro
\MetaLanguage	\HTMLLanguage
\HTMLAuthor	\HTMLAuthor
\NewHTMLDescription	\HTMLDescription
\SetFirstPageTop	\HTMLFirstPageTop
\SetPageTop	\HTMLPageTop
\SetPageBottom	\HTMLPageBottom
\NewCSS	\CSSFilename

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

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

- The file `lwarp_html.xdy` has been renamed `lwarp.xdy`. To update each document's project:
 1. Make the changes shown above.
 2. Recompile the document in print mode. This updates the project's configuration files, and also generates the new file `lwarp.xdy`.
 3. The old file `lwarp_html.xdy` may be deleted.
- The new `lwarp` package option `xindyStyle` may be used to tell `lwarpmk` to use a custom `.xdy` file instead of `lwarp.xdy`. See section 9.5.18.
- Improvements in index processing:
 - `xindy`'s language is now used for index processing as well as glossary.
 - Print mode without `latexmk` now uses `xindy` instead of `makeindex`.
 - `texindy/xindy` usage depends on `pdflatex` vs `xelatex`, `lualatex`.
 - For `pdflatex` and `texindy`, the `-C utf8` option is used. This is supported in modern distributions, but a customized `lwarpmk.lua` may need to be created for use with older distributions.

v0.29:

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

v0.28:

- `\HTMLAuthor {<name>}` assigns HTML meta author if non-empty. Defaults to `\theauthor`.
- Boolean `HTMLDebugComments` controls whether HTML comments are added for closing `<div>`s, opening and closing sections, etc.
- Boolean `FormatEPUB` changes HTML output for easy EPUB conversion via an external program. Removes per-file headers, footers, and nav. Adds footnotes per chapter/section.
- Boolean `FormatWordProcessor` changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments. Name changed to `FormatWP` as of v0.42.

- Boolean `HTMLMarkFloats` adds text marks around floats only if the boolean `FormatWordProcessor` is true. These make it easier to identify float boundaries, which are to be manually converted to word-processor frames. Name changed to `WPMarkFloats` as of v0.42.
- Updated for the new MATHJAX CDN repository.
- Adds `tabulary`.
- Supports the options syntax for graphics.
- Improved index references, now pointing exactly to their target.
- Adds glossaries. `lwarfpmk` is modified to add `printglossary` and `htmlglossary` actions.

3 Introduction

The lwarp project aims to allow a rich L^AT_EX document to be converted to a reasonable HTML5 interpretation, with only minor intervention on the user's part. No attempt has been made to force L^AT_EX to provide for every HTML-related possibility, and HTML cannot exactly render every possible L^AT_EX concept. Where compromise is necessary, it is desirable to allow the print output to remain typographically rich, and compromise only in the HTML conversion.

Several “modern” features of HTML5, CSS3, and SVG are employed to allow a fairly feature-rich document without relying on the use of JAVASCRIPT. Limited testing on older browsers shows that these new features degrade gracefully.

lwarp is a native L^AT_EX package, and operates by either patching or emulating various functions. Source-level compatibility is a major goal, but occasional user intervention is required in certain cases.

As a package running directly in L^AT_EX, lwarp has some advantages over other methods of HTML conversion. TEX itself is still used, allowing a wider range of TEX trickery to be understood. Lua expressions are still available with LuaTEX. Entire categories of L^AT_EX packages work as-is when used with lwarp: definitions, file handling, utilities, internal data structures and calculations, specialized math-mode typesetting for various fields of science and engineering, and anything generating plain-text output. Blocks of PDF output may be automatically converted to SVG images while using the same font and spacing as the original print document, directly supporting Tikz and picture. Numerous packages are easily adapted for HTML versions, either by loading and patching the originals, or by creating nullified or emulated replacements, and all without resorting to external programming. As a result, several hundred packages have already been adapted (table 2), and an uncounted number more work as-is.

Packages have been selected according to several criteria: perceived importance, popularity lists, recent CTAN updates, CTAN topics, mention in other packages, support by other HTML conversion methods, and from sample documents taken from public archives. These include some “obsolete” packages as well.¹

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

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

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

SVG images may be used for math, and are also used for *picture*, *Tikz*, and similar environments. The SVG format has better browser and e-book support than MathML (as of this writing), while still allowing for high-quality display and printing of images (again, subject to potentially bug-ridden² browser support).

Furthermore, SVG images allow math to be presented with the same precise formatting as in the print version. Math is accompanied by *<alt>* tags holding the LATEX source for the expression, allowing it to be copy/pasted into other documents.³ Custom LATEX macros may be used as-is in math expressions, since the math is evaluated entirely inside LATEX. An MD5 hash is used to combine multiple instances of the same inline math expression into a single image file, which then needs to be converted to SVG only a single time.

The MATHJAX JavaScript display engine may be selected for math display instead of using SVG images. Subject to browser support and Internet access, MATHJAX allows an HTML page to display math without relying on a large number of external image files.⁴ lwarf maintains LATEX control for cross-referencing and equation numbering, and attempts to force MATHJAX to tag equations accordingly.

A *texlua* program called *lwarpmk* is used to process either the print or HTML version of the document. A few external utility programs are used to finish the conversion from a LATEX-generated PDF file which happens to have HTML5 tags, to a number of HTML5 plain-text files and accompanying images.

lwarf 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, lwarf first looks to see if it has its own modified version to use instead of the standard LATEX version. These *lwarf-pagename.sty* files contain code used to emulate or replace functions for HTML output.

²FIREFOX has had an on-again/off-again bug for quite some time regarding printing svgs at high resolution.

³There seems to be some debate as to whether MathML is actually an improvement over 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!

⁴One SVG image file per math expression, except that duplicate inline math expressions are combined into a single file according to the MD5 hash function of its contents. A common scientific paper can easily include several thousand files, and in one case the MD5 hash cut the number of files in half and the rendering time by 30%.

3.1 Typesetting conventions

Font weight, family, and style are used to indicate various objects:

Table 1: Typesetting conventions

<i>package</i>	L ^A T _E X package.
<i>program</i>	Program's executable name.
<i>option</i>	Program or package option.
<i>filename</i>	File name in the operating system.
BRAND NAME	Proper name for a program, operating system, etc.
<i>commands</i>	Commands to be entered by the user.
<i>code</i>	Program code.
\macro <i>name</i>	L ^A T _E X macro.
<i>environment</i>	L ^A T _E X environment.
<i>counter</i>	L ^A T _E X counter.
<i>boolean</i>	L ^A T _E X boolean.
<element>	HTML element.
<i>attribute</i>	HTML attribute.
User Interface	A user-interface item.
ACRO	Acronym.

3.2 Supported packages and features

Table 2 lists some of the various L^AT_EX features and packages which may be used. Many are tested to work as-is, some are patches for the original packages, and some are emulations written for source-level compatibility. Many are nullified as being irrelevant to HTML output.

Table 2: L^AT_EX–HTML generation — lwarf package — Supported features

Category	Status and supported features.
Engines:	DVI L ^A T _E X, pdfL ^A T _E X, XE ^A T _E X, LuaL ^A T _E X, upL ^A T _E X
Compiling:	latexmk, perltex, pythontex, <i>make</i> , etc.
Classes:	article, book, report, scrartcl, scrbook, scrreprt, memoir, CJK-related as listed below.
Koma-script:	scrextend, scrhack, scrlayer. Others as listed below.
Memoir:	memhfixc
Languages:	babel, polyglossia. cjkpunct, xeCJK.
Chinese:	CTEX, ctex, upzhkinsoku, xpinyin, zhlineskip, zhspacing.
Japanese:	upL ^A T _E X, LuaT _E X-ja, gentombow, lltext, plarray, plarydshln, plautopatch, plect, plectarray, plectarydshln, plectcolortbl, plectdelarray, pxatbegshi, pxeveryshi, pxftnright, pxgentombow, pxjahyper, pxpdfpages, pxpgfrcs, pxpgfmark, tascmac, zxjatype. bxjsarticle and related, ltjsarticle and related, luatexja, luatexja-fontspec, ujarticle and related, utarticle and related.
Korean:	kotex, luatexko, xetexko.

lwarf Supported Functions — continued

Category	Status
Page layout:	2in1, 2up, a4, a4wide, a5comb, addlines, anysize, atbegshi, blowup, booklet, bophook, bounddvi, bxpapersize, canoniclayout, changepage, chngpage, clrdblpg, continue, draftcopy, draftfigure, draftwatermark, ebook, everyshi, fancyhdr, fancytabs, flippdf, fullminipage, fullpage, fwlw, geometry, gmeometric, grid, grid-system, gridset, layaureo, leading, ltxgrid, nccfancyhdr, notespages, pagegrid, pdfprivacy, pagesel, preview, rmpage, scrlayer-scrpage, scrpage2, textarea, threadcol, thumb, thumbs, titleps, toccenter, turnthepage, twoup, typearea, vmargin, watermark, zwpagelayout. Tested to work as-is: underlin.
Sectioning:	Adds <code>FileDepth</code> for splitting the <code>HTML</code> output. Files may be numbered sequentially or named according to section name. Common short words and punctuation are removed from the filenames. <code>anonchap</code> , <code>bsheaders</code> , <code>fncychap</code> , <code>quotchap</code> , <code>section</code> , <code>sectionbreak</code> , <code>sectsty</code> , <code>titlesec</code> . Tested to work as-is: secdot.
Table of contents, figures, tables:	Supported, with hyperlinks. <code>multitoc</code> , <code>shorttoc</code> , <code>titletoc</code> , <code>tocbasic</code> , <code>tocbibind</code> , <code>tocloft</code> , <code>tocstyle</code> .
Title page:	<code>\maketitle</code> , <code>titlepage</code> , <code>authblk</code> , <code>titling</code> .
Front & back matter:	<code>abstract</code> , <code>appendix</code> .
Indexing:	<code>makeindex</code> and <code>xindy</code> are supported, with hyperlinks. <code>idxlayout</code> , <code>imakeidx</code> , <code>index</code> , <code>makeidx</code> , <code>repeatindex</code> , <code>splitidx</code> . Tested to work as-is: <code>hvindex</code> , <code>sepfootnotes</code> .
Glossary:	<code>glossaries</code> and <code>xindy</code> are used.
Bibliography:	<code>babelbib</code> , <code>backref</code> , <code>bilatex</code> , <code>bibunits</code> , <code>chapterbib</code> , <code>cite</code> , <code>hypernat</code> , <code>natbib</code> . Tested to work as-is: <code>notes2bib</code> .
Cross-references:	<code>bookmark</code> , <code>breakurl</code> , <code>cleveref</code> , <code>fancyref</code> , <code>hypdestopt</code> , <code>hyperref</code> , <code>prettyref</code> , <code>titleref</code> , <code>url</code> , <code>varioref</code> , <code>xurl</code> .

lwarf Supported Functions — continued

Category	Status
Margin notes:	marginal, marginfit, marginfix, scrlayer-notecolumn, versonotes.
Footnotes:	Adds FootnoteDepth to print footnotes at section breaks. dblfnote, endheads, endnotes, fnbreak, fnpos, footmisc, footnote, footnoterange, footnpag, marginnote, nccfoots, pagernote, parnotes, pfnote, sidenote. Tested to work as-is: fixfoot.
Math:	Converted to SVG images with <code>HTML <alt></code> tags containing the <code>LATEX</code> source for the math expression. MATHJAX supported as an alternative. <code>amsmath</code> : \mathcal{AMS} environments are supported. User-defined macros are available during conver- son, due to native <code>LATEX</code> processing.
Theorems:	Native <code>LATEX</code> theorems, <code>amsthm</code> , <code>ntheorem</code> , <code>theorem</code> .
Additional math:	Math fonts via svg images, <code>breqn</code> , <code>cases</code> , <code>resizegather</code> , <code>xy</code> . Tested to work as-is: <code>amscd</code> , <code>bm</code> , <code>braket</code> , <code>delarray</code> , <code>guass</code> , <code>nicematrix</code> , <code>pb-diagram</code> , <code>tikz-cd</code> , etc.
Display math with <code>\displaymathother</code> :	Complicated math objects in display math, such as <code>tikz-cd</code> , etc.
Units and fractions:	<code>nicefrac</code> , <code>Slunits</code> , <code>siunitx</code> , <code>units</code> , <code>unitsdef</code> , <code>xfrac</code> . Tested to work as-is: <code>Slunits</code> .
Floats:	Appear where declared. <code>capt-of</code> , <code>caption</code> , <code>cutwin</code> , <code>dblfloatfix</code> , <code>endfloat</code> , <code>fix2col</code> , <code>flafter</code> , <code>float</code> , <code>floatflt</code> , <code>floatrow</code> , <code>fltrace</code> , <code>hypcap</code> , <code>keyfloat</code> , <code>morefloats</code> , <code>newfloat</code> , <code>nonfloat</code> , <code>placeins</code> , <code>rotfloat</code> , <code>stfloats</code> , <code>subcaption</code> , <code>subfig</code> , <code>subfigure</code> , <code>subfloat</code> , <code>trivfloat</code> , <code>wrapfig</code> .
Tabular:	<code>tabular</code> environment, <code>array</code> , <code>arydshln</code> , <code>bigdelim</code> , <code>booktabs</code> , <code>colortbl</code> , <code>diagbox</code> , <code>longtable</code> , <code>ltablex</code> , <code>ltxtable</code> , <code>multirow</code> , <code>supertabular</code> , <code>tabularx</code> , <code>tabulary</code> , <code>threeparttable</code> , <code>xltabular</code> , <code>xtab</code> .

lwarf Supported Functions — continued

Category	Status
Graphics:	graphics and graphicx. \includegraphics supports width, height, origin, angle, and scale tags, and adds class. References to PDF files are changed to SVG, other image types are accepted as well. \rotatebox and \scalebox are supported as well as HTML can handle. rotating is emulated but all objects are unrotated. picture, tikz, and xy are converted to an SVG image. asymptote, epsfig, epstopdf, figsize, grffile, overpic, psfrag, psfrags, pst-eps, pstool, pstricks, rviewport. Tested to work as-is: eepic, tikz-3dplot.
xcolor:	Full package color names, any color models, and mixing. \textcolor, \colorbox, \fcolorbox. Enhanced for HTML compatibility.
Lists:	Standard L ^A T _E X environments, enumerate, enumitem, hang, paralist.
Environments:	Standard L ^A T _E X environments.
minipage, \parbox:	Some HTML5-imposed limitations. Nested minipages are supported. pbox.
Quotations:	copyrightbox, csquotes, epigraph, quoting, verse.
Verbatim:	fancyvrb, moreverb, shortvrb, verbatim.
Frames:	boxedminipage2e, fancybox, framed, mdframed, shadow, verbbars.
Multi-columns:	adjmulticol, multicol, multicolrule, vwcol.
Margins:	fullwidth, hanging, midpage.
Line numbering:	fnlineno, lineno.
Direct formatting:	\emph, \textsuperscript, \textbf, etc are supported. \bfseries, etc. are only supported in some cases. cancel, ellipsis, embrac, enparen, hyphenat, lettrine, lips, lua-check-hyphen, luacolor, magaz, nowidow, pdfrender, realscripts, resize, scalefont, soul, soulpos, soulutf8, textfit, thinspace, trimclip, ulem, underscore, uspace, widows-and-orphans, xellipsis.

lwarf Supported Functions — continued

Category	Status
Acronyms:	acro, acronym.
Ordinals:	engord, fmtcount, nth.
Text ligatures:	Ligatures for symbols are supported. Ligatures for f, q, t are intentionally turned off because many simpler browsers do not display them correctly. Modern full-featured browsers re-create these ligatures on-the-fly.
Horizontal space:	HTML output for thin-unbreakable, unbreakable, \enskip, \quad, \qquad, \hspace.
Rules:	\rule with width, height, raise, text color.
HTML reserved characters:	\&, \textless, and \textgreater are converted to HTML entities.
Fonts:	Used as-is. Appear in svg math expressions or embedded image environments.
Symbols:	Native L ^A T _E X diacriticals, chemgreek, textalpha, textcomp, textgreek, xunicode. Tested to work as-is: gensymb.
Files:	attachfile, attachfile2, hyperxmp, intopdf, pdfpages, pdfx, xmpincl.
Science and engineering:	algorithm2e, algorithmicx, axodraw2, bitpattern, bytefield, chemfig, chemformula, chemgreek, chemmacros, chemnum, listings, mhchem, phfqt, register. Tested to work as-is: blochsphere, bodegraph, bohr, circuitikz, elements, engtlc, fast-diagram, hepnicenames, heppennames, linop, pgfgantt, physics, simpler-wick, slashed, structmech, tikzcodeblocks.
Arts and humanities:	forest, musicography, octave, schemata, semantic-markup, vowel, xpiano. Tested to work as-is: phonrule, piano, tikz-dependency.

l warp Supported Functions — continued

Category	Status
Editorial:	changebar, easy-todo, errata, fixme, fixmetodonotes, pdfcomment, pdfmarginpar, todo, todonotes, tram, xexchangebar.
Accessibility:	accsupp, axessibility.
Debug:	chkfloat, cmdtrack, dprogress, lua-visual-debug, refcheck, srcltx, srctex, vpe, xbmks.
Working as-is:	Various utility, calculation, file, and text-only packages, such as calc, fileerr, somedefs, trace, xspace. Also, most math-only packages, including specialized typesetting for various fields of science and engineering.

4 Alternatives

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

4.1 internet class

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

4.2 TEX4HT

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

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

4.3 Translators

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

Prog **Hevea** **H^Ev^Fa:** <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^ATeXML:** <http://dlmf.nist.gov/LaTeXML/>

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

Prog **LaTeX2HTML** **L^ATeX2HTML:** <http://www.latex2html.org/>
and <http://ctan.org/pkg/latex2html>.

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

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

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

4.4 ASCIIDOC and ASCIIDOCTOR

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

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

4.4.1 ASCIIDOCTOR-L^AT_EX

The Asciidoctor-LaTeX project is developing 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>

4.5 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.)

4.6 Word processors

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

4.7 Commercial systems

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

Prog Flare

- Prog Madcap

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

advanced features where needed, along with a large number of pre-existing packages which provide solutions to numerous common tasks.

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

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

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

5 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.

Detailed installation instructions follow.

Table 3: Required software programs

Provided by your L^AT_EX distribution:

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

L^AT_EX: *pdflatex*, *xelatex*, or *lualatex*.

The lwarf package: This package.

The *lwarpmk* utility: Provided along with this package. This should be an operating-system executable in the same way that *pdflatex* or *latexmk* is. It is possible to have the lwarf 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* program is used by lwarf to create indexes. On a MiK^TE_X system this may have to be acquired separately, but it is part of the regular installer as of mid 2015.

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

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

POPLER 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, and MiK^TE_X provides *miktex-poppler-bin-** packages.

From POPLER: poppler.freedesktop.org.

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

```
Enter ⇒ brew install poppler
```

For WINDOWS, see Mik^TE_X *miktex-poppler-bin-**, or:

<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 may be required for some of the POPLER PDF utilities.

strawberrypperl.com (recommended), perl.org

Automatically downloaded from the internet as required:

MATHJAX: Optionally used to display math. From: mathjax.org

5.1 Installing the l warp package

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

Pre-installed: Try entering into a command line:

Enter ⇒ kpsewhich l warp.sty

If a path to l warp.sty is shown, then l warp is already installed and you may skip to the next section.

TEX Live: If using a T E X Live distribution, try installing via *t lmgr*:

Enter ⇒ t lmgr install l warp

MiK T E X:

1. For newer versions of MiK T E X, install or update l warp using the *MiK T E X Console* program.
2. For older versions of MiK T E X, to install l warp the first time, use the *MiK T E X Package Manager (Admin)*. To update l warp, use *MiK T E X Update (Admin)*.
3. Either way, also update the package *mik tex-misc*, which will install and update the *l warpmk* executable.

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

CTAN TDS archive: l warp may be downloaded from the Comprehensive T E X Archive:

1. See <http://ctan.org/pkg/l warp> for the l warp package.
2. Download the TDS archive: l warp.tds.zip
3. Find the T E X local directory:

TEX Live:

Enter ⇒ kpsewhich -var-value TEXMFLOCAL

MiK T E X:

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

This should be something like:

/usr/local/texlive/texmf-local/

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

Enter ⇒ mktexlsr

— or —

Enter ⇒ texhash

Or, for WINDOWS 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 .dtx and .ins source files. These files are used to create the documentation and .sty files.

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

Enter ⇒ pdflatex l warp.ins

6. Create the documentation:

```
pdflatex l warp.dtx (several times)
makeindex -s gglo.ist -o l warp.gls l warp.glo
makeindex -s gind.ist l warp.idx
pdflatex l warp.dtx (several times)
```

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/l warp

8. Copy l warp_baseline_marker.png and l warp_baseline_marker.eps to the same place as the .sty files.

9. Copy the documentation l warp.pdf to a source directory in the local tree, such as:

<texlocal>/doc/local/l warp

10. Renew the cache:

Enter ⇒ mktexlsr

— or —

Enter ⇒ texhash

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

11. See section 5.2.1 to generate your local copy of *l warpmk*.

12. Once the local version of l warpmk.lua is installed, it may be made available system-wide as per section 5.2.

Project-local CTAN .dtx and .ins files: The .dtx and .ins files may be downloaded to a project directory, then compiled right there, alongside the document source files. The resultant *.sty and l warpmk.lua files may be used as-is, so long as they are in the same directory as the document source.

Just testing!

The files `lwarp_baseline_marker.png` and `lwarp_baseline_marker.eps` must also be copied as well. This approach is especially useful if you would like to temporarily test `lwarp` before deciding whether to permanently install it.

5.2 Installing the `lwarpmk` utility

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

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

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

Otherwise, continue with the following:

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

Enter ⇒ `kpsewhich lwarpmk.lua`

(If the file is not found, you may also generate a local copy and use it instead. See section 5.2.1.)

4. Create `lwarpmk`:

Unix: Create a symbolic link and make it executable:

- (a) Locate the `TEX Live` binaries:

Enter ⇒ `kpsewhich -var-value TEXMFROOT`

This will be something like:

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

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

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

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

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

Enter ⇒ `ln -s <path to lwarpmk.lua> lwarpmk`

- (c) Make the link executable:

Enter ⇒ `chmod 0755 lwarpmk`

WINDOWS `TEX Live`: Create a new `lwarpmk.exe` file:

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

WINDOWS MiKTEX: Create a new `lwarpmk.bat` file:

- (a) Locate the MiKTEX 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\l warp\l warp.texlua" %*`
 This will call the copy of `lwarpmk.lua` which is in the `scripts` directory of the distribution.

5.2.1 Using a local copy of `lwarpmk`

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

1. When compiling the tutorial in section 6, use the `lwarpmk` option for the `l warp` package:
`\usepackage[lwarpmk]{l warp}`
2. When the tutorial is compiled with `pdflatex`, the file `lwarpmk.lua` will be generated along with the other configuration files.
3. `lwarpmk.lua` may be used for this project:

Unix:

- (a) Make `lwarpmk.lua` executable:
`Enter ⇒ chmod 0755 lwarpmk.lua`
- (b) Compile documents with
`Enter ⇒ ./lwarpmk.lua html`
`Enter ⇒ ./lwarpmk.lua print`
`etc.`
- (c) It may be useful to rename or link to a version without the `.lua` suffix.

WINDOWS:

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

```
Enter ⇒ texlua lwarpmk.lua html
Enter ⇒ texlua lwarpmk.lua print
etc.
```

Or:

```
Enter ⇒ lwarpmk html
Enter ⇒ lwarpmk print
etc.
```

5.3 Installing additional utilities

To test for the existence of the additional utilities:

Enter the following in a command line. If each programs' version is displayed, then that utility is already installed. See table 3 on page 89.

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

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

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

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

Prog pdftotext **To install the POPPLER utilities to a UNIX/LINUX system:**

Prog pdfseparate
Prog pdftocairo 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:

If using *MikTeX*, install a *miktex-poppler-bin-** package. Otherwise:

1. See table 3 on page 89.

2. Download and extract the POPPLER utilities *pdftotext*, *pdfseparate*, and *pdfseparate* to a directory, such as Poppler.
3. In the **Start** window, type "Path" to search for results related to Path. Or, open the control panel and search for "Path".
4. Choose **Edit the system environment variables** in the control panel.
5. Choose the **Environment Variables** button.
6. Choose the **Path** variable, then the **Edit** button.
7. Choose the **New** button to make an additional entry.
8. Enter the bin directory of the POPPLER utilities, such as:
C:\Users\<myname>\Desktop\Poppler\poppler-0.5_x86\poppler-0.5\bin
Be sure to include \bin.
9. Click **Ok** when done.

Prog perl **To install PERL to a WINDOWS machine:**

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

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

6 Tutorial

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

See the [General Index](#) for “how-to”, and the [Troubleshooting Index](#) if something doesn’t work. The [Index of Objects](#) contains automated entries for each package, macro, environment, counter, boolean, and other objects; individually and also sorted by category. A [Troubleshooting](#) section is also available.

6.1 Starting a new project

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

Copy from the documentation PDF:

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

Copy from the lwarp documentation directory:

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

Enter ⇒ `texdoc -l lwarp_tutorial.txt`

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

⚠ Bad formatting!

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

3. Compile the project:

Enter ⇒ `pdflatex tutorial.tex`

(several times)

(*xelatex* or *lualatex* may be used as well. `lwarp` also supports DVI *latex* for use with `.eps` images.)

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 engine (*latex*, *pdflatex*, *xelatex*, or *lualatex*), the filenames of the source code and HTML output, and whether the additional helper program *latexmk* will be used to compile the document.)

Figure 1: tutorial.tex listing

Note: There are two pages!

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

\documentclass{book}

\usepackage{iftex}

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

\ifPDFTeX
\usepackage{lmodern}           % pdflatex or dvi latex
\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.
]{lwarf}
% \boolfalse{FileSectionNames} % If false, numbers the files.

% --- LOAD PDFLATEX MATH FONTS HERE ---

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

% --- LATEX AND HTML CUSTOMIZATION ---
\title{The Lwarf Tutorial}
\author{Some Author}
\setcounter{tocdepth}{2}        % Include subsections in the \TOC.
\setcounter{secnumdepth}{2}      % Number down to subsections.
\setcounter{FileDepth}{1}        % Split \HTML\ files at sections
\booltrue{CombineHigherDepths} % Combine parts/chapters/sections
\setcounter{SideTOCDepth}{1}      % Include subsections in the side\TOC
\HTMLTitle{Webpage Title}       % Overrides \title for the web page.
\HTMLAuthor{Some Author}        % Sets the HTML meta author tag.
```

```

\HTMLLanguage{en-US} % Sets the HTML meta language.
\HTMLDescription{A description.}% Sets the HTML meta description.
\HTMLFirstPageTop{Name and \fbox{HOMEPAGE LOGO}}
\HTMLPageTop{\fbox{LOGO}}
\HTMLPageBottom{Contact Information and Copyright}
\CSSFilename{l warp_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}

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

\end{document}

```

Table 4: Files created along with the print version

tutorial.pdf: The PDF output from LATEX. The print version of the document.

tutorial_html.tex: A small .tex file used to create a parallel HTML version of the document, which co-exists with usual the PDF version, and which will have its own auxiliary files. In this way, both PDF and HTML documents may co-exist side-by-side.

Auxiliary files: The usual LATEX files .aux, .log, .out, .toc, .lof, .idx. When an HTML version of the document is created, _html versions of the auxiliary files will also be generated.

lwarpmk.conf: A configuration file for *lwarpmk*, which is used to automate the compilation of PDF or HTML versions of the document.

tutorial.lwarpmkconf: Another configuration file used by *lwarpmk*, which is only useful if you wish to have several projects residing in the same directory.

.css files: lwarf.css, lwarf_formal.css, lwarf_sagebrush.css These files are standard for lwarf, 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 lwarf defaults.

lwarf.ist: Used by lwarf while creating an index using *makeindex*. This file should not be modified by the user. A custom file may be used instead, if necessary.

lwarf.xdy: Used by lwarf while creating an index using *xindy*. This file should not be modified by the user. A custom file may be used instead, if necessary.

lwarf_one_limage.txt: For WINDOWS only. Used to process SVG images in the background. Copied to lwarf_one_limage.cmd when images are generated.

lwarf_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 lwarf to conditionally process blocks of text. This file may be ignored.

When the lwarpmk option is given to the lwarf package:

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

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

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

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

6.2 Compiling the print version with *lwarpmk*

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

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

1. Re-compile the print version:

Enter ⇒ `lwarpmk print`

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

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

Enter ⇒ `lwarpmk print`

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

4. Force a recompile to occur.

Enter ⇒ `lwarpmk again`

Enter ⇒ `lwarpmk print`

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

5. Process the index.⁶ ⁷

Enter ⇒ `lwarpmk printindex`

6. Recompile again to include the index.

Enter ⇒ `lwarpmk print`

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

Enter ⇒ `lwarpmk print1`

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

⁵Although, when using the utility *latexpmk* (introduced later), the changed date is ignored and an actual change in contents must occur to cause a recompile.

⁶The command *lwarpmk printglossary* is also available to process a glossary produced with the *glossaries* package. See section 9.5.10.

⁷Also see section 9.5.12 for index options.

6.3 Compiling the HTML version with *lwarpmk*

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

1. Compile the HTML version:

Enter ⇒ `lwarpmk html`

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

2. View the HTML page in a web browser.

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

math

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

3. Force a recompile:

Enter ⇒ `lwarpmk again`

Enter ⇒ `lwarpmk html`

Enter ⇒ `lwarpmk print`

4. Process the HTML index and recompile: ¹⁰¹¹

Enter ⇒ `lwarpmk htmlindex`

Enter ⇒ `lwarpmk html`

`_Index.html` is updated for the new L^AT_EX index.

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

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

Enter ⇒ `lwarpmk html1`

⁸`First-chapter.html` also contains the first section, even though the second section is its own HTML page. This behavior is controlled by the boolean `CombineHigherDepths`.

⁹`index.html` is commonly used as a homepage, so the document index is in `_Index.html`.

¹⁰The command `lwarpmk htmlglossary` is also available to process a glossary produced with the `glossaries` package. See section 9.5.10.

¹¹Also see section 9.5.12 for index options.

6.4 Generating the svg images

math as svg images

By default lwarf represents math as SVG images with the L^AT_EX source included in `alt` attributes. In this way, the math is displayed as it was drawn by L^AT_EX, and the L^AT_EX source may be copied and pasted into other documents.

picture and Tikz

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

1. Create the SVG images:

```
Enter ⇒ lwarpmk limages
Enter ⇒ lwarpmk html
```

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

⚠️ adding/removing

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

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

⚠️ HTML instead of images

If `HTML` appears where an SVG image should be, recompile the document one more time to get the page numbers back in sync, then remake the images one more time. `lwarpmk` attempts to detect this situation and print a warning.

⚠️ page counter

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

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

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

Expressing math as SVG images has the advantage of representing the math exactly as L^AT_EX would, but has the disadvantage of requiring an individual file for each math expression. For inline math, and some other objects, lwarf uses an MD5 hash on its L^AT_EX source to combine multiple instances of identical inline expressions into a single image file, but display math and other environments such as `picture` and

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

Tikz require one image file each. For a document with a large amount of math, see section [6.5](#) to use MATHJAX instead.

6.5 Using MATHJAX for math

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

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

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

2. Recompile

```
Enter ⇒ lwarfpmk html
```

3. Reload the math page.

MATHJAX requirements

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

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

6.6 Changing the css style

For a formal css style, add to the preamble:

```
\usepackage{lwarf}
...
\CSSFilename{lwarf_formal.css}
...
\begin{document}
```

For a modern css style, lwarf_sagebrush.css is also provided:

```
\CSSFilename{lwarf_sagebrush.css}
```

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

6.7 Customizing the HTML output

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

See section 8.4 for more information.

6.8 Using *latexmk*

latexmk is a L^AT_EX 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`

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

3. Recompile the document.

Enter ⇒ `lwarpmk print`

and/or

Enter ⇒ `lwarpmk html`

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

Enter ⇒ `lwarpmk print1`

and/or

Enter ⇒ `lwarpmk html1`

[forced single-pass recompile](#)

6.9 Using X_ELATEX or LuaLATEX

X_ELATEX or LuaLATEX may be used instead of LATEX.

1. Remove the auxiliary files for the project:

Enter ⇒ lwarfpmk cleanall

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

Enter ⇒ xelatex tutorial.tex

-or-

Enter ⇒ lualatex tutorial.tex

When the compile occurs, the configuration files for *lwarfpmk* are modified to remember which TEX engine was used. X_ELATEX or LuaLATEX will be used for future runs of *lwarfpmk*.

3. To recompile the document:

Enter ⇒ lwarfpmk print

-and-

Enter ⇒ lwarfpmk html

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

Enter ⇒ lwarfpmk htmlindex

Enter ⇒ lwarfpmk html

Enter ⇒ lwarfpmk printindex

Enter ⇒ lwarfpmk print

6.10 Using DVI LATEX

Traditional DVI LaTeX may also be used along with .eps image files. An SVG version of each image must also be provided. *lwarfpmk* may be used to convert image formats.

To convert EPS files to PDF:

Enter ⇒ lwarfpmk epstopdf *.eps (or a list of files)

To convert PDF files to SVG:

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

⚠️ **bitmapped fonts** See section 8.2 regarding font selection to avoid the use of bitmapped fonts.

6.11 Using a glossary

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

To process the glossary for the print version:

Enter ⇒ `lwarpmk printglossary`

 (If `makeglossaries` is not found, see section 9.5.10.)

To process the glossary for the HTML version:

Enter ⇒ `lwarpmk htmlglossary`

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

Enter ⇒ `lwarpmk html1`

Enter ⇒ `lwarpmk html`

Enter ⇒ `lwarpmk print1`

Enter ⇒ `lwarpmk print`

See section 9.5.10 to set options for processing glossaries.

6.12 Cleaning auxiliary files

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

Enter ⇒ lwarfpmk clean

6.13 Cleaning auxiliary and output files

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

Enter ⇒ lwarfpmk cleanall

6.14 Cleaning the images from the lateximages directory

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

Enter ⇒ lwarfpmk cleanlimages

6.15 Converting PDF or EPS images to SVG

HTML cannot display PDF or EPS images, so any external PDF graphics images must be converted to SVG format. *pdftocairo* and *epstopdf* may be used one image at a time, but *lwarfpmk* also provides a way to convert PDF or EPS images in bulk:

Enter ⇒ lwarfpmk epstopdf *.eps (or a list of files)

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

Be sure to always provide SVG files for HTML output.

6.16 Creating HTML from an incomplete compile

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

Enter ⇒ lwarfpmk pdftohtml [-p project]

6.17 Processing multiple projects in the same directory

It is possible to have several projects in the same directory. *lwarfmk* 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 -p project_a
```

```
Enter ⇒ lwarpmk html -p project_b
```

6.18 Using the *make* utility

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

```
lwarpmk pdftohtml [-p project]
```

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

7 Converting an existing document

To convert an existing document for use with l warp:

1. Arrange the document in the following order:
 - (a) Declare the \documentclass.
 - (b) Load text fonts.
 - (c) Load inputenc or inputenx, fontenc, and/or fontspec.
 - (d) Load l warp.
 - (e) Load remaining packages.
2. Modify the document:
 - (a) Avoid the scale option. Change:

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

to:

```
\includegraphics[width=<yy>\linewidth]
```
 - (b) Possible changes to tabular environments include * columns, multirow, longtable, supertabular, xtab, bigdelim. See section 9.9.
 - (c) Possible option clashes with memoir. See section 9.12.
 - (d) If using indexes, see section 9.5.12.
 - (e) If using many indexes, glossaries, . aux files, etc., see section 9.5.12 regarding morewrites. If morewrites is already used, be sure to add the setup with allocate=10.
 - (f) Other changes as per [Special cases and limitations](#), section 9.
3. Convert any PDF images to SVG. See section 9.7.
4. Manually compile the print version with *latex*, *pdflatex*, *lualatex*, or *xelatex*.
5. l warpmk print to finish the print version.
6. l warpmk html to create the HTML version.
7. l warpmk limages to create the SVG images of any svg math, lateximage, Tikz, etc.

Need help?

See the [General Index](#) for “how-to”, and the [Troubleshooting Index](#) if something doesn’t work. The [Index of Objects](#) contains automated entries for each package, macro, environment, counter, boolean, and other objects; individually and also sorted by category. A [Troubleshooting](#) section is also available.

8 Additional details

8.1 Shell escape

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

8.2 Font and UTF-8 support

 **type 3 bitmapped fonts** lwarf uses *pdftotext* to convert PDF output into UTF-8-encoded text. This process requires that UTF-8 information be embedded in the PDF file, which may prevent the use of older “type 3” bit-mapped fonts, and of older packages such as ae. The lwarf option `pdftotextEnc` may be useful in some situations. See section 8.3.

vector fonts While using DVI *latex* or PDF *pdflatex*, if no font-related package is specified then the default Computer Modern font is used, which may be a “type 3” bit-mapped font which may not convert well to plain text. A “type 1” vector font is required.

Computer Modern

 `pdflatex`

 `DVI latex`

Pkg `cm-super`

Pkg `lmodern`

To use the updated cm-super’s type 1 fonts instead of Computer Modern, install the cm-super font package.

To use Latin Modern instead, add

```
usepackage{lmodern}
```

to the preamble.

Pkg `dejavu` Another useful option is the Deja Vu series of fonts, which have an increased coverage of language and glyphs:

```
\usepackage{dejavu}
```

latex, *pdflatex*, T1, UTF8

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

xelatex and *lualatex*

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

Place `fontspec` or `fontenc`, `xunicode`, and other font and UTF-8 related commands after the `\documentclass` command and before `\usepackage{lwarf}`.

⚠ package conflicts

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

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

- For X E T E X or L u a L T E X :

Pkg `fontspec`

ligatures

l warp sets the following to turn off T E X 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 p d f l a t e x :

Pkg `lmodern`

Pkg `fontenc`

Pkg `inputenc`

Pkg `inputenx`

Pkg `newunicodechar`

File `glyptounicode.tex`

Pkg `cmap`

Pkg `textcomp`

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

3. `\usepackage{l warp}` (section 8.3) goes after any of the above, followed by:
4. `\usepackage{newtxmath}` or other math-related font packages. Many of these load amsmath, which must be loaded after l warp, so they must also be loaded after l warp.
5. `\setmonofont{TeX Gyre Cursor}` or similar may be required if using X E T E X or L u a L T E X and fontspec along with traditional font packages such as txfonts, newtxtext, etc. This is required to turn off the monospaced font's ligatures with fontspec after loading the traditional font packages. Monospaced output ligatures must be turned off to produce the correct HTML characters.
6. ... the rest of the preamble and the main document.

⚠ fontspec with traditional font packages

⚠ UTF-8 locale

In some cases, an external program may require a UTF-8 “locale”. See section 10.8.

8.2.1 Indexes, glossaries, and encoding

lwarf uses the *xindy* program to processes indexes. *xelatex* and *lualatex* use *xindy* and *pdflatex* uses *texindy*.

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

8.3 lwarf package loading and options

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

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

Package options may be set while loading lwarf, or later with

```
\lwarfsetup{\{key=value, ...}\}}
```

Pkg lwarf

Opt `mathsvg` `mathjax`: For math display, select `mathsvg` (default) or `mathjax`. For more information about the math options, see section 9.6.

Opt `latexmk` `latexmk`: Tells *lwarfmk* to use *latexmk* to recompile the document several times if necessary. Otherwise, *lwarfmk* attempts to determine for itself whether to recompile. See section 8.4.

Opt `dvips` `dvips`: Tells *lwarfmk* to use *dvips* and *ps2pdf* to convert DVI output to PDF.

`Default: false` `dvipdfm`: Tells *lwarfmk* to use *dvipdfm* to convert DVI output to PDF.

`Default: false` `dvipdfmx`: Tells *lwarfmk* to use *dvipdfmx* to convert DVI output to PDF.

Opt `dvipdfmx` `HomeHTMLFilename`: See section 8.4.

Opt `HomeHTMLFilename` `Default: {}`

Opt `HTMLFilename` `Default: {}`

Opt `PrintLatexCmd`

`Default: <automatic>`

Opt `HTMLLatexCmd` `Default: <automatic>`

`PrintLatexCmd`: Sets the shell commands executed by *lwarfmk print*. If not specified, will automatically be set according to the detected L^AT_EX engine and the use of `--shell-escape`.

`HTMLLatexCmd`: Sets the shell commands executed by *lwarfmk html*. If not specified, will automatically be set according to the detected L^AT_EX engine and the use of `--shell-escape`.

Opt `makeindex` `makeindex`: Sets `PrintIndexCmd`, `HTMLIndexCmd`, and `LatexmkImageCmd` to use `makeindex` when generating indexes with *lwarfmk printindex*, *lwarfmk htmlindex*, or *latexmk*. If neither `makeindex` nor `xindy` is used, `makeindex` is assumed.

Opt `xindy` `xindy`: Sets `PrintIndexCmd`, `HTMLIndexCmd`, and `LatexmkImageCmd` to use `xindy` when generating indexes with *lwarfmk printindex*, *lwarfmk htmlindex*, or *latexmk*.

Opt `makeindexStyle` `makeindexStyle`: If you wish to use a custom `.ist` file for index generation, see section 28.

Opt `xindyStyle` `xindyStyle`: If you wish to use a custom `.xdy` file for index generation, see section 28.

Table 5: l warp package options

Option	Description
<code>mathsvg</code>	Show math using SVG images.
<code>mathjax</code>	Show math using MATHJAX.
<code>latexmk</code>	Use <i>latexmk</i> for compiling documents.
<code>dvips</code>	Use <i>dvips</i> and <i>ps2pdf</i> to convert DVI documents.
<code>dvipdfm</code>	Use <i>dvipdfm</i> to convert DVI documents.
<code>dvipdfmx</code>	Use <i>dvipdfmx</i> to convert DVI documents.
<code>HomeHTMLFilename</code>	The filename of the home page.
<code>HTMLFilename</code>	A prefix for the filenames of the remaining web pages.
<code>PrintLatexCmd</code>	The shell commands for <i>l warpmk print</i> .
<code>HTMLLatexCmd</code>	The shell commands for <i>l warpmk html</i> .
For indexing (section 9.5.12) and glossaries (section 9.5.10):	
<code>makeindex</code>	Use <i>makeindex</i> to generate indices.
<code>xindy</code>	Use <i>xindy</i> to generate indices.
<code>makeindexStyle</code>	Set a custom style for <i>makeindex</i> .
<code>xindyStyle</code>	Set a custom style for <i>xindy</i> .
<code>xindyLanguage</code>	The <i>xindy</i> language option used for index generation.
<code>xindyCodepage</code>	The <i>xindy</i> codepage option used for index generation.
<code>PrintIndexCmd</code>	Shell commands executed by <i>l warpmk printindex</i> .
<code>HTMLIndexCmd</code>	Shell commands executed by <i>l warpmk htmlindex</i> .
<code>LatexmkIndexCmd</code>	Shell commands executed by <i>latexmk</i> .
<code>GlossaryCmd</code>	Shell command executed by <i>l warpmk printglossary</i> and <i>l warpmk htmlglossary</i> .
Seldom necessary:	
<code>OSWindows</code>	Force compatibility with MS-WINDOWS.
<code>pdftotextEnc</code>	Set the encoding for <i>pdftotext</i> .
<code>l warpmk</code>	Generate a local copy of <i>l warpmk.lua</i> .
Used internally by l warp:	
<code>warpprint</code>	Generate print output, and also generate configuration files.
<code>warpHTML</code>	Generate HTML output.
<code>BaseJobname</code>	The \jobname to use. Set to the \jobname of the printed version even while generating HTML.

	Opt <code>xindyLanguage</code> Default: <code>english</code>	xindyLanguage: If using an index or glossary, see section 28.
	Opt <code>xindyCodepage</code> Default: <code>utf8</code>	xindyCodepage: If using an index, see section 28.
	Opt <code>PrintIndexCmd</code> Default: <code><automatic></code>	PrintIndexCmd: Sets the shell commands executed by <code>lwarpmk printindex</code> . If not specified, will be set by the selection of <code>makeindex</code> or <code>xindy</code> . May be used to specify the creation of multiple indexes. See section 9.5.12.
		Examples: <pre>makeindex -s lwarf.ist projectname.idx (makeindex) xindy -M lwarf.xdy -L english -C utf8 projectname.idx (xindy)</pre>
automatic setting		The use of the <code>makeindex</code> or <code>xindy</code> options sets <code>PrintIndexCmd</code> to sensible values for each of those programs while compiling a single index. <code>lwarf</code> 's <code>makeindexStyle</code> , <code>xindyStyle</code> , <code>xindyLanguage</code> , and <code>xindyCodepage</code> options will be used if specified.
 xindy		If specifying <code>PrintIndexCmd</code> manually, be sure to assign an <code>xindy</code> language and codepage with the <code>-L</code> and <code>-C</code> <code>xindy</code> options, as the <code>lwarf</code> <code>xindyLanguage</code> and <code>xindyCodepage</code> options are not used for the <code>PrintIndexCmd</code> option when it is set manually.
		This option is stored in the configuration files <code>lwarpmk.conf</code> and <code>*.lwarpmkconf</code> , and is then passed by the <code>lwarpmk printindex</code> command to the operating system to compile the print indexes. Since the command string is parsed by <code>TEX</code> , written to a file, read from the file by <code>LuaTEX</code> , and finally passed to the operating system, any attempt at quoting will be problematic. For complicated commands, it would be best to create a shell script, and simply refer to the script with the <code>lwarf PrintIndexCmd</code> option.
Opt <code>HTMLIndexCmd</code> Default: <code><automatic></code>	HTMLIndexCmd: Sets the shell commands executed by <code>lwarpmk htmlindex</code> . If not specified, will be set by the selection of <code>makeindex</code> or <code>xindy</code> . May be used to specify the creation of multiple indexes. See section 9.5.12.	
 filenames		Example settings are similar to <code>PrintIndexCmd</code> , but append <code>_html</code> to the filenames: <pre>makeindex -s lwarf.ist projectname_html.idx (makeindex) xindy -M lwarf.xdy -L english -C utf8 projectname_html.idx (xindy)</pre>
automatic setting		The use of the <code>makeindex</code> or <code>xindy</code> options sets <code>HTMLIndexCmd</code> to sensible values for each of those programs while compiling a single index. <code>lwarf</code> 's <code>makeindexStyle</code> , <code>xindyStyle</code> , <code>xindyLanguage</code> , and <code>xindyCodepage</code> options will be used if specified.
 xindy		If specifying <code>HTMLIndexCmd</code> manually, be sure to assign an <code>xindy</code> language and codepage with the <code>-L</code> and <code>-C</code> <code>xindy</code> options, as the <code>lwarf</code> <code>xindyLanguage</code> and <code>xindyCodepage</code> options are not used for the <code>HTMLIndexCmd</code> option when it is set manually.
		As with <code>PrintIndexCmd</code> , to generate complicated indexes it may be worthwhile to use a shell script, then refer to that script with <code>HTMLIndexCmd</code> .

Opt <code>LatexmkIndexCmd</code>	LatexmkIndexCmd: Sets the shell commands executed by <i>latexmk</i> . Unlike <code>PrintIndexCmd</code> and <code>HTMLIndexCmd</code> , <code>LatexmkIndexCmd</code> does not include any filenames, which will be provided instead by <i>latexmk</i> . See section 9.5.12.
Default: <code><automatic></code>	Example settings are similar to <code>PrintIndexCmd</code> , but without a filename: <code>makeindex -s lwarf.ist</code> <i>(makeindex)</i> <code>xindy -M lwarf.xdy -L english -C utf8</code> <i>(xindy)</i>
automatic setting	The use of the <code>makeindex</code> or <code>xindy</code> options sets <code>LatexmkIndexCmd</code> to either of the two settings shown above. <i>lwarf</i> 's <code>makeindexStyle</code> , <code>xindyStyle</code> , <code>xindyLanguage</code> , and <code>xindyCodepage</code> options will be used if specified. Unlike <code>PrintIndexCmd</code> and <code>HTMLIndexCmd</code> , <i>latexmk</i> uses either of the single-line settings of <code>LatexmkIndexCmd</code> shown above to compile each of multiple indexes if necessary.
 <code>xindy</code>	If specifying <code>LatexmkIndexCmd</code> manually, be sure to assign an <code>xindy</code> language and codepage with the <code>-L</code> and <code>-C</code> <code>xindy</code> options, as the <i>lwarf</i> <code>xindyLanguage</code> and <code>xindyCodepage</code> options are not used for the <code>LatexmkIndexCmd</code> option when it is set manually.
Opt <code>GlossaryCmd</code>	GlossaryCmd: Sets the shell command executed by <i>lwarpmk printglossary</i> and <i>lwarpmk htmlglossary</i> . The print or HTML glossary filename is appended to this command. See section 9.5.10.
Default: <code>makeglossaries</code>	
Opt <code>OSWindows</code>	OSWindows: <i>lwarf</i> attempts to automatically sense WINDOWS, but it may be forced with this option. See section 8.6.
Opt <code>pdftotextEnc</code>	pdftotextEnc: Used to specify the encoding used by <i>pdftotext</i> during the PDF-HTML conversion. In most situations, the default is the correct choice.
Default: <code>UTF-8</code>	
Opt <code>lwarpmk</code>	lwarpmk: If you wish to have <i>lwarf</i> generate a local copy of <code>lwarpmk.lua</code> for archival or local-installation purposes, compile the print version with the <code>lwarpmk</code> option set. See section 28.
<hr/>	
The following options are used internally by <i>lwarf</i> , and usually are not used in the user's document:	
Opt <code>warpprint</code>	warpprint and warpHTML: Usually controlled by <i>lwarpmk</i> , and not set in the document. Select the <code>warpprint</code> option to generate print output (default), or the <code>warpHTML</code> option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual <i>pdflatex</i> , etc. When <i>lwarf</i> is loaded in print mode, it creates <code><project>.html.tex</code> , which sets the <code>warpHTML</code> option before calling the user's source code <code><project>.tex</code> . In this way, <code><project>.tex</code> can <code>\usepackage{lwarf}</code> without any options to create a printed version, while <code><project>.html.tex</code> will create an HTML version.
Opt <code>warpprint</code>	
Opt <code>warpHTML</code>	
Opt <code>BaseJobname</code>	BaseJobname: Not intended for the user. Used internally by <i>lwarf</i> when creating the <code>*_html.tex</code> file used to compile the HTML version. See section 28.
Default: <code>\jobname</code>	

8.4 Customizing the HTML output

⚠ Placement! Table 6 shows 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 ⇒ `lwarfpmk clearall`

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

Enter ⇒ `lwarfpmk print`

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

Enter ⇒ `lwarfpmk html`

Options for the lwarf package:

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

Opt `HomeHTMLFilename`
Default: `\BaseJobname`

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

`HomeHTMLFilename=index`

filename underscores

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

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

See section 8.4.1 for examples of naming and numbering HTML files.

Opt `HTMLFilename`
Default: `<empty>`

HTMLFilename: A filename prefix for the rest of the HTML web pages. Useful for numbered web pages with a common prefix. May be empty. See section 8.4.1 for examples of naming and numbering HTML files.

Opt `latexmk`
Default: `false`

latexmk: Controls whether lwarf uses `latexmk` to compile the document. This setting is written to `lwarfpmk`'s configuration files.

Opt `mathsvg`
Default: `true`

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

Opt `mathjax`
Default: `false`

mathjax: Selects MATHJAX for math output.

Opt `makeindex`
Default: `makeindex`

makeindex: Selects `makeindex` for index generation by `lwarfpmk`.

Opt `xindy`
Default: `makeindex`

xindy: Selects `xindy` for index generation by `lwarfpmk`.

Table 6: HTML settings

Macro / Cntr / Bool	Description
\linkhomename	Name of the link to the homepage.
SideTOCDepth	Sectioning depth of the sidetoc.
\sidetocname	Name of the sidetoc.
FileDepth	Sectioning depth of the file splits.
CombineHigherDepths	Combine higher section levels.
FileSectionNames	Use section names for file names, else use numbers.
FootnoteDepth	Sectioning depth of footnotes.
\abstractname	The name of the abstract.
\mathimagename	The SVG math image <code>latexitimage alt</code> tag.
\packagediagramname	The suffix for a package's <code>latexitimage alt</code> tags.
\CSSFilename	The css for the following files.
\HTMLLanguage	The HTML lang tag.
\HTMLTitle	The homepage's <title>, overriding \title.
\HTMLTitleBeforeSection	Set subpage <title>s to \HTMLTitle - sectionname
\HTMLTitleAfterSection	Set subpage <title>s to sectionname - \HTMLTitle
\HTMLAuthor	The HTML author meta tag, overriding \author.
\HTMLDescription	The HTML description meta tag.
\HTMLFirstPageTop	Heading for the home page.
\HTMLPageTop	Heading for the other pages.
\HTMLPageBottom	Footing for all pages.
HTMLDebugComments	Boolean to generate HTML comments.

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

`\linkhomename`
 Default: `Home`
`Ctr tocdepth`

`\linkhomename`: Name of the link to the home page. Paragraphs are allowed.

`tocdepth`: Sectioning depth of the table of contents. See section 17 for a list of L^AT_EX stack depths.

`Ctr SideTOCDepth`
 Default: `1`

sideroc

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

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

It is recommended to set:

`SideTOCDepth = FileDepth`

or

`SideTOCDepth = FileDepth+1`

 **inaccessible pages**

`\sidetocname`
 Default: `Contents`
`Ctr FileDepth`
 Default: `-5`

`\sidetocname`: Name of the sideroc. Paragraphs are allowed.

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

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



`Bool CombineHigherDepths`
 Default: `true`

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

The file splits are controlled by the counter `FileDepth` and the boolean `CombineHigherDepths`. Setting `FileDepth` to 0 splits the file at chapters, 1 at sections, etc. `CombineHigherDepths` controls whether to combine pages at levels higher than the chosen `FileDepth`, such as in this tutorial where the page which opens the chapter also contains the first section.

⚠ Inaccesible pages!

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`.

⚠ Lost in an old page!

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

Bool `FileSectionNames`
Default: `true`

FileSectionNames: If true, web page filenames are derived from a sanitized version of the section names. If false, web pages are numbered. Either way, the `HTMLFilename` option is used as a prefix. See section [8.4.1](#) for examples of naming and numbering HTML files. The user must ensure that filenames are unique after begin sanitized. For example, math in the section name is removed before creating the filename, so the rest of the filename must be sufficiently unique to avoid name collisions.

Ctr `FootnoteDepth`
Default: `3`

FootnoteDepth: Determines where to place pending footnotes. 3 places footnotes before each break down to the `\subsubsection` level. 1 places footnotes before each `\section` break. Any pending footnotes are also placed at the bottom of each page before each file break.

Bool `HTMLDebugComments`
Default: `false`

HTMLDebugComments: Set true to generate HTML comments, such as which section or `<div>` is being opened or closed.

`\abstractname`
Default: `Abstract`

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

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

`\CSSFilename`
Default: `lwarp.css`

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

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

`\HTMLLanguage`
Default: `en-US`

\HTMLLanguage: {`<language>`} The HTML file's HTML lang meta tag. Defaults to `en-US`.

\HTMLTitle
Default: \thetitle

\HTMLTitleBeforeSection
Default:
 \HTMLTitleBeforeSection
\HTMLTitleAfterSection

\HTMLTitle: {*title*} Overrides \title for the HTML header's meta title. Defaults to \thetitle, which is set by \title, or empty otherwise. Unlike the author, \thetitle is set by \title even if not using the titling package.

\HTMLTitleBeforeSection: Sets subpage <title> tags to show the website title followed by the section name.

\HTMLTitleAfterSection: Sets subpage <title> tags to show the section name followed by the website title.

custom <title>

To customize subpage <title>s, redefine \theHTMLTitleSection, which defaults to:

```
\def\theHTMLTitleSection{%
  \theHTMLTitle\theHTMLTitleSeparator\theHTMLSection%
}
```

\HTMLAuthor
Default: \theauthor

\HTMLAuthor: {*author*} The HTML header's meta author. Defaults to \theauthor, which is set by \author if using the titling package, but is empty otherwise. There are several ways to represent the author and affiliations, especially if using the authblk package, most of which do not result in a sensible \theauthor, so \HTMLAuthor is useful to create a list of authors without their affiliations.

\HTMLDescription
Default: <empty>

\HTMLDescription: {*description*} Sets the HTML description tag for the following files. May be changed before each each sectioning command which would cause a file split.

\HTMLFirstPageTop
Default: <empty>

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

\HTMLPageTop
Default: <empty>

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

\HTMLPageBottom
Default: <empty>

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

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

⚠ \tableofcontents
TOC on the homepage!

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

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

Placed in the document wherever necessary:

`\mathimagename`
Default: `math image`

`\mathimagename`: When creating an SVG math image, its `alt` tag may be set to the math expression, which may be hashed for image reuse. In the case of `\ensuremath` or after `\inlinemathother`, where the contents require a unique image for each instance of the same expression, the `alt` tag is set to `\mathimagename`, and the image is not reused.

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

`\packagediagramname`
Default: `diagram`

`\packagediagramname`: For many packages, the output is placed inside a `lateximage` with an HTML `alt` tag set to the package name followed by `\packagediagramname`. For example:

(-xy- diagram)

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

Env `warpprint`

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



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

Env `warpHTML`

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



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

`\warpprionly`

`\warpprionly: {<contents>}` A macro version of the `warpprint` environment.

`\warpHTMLonly`

`\warpHTMLonly: {<contents>}` A macro version of the `warpHTML` environment.

8.4.1 Example HTML file naming

Examples of ways to name or number HTML files:

Numbered HTML nodes:

Example: Homepage index.html, and node-1, node-2.¹³

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

Named HTML sections, no prefix:

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

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

Named HTML sections, with prefix:

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

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

8.5 Customizing the css

\CSSFilename
Default: `lwarp.css`

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

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

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

The styles provided by lwarp include:

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

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

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

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

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

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

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

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

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

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

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

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

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

8.6 Selecting the operating system

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

8.7 Selecting actions for print or HTML output

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

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

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

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

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

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

Env warpprint	Anything which is to be done only for print output is surrounded by a warpprint
---------------	---

environment:

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

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

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

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

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

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

`\warpprintonly {<actions>}`

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

`\warpHTMLonly {<actions>}`

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

8.8 Commands to be placed into the `warpprint` environment

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

- Paragraph formatting: `\parindent` `\parskip`
- Manual page positions such as the `textpos` package, which is emulated but only in a limited way.
- Anything changing the page counter. `lwarf` requires that the page counter not be adjusted during HTML output.

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 14: [Troubleshooting](#).

8.9 Title page

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

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

The title is used in the meta tags in the HTML files, unless overridden by \HTMLTitle, and the rest are used in \maketitle. To use a \subtitle or \published field, see section 65.8.

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

\title {\langle title\rangle}

- ⚠ **newlines** Avoid newlines in the \title; these will interfere with the file break and css detection. Use a \subtitle command instead (section 65.8). The title will appear in the document \maketitle as a heading <h1>. The HTML meta title tag will also have this title, unless \HTMLTitle is used to set the meta title to something else instead.

\author {\langle author\rangle}

In \author, \protect may be needed before some formatting commands. In HTML, the author will appear in a <div> of class author in the \maketitle. If the titling package is used, the author will also appear in a HTML meta tag, but \HTMLAuthor may be necessary to create a plain list of names if \author had affiliations added. \affiliation is a new addition to lwarp.

\date {\langle date\rangle}

\date works as expected. In HTML, this will appear in a <div> class titled date.
\thanks {\langle text \rangle}
\thanks are allowed in the titlepage fields, and will be rendered as HTML notes at the bottom of the title page.

8.10 HTML page meta descriptions

- \HTMLDescription {\langle A description of the web page. \rangle} The default is no description.
- limitations** Each page of HTML output should have its own HTML meta description, which usually shows up in web search results, is limited to around 150 characters in length, and should not include the ASCII double quote character ("').
- placement** Use \HTMLDescription just before \begin{document} to set the description of the home page, and also just before each sectioning command such as \chapter or \section where a new file will be generated, depending on FileDepth. For example, if FileDepth is 1, use \HTMLDescription just before each \section command, and that description will be placed inside the HTML page for that \section. The same description will be used for all following HTML files as well, until reset by a new \HTMLDescription. It is best to use a unique description for each HTML file.
- disabling** To disable the generation of HTML description meta tags, use:

```
\HTMLDescription{}
```

8.11 HTML homepage meta title

- \HTMLTitle {\langle title \rangle} Sets the contents of the web page <meta name="title"> element. Defaults to \HTMLtitle{\thetitle}. May be set empty to cancel the meta title tag.
- See section 8.4 for \HTMLTitleBeforeSection and \HTMLTitleAfterSection, used to set the title for HTML subpages.

8.12 HTML page meta author

- \HTMLAuthor {\langle author \rangle} Sets the contents of the web page <meta name="author"> element. Defaults to \HTMLAuthor{\theauthor}. May be set empty to cancel the meta author tag.
- \author may be used to create a list of authors and their affiliations, in several formats if using authblk, and these may not successfully parse properly into a sensible

list for `\theauthor`. `\HTMLAuthor` may be used to set the meta tag to a simple list of names.

9 Special cases and limitations

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

See the [General Index](#) for “how-to”, and the [Troubleshooting Index](#) if something doesn’t work. The [Index of Objects](#) contains automated entries for each package, macro, environment, counter, boolean, and other objects; individually and also sorted by category. A [Troubleshooting](#) section is also available.

9.1 Things to avoid

In the document, avoid the following:

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

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

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

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

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

9.2 Formatting

9.2.1 Text formatting

- ⚠ \bfseries, etc. \textbf{, etc.} are supported, but \bfseries, etc. work only in some situations.
- ⚠ HTML special chars &, <, and > have special meanings in HTML. If \&, \textless, and \textgreater are used, proper HTML entities will be used, but there may be HTML parsing problems if these special characters occur unescaped in program listings or other verbatim text.
- program listings For program listings, the listings package is supported, and its literate option is used to convert &, <, and > to proper HTML entities.
- verbatim The various verbatim-related environments do not convert &, <, and >, so care must be taken to avoid accidentally including valid HTML code inside these environments. Adding a space on either side may be sufficient.

9.2.2 Horizontal space

- \hspace \hspace is converted to an inline HTML span of the given width, except that 0 width is ignored, a width of .16667em is converted to an HTML thin breakable space (U+2009), and a \fill is converted to a \quad.
- \, \~ and \, are converted to HTML entities.
- \kern \kern and \hskip are entered into the HTML PDF output as-is, then interpreted by \pdftotext, and thus usually appear as a single space.

9.2.3 Text alignment

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

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

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

9.2.4 Accents

Native L^AT_EX accents such as `\"` will work, but many more kinds of accents are available when using Unicode-aware X_EL^AT_EX and LuaL^AT_EX.

9.2.5 textcomp package

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

 **missing symbols** Many textcomp symbols are not supported by many fonts. In the css try referencing fonts which are more complete, but expect to see gaps in coverage.

9.2.6 Superscripts and other non-math uses of math mode

Use `x` instead of `x`

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

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

9.2.8 Filenames and URLs in lists or footnotes

filename underscore Escape underscores in the filenames:

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

9.2.9 relsize package

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

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

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

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

9.3 Boxes and minipages

9.3.1 Marginpars

- \marginpar [⟨left⟩] {⟨right⟩} \marginpar may contain paragraphs, but in order to remain inline with the surrounding text lwarp nullifies block-related macros inside the \marginpar. Paragraph breaks are converted to
 tags.
- \marginparBlock [⟨left⟩] {⟨right⟩} To include block-related macros, use \marginparBlock, which takes the same arguments but creates a <div> instead of a . A line break will occur in the text where the \marginBlock occurs.

9.3.2 Save Boxes

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

9.3.3 Minipages

- ⚠ inline** A line of text with an inline minipage or parbox will have the minipage or parbox placed onto its own line, because a paragraph is a block element and cannot be made inline-block.
- placement** Minipages and parboxes will be placed side-by-side in HTML unless you place a \newline between them.
- side-by-side** Side-by-side minipages may be separated by \quad, \quad, \enskip, \hspace, \hfill, or a \rule. When inside a center environment, the result is similar in print and HTML. Paragraph tags are suppressed between side-by-side minipages and these spacing commands, but not at the start or end of the paragraph.
- in a span** There is limited support for minipages inside an HTML . An HTML <div> cannot appear inside a . While in a , minipages, and parboxes, and any enclosed lists have limited HTML tags, resulting in an “inline” format, without markup except for HTML breaks. Use \newline or \par for an HTML break.
- size** When using \linewidth, \textwidth, and \textheight, widths and heights are scaled proportionally to a 6×9 inch text area.
- no-width minipages** A minipage of width exactly \linewidth is automatically given no HTML width.

full-width minipages

A new macro `\minipagefullwidth` requests that the next minipage be generated without an `HTML width` attribute, allowing it to be the full width of the display rather than the fixed width given.

⚠ text alignment

Nested minipages adopt their parent's text alignment in `HTML`, whereas in regular L^AT_EX PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

9.3.4 Side-by-side minipages

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

9.3.5 Framed minipages and other environments

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

`\fbox` For a framed object, options include:
`\fboxBlock`
`\fboxEnv` `fminipage`

To remove the frame in `HTML` output: Place the `\fbox` command and its closing brace inside `warpprint` environments. This will nullify the frame for `HTML` output.

For inline text: **To frame the contents inline with some formatting losses in `HTML`:** This is the default action of `\fbox` when enclosing a minipage. During `HTML` output, `\fbox` nullifies the `HTML` tags for `minipage`, `\parbox`, and lists. The contents are included as inline text inside the `\fbox`'s `` of class `framebox`. For lists, line breaks are converted to `HTML` breaks. The result is a plain-text inline version of the contents, framed inline with the surrounding text, but lacking any extra `HTML` markup.

For inline `minipage` and lists: **To frame the contents on their own line with improved formatting in `HTML`:** A new command `\fboxBlock` is included, intended to be a direct replacement for `\fbox` for cases where the `\fbox` surrounds a minipage, table, or list. For print output, this behaves as `\fbox`. For `HTML` output, the contents are placed inside an `HTML <div>` with the class `framed`, resulting in the contents being placed on their own line with a frame surrounding them. The contents preserve their `HTML` formatting, so lists and minipages look nicer, and valid `HTML` is created for a `tabular`. While an `\fbox` containing a `tabular` is valid L^AT_EX code, the result in `HTML` is problematic since a table is a `<div>` not a ``, so use

\fboxBlock around a tabular, or else place the tabular inside a minipage, or use fminipage, described next. Also see below regarding the “Misplaced alignment tab character &.” error.

To create a framed minipage in both print and HTML: A new environment fminipage is included. For print output, this is identical to minipage, except that it is also framed. For HTML output, this forms a <div> of class framed, the contents preserve their HTML formatting, and valid HTML is created for a tabular. Also see section 84 for a new environment fcolorminipage. Also see below regarding the “Misplaced alignment tab character &.” error.

colored boxes and frames: **To create colored frames and boxes:** See section 452 for xcolor’s \colorbox and \fcolorbox, and l warp’s additional \colorboxBlock and \fcolorboxBlock.

⚠ Misplaced alignment tab character &

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

To frame equations: See section 201 for the fancybox package.

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

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

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

9.3.6 fancybox package

Pkg fancybox
framed equation example

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

```
\newenvironment{FramedEqn}
{
  \latextimage% NEW
  \setlength{\fboxsep}{15pt}
  ...{...
  \[\fbox{\TheSbox}\]}
  \endlatextimage% NEW
}
```

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

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

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

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

```
\newenvironment{FramedVerb}[1] % width
{
  \VerbatimEnvironment
  \fminipage{#1}
  \begin{Verbatim}
  \end{Verbatim}
  \endfminipage
}
```

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

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

9.3.7 mdframed package

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

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

font For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

where `\textbf` must appear just before the comma and will receive the following text as its argument (since the text happens to be between braces in the mdframed source). Since lwarf 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.

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

css classes Environments created or encapsulated by mdframed are enclosed in a `<div>` of class `mdframed`, and also class `md<environmentname>` for new environments.

Frame titles are placed in a `<div>` of class `|mdframedtitle|`. Subtitles are in a `<div>` of class `|mdframedsubtitle|`, and likewise for subsubtitles.

9.4 Cross-references

labels Labels with special characters may be a problem. It is best to stick with alphanumeric, hyphen, underscore, and perhaps the colon (if not French).

 **label characters** `\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.

9.4.1 Page references

 **LATEX page numbers** The printed page does not translate to the HTML page, so `\pageref` references are

converted to parentheses containing \pageref{PageFor}, which defaults to “see”, followed by a hyperlink to the appropriate object.

Ex:

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

\pageref{PageFor} may be redefined to “page for”, empty, etc. See page 497.

9.4.2 cleveref and varioref packages

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

⚠ cleveref page numbers

Ex:

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

See \cpagerefFor at page 576 to redefine the message which is printed for page number references.

9.4.3 Hyperlinks, hyperref, and url

Pkg hyperref	lwarf emulates hyperref, including the creation of active hyperlinks, but does not require that hyperref be loaded by the document.
Pkg url	

⚠ comments between arguments

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

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

⚠ backref

When generating HTML, lwarf’s emulation of hyperref does not automatically load backref, so backref must be loaded explicitly.

9.4.4 Footnotes and page notes

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

The footmisc stable option is emulated by lwarf.

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

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

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

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

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

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

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

and likewise for equations or display math.

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

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

9.5 Front and back matter

9.5.1 Custom classes with multiple authors and affiliations

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

```
%\documentclass{customclass} % for print document  
\documentclass{article} % for HTML document  
  
\usepackage{lwarp}  
\begin{warpHTML}  
\usepackage{authblk}  
\let\affiliation\affil % maybe required  
\end{warpHTML}
```

9.5.2 Starred chapters and sections

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

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

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

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

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

 **inaccessible HTML page**

- \ForceHTMLTOC To automatically force the HTML version of the document to have a TOC entry for a

starred section, use `\ForceHTMLTOC` just before the `\chapter*` or `\section*`, and place `\phantomsection` and `\addcontentsline` inside a `warpprint` environment.

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

9.5.3 abstract package

Pkg `abstract`

⚠ `missing TOC`

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

9.5.4 titling and authblk

Pkg `titling`

Pkg `authblk`

`package support`

⚠ `load order`

`\published` and `\subtitle`

lwarf supports the native L^AT_EX `titling` commands, and also supports the packages `authblk` and `titling`. If both are used, `authblk` should be loaded before `titling`.

If using the `titling` package, additional titlepage fields for `\published` and `\subtitle` may be added by using `\AddSubtitlePublished` in the preamble. See section 65.8.

9.5.5 tocloft package

Opt `tocloft titles`

Pkg `tocloft`

Pkg `tocloft`

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

⚠ `tocloft & other packages`

9.5.6 appendix package

Pkg `appendix`

⚠ `incorrect TOC link`

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

9.5.7 pagenote package

Pkg `pagenote`

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

9.5.8 endnotes package

Pkg `endnotes` To place the endnotes in the TOC, use:

table of contents

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

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

```
\ForceHTMLPage
\theendnotes
```

9.5.9 BIBTeX

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



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

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

9.5.10 glossaries package

Pkg `glossaries` `lwarpmk` has the commands `lwarpmk printglossary` and `lwarpmk htmlglossary`, which process the glossaries created by the `glossaries` package using that package's `makeglossaries` command.

Opt `GlossaryCmd`

Default: `makeglossaries`

Opt `lwarpmk printglossary`

Opt `lwarpmk htmlglossary`

The shell command to execute is set by the lwarf option `GlossaryCmd`, which defaults to `makeglossaries`. The print or HTML glossary filename is appended to this command.

⚠ `makeglossaries` not found

In some situations it may be required to modify the default command, such as to add the `perl` command in front:

```
\usepackage[
    GlossaryCmd={perl makeglossaries},
] {lwarf}
```

`xindy language` To set the language to use for processing glossaries with `xindy`:

```
\usepackage[
    GlossaryCmd={makeglossaries -L english},
] {lwarp}
```

Other options for `makeglossaries` may be set as well.

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

Numbered section, on its own HTML page:

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

Unnumbered section, inline with the current HTML page:

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

Unnumbered section, on its own HTML page:

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

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

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

print/HTML versions The print and HTML versions of the glossary differ in their internal page numbers. Separate commands for generating print and HTML glossaries are used, even though the page number is currently ignored.

9.5.11 Indexing overview

There are many ways to process indexes for a L^AT_EX document, including native L^AT_EX capabilities, a number of packages and classes, the possible availability of shell escape and *latexmk*, and the need to process print and HTML versions. `lwarp` attempts to provide easy recompilation of indexes along with the rest of the document, but the various indexing options must be set correctly. Numerous examples are given below. Some differ in minor details, so the important parts are highlighted in red, and options are in green.

Once set up properly, the entire document may be recompiled with *lwarpmk print* and *lwarpmk html*. In some cases, it will also be necessary to compile the indexes with *lwarpmk printindex* and *lwarpmk htmlindex*. A recompile may then be forced with *lwarpmk print1* and *lwarpmk html1*.

- manual processing** The user may continue to process indexes manually or by shell script without the use of *lwarpmk*, but adjustments will be required to process HTML indexes as well. In general, *.idx and *.ind files will be accompanied by *_html.idx and *_html.ind files.
- custom index style** If using a custom indexing style file, see sections 9.5.17 and 9.5.18.
- source code** See section 75 for lwarf's core index and glossary code, section 258 for index, section 390 for splitidx, section 256 for imakeidx, section 420 for tocbibind, and section 469.17 for memoir's indexing patches.

9.5.12 Indexing with basic L^AT_EX and makeidx

- lwarpmk* processing** The following allow the user to process indexes automatically, or using *lwarpmk*'s commands:

```
Enter ⇒ lwarpmk printindex  
Enter ⇒ lwarpmk htmlindex
```

For a single index using *makeindex*:

```
\usepackage[makeindex,latexmk]{lwarf}
```

The usual .idx and .ind files will be used, along with the new *lwarf.ist* style file. When creating the HTML index, “_html” is automatically appended to each of the names.

lwarpmk will use *latexmk* if specified, in which case *latexmk* will create the index automatically. Otherwise, use

```
Enter ⇒ lwarpmk printindex  
Enter ⇒ lwarpmk htmlindex
```

to compile the indexes.

For a single index using *xindy*:

```
\usepackage[
    xindy,
    xindyLanguage=english,                                     <optional>
    xindyCodepage=utf8,                                      <optional>
    latexmk
]{lwarp}
```

The usual .idx and .ind files will be used, along with the new lwarp.xdy style file.

lwarpmk will use *latexmk* if specified, in which case *latexmk* will create the index automatically. Otherwise, use

Enter ⇒ lwarpmk printindex

Enter ⇒ lwarpmk htmlindex

to compile the indexes.

9.5.13 Indexing with index

lwarp is told how to use *makeindex* using the PrintIndexCmd and HTMLIndexCmd options. The file lwarp.ist is specified, which generates index letter heads for print output and also allows special HTML formatting for HTML output.

For multiple indexes using *makeindex* and *index*:

(Assuming that the second index has file extensions .sist and .sind)

```
\usepackage[
    makeindex, latexmk,
    PrintIndexCmd={
        makeindex -s lwarp.ist <projectname>.idx ;
        makeindex -s lwarp.ist
            -o <projectname>.sind <projectname>.sidx
    },
    HTMLIndexCmd={
        makeindex -s lwarp.ist <projectname>_html.idx ;
        makeindex -s lwarp.ist
            -o <projectname>_html.sind <projectname>_html.sidx
    }
]{lwarp}
\usepackage{index}
...
\makeindex
\newindex{secondname}{sidx}{sind}{Second Index}
```

 **WINDOWS**

For WINDOWS, replace the two ";" characters with "&".

When creating the HTML index, “_html” is automatically appended to the index filenames.

Use

```
Enter ⇒ l warpmk printindex  
Enter ⇒ l warpmk htmlindex
```

to compile the indexes.

If the `latexmk` option is selected for l warp, `latexmk` will compile the document but will *not* compile the indexes. `l warpmk printindex` and `l warpmk htmlindex` will still be required.

9.5.14 Indexing with splitidx

l warp is told how to use *splitindex* using the `PrintIndexCmd` and `HTMLIndexCmd` options. The file `l warp.ist` is specified, which generates index letter heads for print output and also allows special HTML formatting for HTML output.

If the `latexmk` option is selected for l warp, `latexmk` will compile the document but will *not* compile the indexes. `l warpmk printindex` and `l warpmk htmlindex` will still be required.

- ⚠ `\thepage` When using `\AtWriteToIndex` or `\AtNextWriteToIndex`, the user must not refer to `\thepage` during HTML output, as the concept of a page number is meaningless. Instead, do

```
\addtocounter{LWR@autoindex}{1}  
\LWR@new@label{LWRindex-\arabic{LWR@autoindex}}
```

where the `\index`-like action occurs, and then refer to `\arabic{LWR@autoindex}` instead of `\thepage` where the reference should occur.

See section 469.17 in the l warp-patch-memoir package for the `\@wrspindexhyp` macro as an example.

For multiple indexes using *makeindex* and *splitidx*:

```
\usepackage[
    makeindex, latexmk,
    PrintIndexCmd={
        splitindex <projectname> -- -s lwarf.ist
    },
    HTMLIndexCmd={
        splitindex <projectname>_html -- -s lwarf.ist
    }
]{lwarf}
\usepackage{splitidx}
...
\makeindex
\newindex[Second Index]{secondname}
```

When creating the HTML index, “_html” is automatically appended to each of the names.

Use

```
Enter ⇒ lwarpmk printindex
Enter ⇒ lwarpmk htmlindex
```

to compile the indexes.

For multiple indexes using *xindy* and *splitidx*:

```
\usepackage[
    xindy, latexmk,
    PrintIndexCmd={
        splitindex -m xindy <projectname> -- -M lwarf.xdy
        -L english -C utf8                                <optional>
    },
    HTMLIndexCmd={
        splitindex -m xindy <projectname>_html -- -M lwarf.xdy
        -L english -C utf8                                <optional>
    }
]{lwarf}
\usepackage{splitidx}
...
\makeindex
\newindex[Second Index]{secondname}
```

When creating the HTML index, “_html” is automatically appended to each of the names.

Use

```
Enter ⇒ lwarpmk printindex
Enter ⇒ lwarpmk htmlindex
```

to compile the indexes.

9.5.15 Indexing with `imakeidx`

Due to the number of methods which may be used to process multiple indexes, the options for style file and *xindy* language and codepage must be specified in one of several different ways. These are described in detail later in this section, but are summarized here.

If shell escape is used, `imakeidx` will automatically compile the indexes by itself. Options specifying a custom style file and *xindy* language and codepage must be specified for each `\makeindex` command using its `options=` option, which must include l warp's special `l warp.ist` or `l warp.xdy` file, or a file based on them. If using a custom indexing style file, see sections 9.5.17 and 9.5.18. The `splitindex` option is also available if shell escape is used, in which case the `splitidx` package and `splitindex` program will also be used.

If shell escape is not possible, `latexmk` may be used to automatically compile the indexes. The style, language, and codepage options are specified with l warp's `makeindexStyle`, `xindyStyle`, `xindyLanguage`, and `xindyCodepage` options. These are passed to `latexmk` by `l warpmk`'s `l warpmk printindex` and `l warpmk htmlindex` commands.

Where shell escape and `latexmk` are not possible, `l warpmk` may be used to manually compile the indexes. l warp's `PrintIndexCmd` and `HTMLIndexCmd` options are used.

For a single or multiple indexes using `makeindex` and `imakeidx`:

The index style `l warp.ist` is automatically used for HTML output. This file turns on letter headings, so it may be desirable to specify it as an option, in which case it will also be used for print output, which will help match the print and HTML output.

```
\usepackage[makeindex,latexmk]{l warp}
\usepackage[makeindex]{imakeidx}
...
\makeindex[options={-s l warp.ist}]
\makeindex[name=secondname,options={-s l warp.ist}]
```

`imakeidx` will automatically compile the indexes. Shell escape is not required while using `makeindex`. `latexmk` may be specified, and if so it will be used for `l warpmk print` and `l warpmk html`, but `imakeidx` will actually create the indexes.

For a single or multiple indexes using `makeindex` and `splitindex` with `imakeidx`:

The index style `lwarf.ist` is automatically used for HTML output. This file turns on letter headings, so it may be desirable to specify it as an option, in which case it will also be used for print output, which will help match the print and HTML output.

```
\usepackage[makeindex,latexmk]{lwarf}
\usepackage[makeindex,splitindex]{imakeidx}
...
\makeindex[options={-s lwarf.ist}]
\makeindex[name=secondname,options={-s lwarf.ist}]
```

enable shell escape

Shell escape is required while using `splitindex`. For the first compile, use

Enter ⇒ `pdflatex --shell-escape projectname.tex`

Enter ⇒ `pdflatex --enable-write18 projectname.tex` (MikTeX)

or similar with `xelatex` or `lualatex`. `lwarf` will remember that shell escape was used.

`imakeidx` will automatically execute `splitindex`, and will also use `makeindex` to compile the indexes.

`latexmk` may be specified, and if so it will be used for `lwarfmk print` and `lwarfmk html`, but `imakeidx` will actually create the indexes.

For multiple indexes using `xindy` and `imakeidx`, using shell escape:

Options may be given to `imakeidx`'s `\makeindex` command. The style file `lwarf.xdy` is automatically used for HTML output, and is not necessary for print output since the output will be similar. If language or codepage must be set, they should be specified as options for `\makeindex`, since `imakeidx` will process the indexes.

```
\usepackage[xindy,latexmk]{lwarf}
\usepackage[xindy,splitindex]{imakeidx}
...
\makeindex[
  options={ -M lwarf.xdy -L english -c utf8 }
]
\makeindex[
  name=secondname,
  options={ -M lwarf.xdy -L english -c utf8 }
]
```

enable shell escape

For the first compile, use

Enter ⇒ `pdflatex --shell-escape projectname.tex`

Enter ⇒ `pdflatex --enable-write18 projectname.tex` (MikTeX)

or similar with `xelatex` or `lualatex`. `lwarf` will remember that shell escape was used.

imakeidx will automatically execute *splitindex* if selected, and will also use *xindy* to compile the indexes.

If selected, *latexmk* will automatically recompile the entire document as necessary.

For indexes using *xindy* and *imakeidx*, without shell escape, but with *latexmk*:

lwarf's options are used, and are passed to *latexmk*.

```
\usepackage[
    xindy,
    xindyLanguage=english,                                <optional>
    xindyCodepage=utf8,                                 <optional>
    latexmk,
]{lwarf}
\usepackage[xindy]{imakeidx}
...
\makeindex
\makeindex[name=secondname]
```

latexmk will create the indexes automatically when *lwarpmk print* and *lwarpmk html* are executed.

For indexes using *xindy* and *imakeidx*, without shell escape, and without *latexmk*:

lwarpmk must be told how to create the indexes:

```
\usepackage[
    xindy,
    PrintIndexCmd={
        xindy -M lwarf.xdy -L english -C utf8
            <projectname>.idx ;
        xindy -M lwarf.xdy -L english -C utf8
            secondname.idx
    },
    HTMLIndexCmd={
        xindy -M lwarf.xdy -L english -C utf8
            <projectname>_html.idx ;
        xindy -M lwarf.xdy -L english -C utf8
            secondname_html.idx
    }
]{lwarf}
\usepackage[xindy]{imakeidx}
...
\makeindex
\makeindex[name=secondname]
```

⚠ WINDOWS

For WINDOWS, replace the two ";" characters with "&".

<projectname> is the \jobname: if compiling "name.tex", use the filenames name.idx and name_html.idx.

Use

Enter ⇒ lwarpmk printindex

Enter ⇒ lwarpmk htmlindex

to compile the indexes.

9.5.16 Indexes with memoir

For a single index with memoir and makeindex:

```
\documentclass{memoir}
\usepackage[makeindex,latexmk]{lwarp}
...
\makeindex
```

The usual .idx and .ind files will be used, along with the lwarp.ist style file. *lwarpmk* will use *latexmk* if specified, in which case *latexmk* will create the index automatically. Otherwise, use

Enter ⇒ lwarpmk printindex

Enter ⇒ lwarpmk htmlindex

to compile the indexes.

For multiple indexes with memoir and makeindex, using latexmk:

lwarp's options are used, and are passed to *latexmk*.

```
\documentclass{memoir}
\usepackage[makeindex,latexmk]{lwarp}
...
\makeindex
\makeindex[secondname]
```

lwarpmk will use *latexmk* to create the indexes automatically when the user executes *lwarpmk print* and *lwarpmk html*.

For multiple indexes with memoir and makeindex, *without* latexmk:

lwarfmk must be told how to create the indexes:

```
\documentclass{memoir}
\usepackage[
    makeindex,
    PrintIndexCmd={
        makeindex -s lwarf.ist <projectname>.idx ;
        makeindex -s lwarf.ist secondname.idx
    },
    HTMLIndexCmd={
        makeindex -s lwarf.ist <projectname>_html.idx ;
        makeindex -s lwarf.ist secondname_html.idx
    }
]{lwarf}
...
\makeindex
\makeindex[secondname]
```

 **WINDOWS**

For WINDOWS, replace the two “;” characters with “&”.

<projectname> is the \jobname: if compiling “name.tex”, use the filenames name.idx and name_html.idx.

Use

```
Enter ⇒ lwarfmk printindex
Enter ⇒ lwarfmk htmlindex
```

to compile the indexes.

For a single index with memoir and xindy:

```
\documentclass{memoir}
\usepackage[
    xindy,
    xindyLanguage=english,                                <optional>
    xindyCodepage=utf8,                                 <optional>
    latexmk
]{lwarf}
...
\xindyindex
\makeindex
```

The usual .idx and .ind files will be used, along with the lwarf.xdy style file.

lwarfmk will use *latexmk* if specified, in which case *latexmk* will create the index automatically. Otherwise, use

```
Enter ⇒ lwarfmk printindex
Enter ⇒ lwarfmk htmlindex
```

to compile the indexes.

For multiple indexes with memoir and xindy, using latexmk:

l warp's options are used, and are passed to *latexmk*.

```
\documentclass{memoir}
\usepackage[
    xindy,
    xindyLanguage=english,
    xindyCodepage=utf8,                                <optional>
    latexmk
]{l warp}
...
\xindyindex
\makeindex
\makeindex[secondname]
```

l warpmk will use *latexmk* to create the indexes automatically.

For multiple indexes with memoir and xindy, without latexmk:

l warpmk must be told how to create the indexes:

```
\documentclass{memoir}
\usepackage[
    xindy,
    PrintIndexCmd={
        xindy -M l warp.xdy -L english -C utf8
        <projectname>.idx ;
        xindy -M l warp.xdy -L english -C utf8
        secondname.idx
    },
    HTMLIndexCmd={
        xindy -M l warp.xdy -L english -C utf8
        <projectname>_html.idx ;
        xindy -M l warp.xdy -L english -C utf8
        secondname_html.idx
    }
]{l warp}
...
\xindyindex
\makeindex
\makeindex[secondname]
```

 **WINDOWS**

For WINDOWS, replace the four ";" characters with "&".

<projectname> is the \jobname: if compiling "name.tex", use the filenames name.idx and name_html.idx.

Use

Enter ⇒ l warpmk printindex

Enter ⇒ `lwarpmk htmlindex`
 to compile the indexes.

9.5.17 Using a custom *makeindex* style file

Prog `makeindex` When using *makeindex*, *lwarpmk* uses the file `lwarf.ist` to process the index. This file is over-written by *lwarf* whenever a print version of the document is processed.

To use a custom *makeindex* style file:

1. Copy `lwarf.ist` to a new filename such as `projectname.ist`
2. Make changes to `projectname.ist`. Keep the lines which refer to `\hyperindexref`. These lines creates the hyperlinks for the HTML index. During print output `\hyperindexref` becomes a null function.

Opt `makeindexStyle` 3. In the document source use the `makeindexStyle` option for *lwarf*:

```
\usepackage[
    ... other options ...
    \textred{makeindexStyle=projectname.ist},
]{lwarf}
```

Likewise, refer to the custom style file if using `\PrintIndexCmd`, `\HTMLIndexCmd`, or `\LatexmkIndexCmd`.

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

9.5.18 Using a custom *xindy* style file

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

To use a custom *xindy* style file:

1. Copy `lwarf.xdy` to a new filename such as `projectname.xdy`
2. Make changes to `projectname.xdy`.
 Keep the lines which refer to `\hyperindexref`:

```
(define-attributes ((\"hyperindexref\")))
(markup-locref :open "\hyperindexref{" :close "}")
...
(markup-locref :open "\textit{\hyperindexref{" :close "}}" :attr "textit")
```

These lines create the hyperlinks for the HTML index. During print output `\hyperindexref` becomes a null function.

To create custom styles, refer to the lines for `\textbf` and `\textit`.

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

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

Likewise, refer to the custom style file if using `\PrintIndexCmd`, `\HTMLIndexCmd`, or `\LatexmkIndexCmd`.

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

9.5.19 Additional indexing limitations

 **xindy with hyperref** *xindy* and *hyperref* may not work well together for print output with “see”, “see also”, reference ranges, or stylized index references. It may be necessary to turn off hyper-referencing for indexes:

```
\usepackage[hyperindex=false]{hyperref}
```

 **empty index** If an HTML index is empty, it may be necessary to add the following before `lwarp` is loaded:

```
\usepackage{morewrites}
\morewritessetup{allocate=10}
...
\usepackage{lwarp}
```

makeindex custom display styles When using *makeindex*, custom display styles are possible:

```
\begin{warpprint}
\newcommand{\notesstyle}[1]{#1nn}
\end{warpprint}

\begin{warpHTML}
\makeatletter
\newcommand{\notesstyle}[1]{\LWR@doindexentry{#1} notes }
\makeatother
\end{warpHTML}
...
A sentence.\index{key|notesstyle}
```

xindy custom display styles For custom styles with *xindy*, see `lwarf.xdy` for `\textbf` and `\textit` as examples.

9.5.20 Index positions, `toc`, `tocbibind`

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

Pkg `makeidx` **Inline, with a manual toc entry:**

A commonly-used method to introduce an index in a L^AT_EX document:

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

Pkg `makeidx` **On its own HTML page, with a manual toc entry:**

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

Pkg `tocbibind` **Inline, with an automatic toc entry:**

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

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

Pkg `tocbibind` **On its own HTML page, with an automatic toc entry:**

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

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

Other packages, such as `imakeidx`, may also have options for including the index in the Table of Contents.

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

9.6 Math

9.6.1 Rendering tradeoffs

Math rendering Math may be rendered as SVG graphics or using the MATHJAX JavaScript display engine.

SVG files Rendering math as images creates a new SVG file for each expression, except that an MD5 hash is used to combine identical duplicates of the same inline math expression into a single file, which must be converted to SVG only once. Display math is still handled as individual files, since it may contain labels or references which are likely to change.

SVG inline The SVG images are currently stored separately, but they could be encoded in-line directly into the HTML document. This may reduce the number of files and potentially speed loading the images, but slows the display of the rest of the document before the images are loaded.

PNG files Others L^AT_EX-to-HTML converters have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but SVG files are the preferred approach for scalable graphics.

MathML Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 11 regarding EPUB output with MATHJAX.

9.6.2 svg option

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

¹⁴See section 456 regarding fonts and fractions.

SVG image font size

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

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

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

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

SVG math copy/paste

For SVG math, text copy/paste from the HTML `<alt>` tags lists the equation number or tag for single equations, along with the L^AT_EX code for the math expression. For *AMS* environments with multiple numbers in the same environment, only the first and last is copy/pasted, as a range. No tags are listed inside a starred *AMS* environment, although the `\tag` macro will still appear inside the L^AT_EX math expression.

⚠ SVG math in TeX boxes

SVG math does not work inside TeX boxes, since a `\newpage` is required before and after each image.

9.6.3 MATHJAX option**MATHJAX math option**

Prog MathJax

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

When MATHJAX is enabled, math is rendered twice:

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

9.6.4 Customizing MATHJAX

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

Examples:

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

9.6.5 MATHJAX limitations

MATHJAX limitations

Prog MathJax

subequations

- MATHJAX itself does not support subequations. This may be improved by parsing the L^AT_EX math expression to manually insert tags, but this has not yet been done.

footnotes in math

latextimage

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

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

siunitx

⚠ siunitx inside an equation

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

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

Also see section 9.6.10.

tabbing

⚠ other macros and packages

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

- Other math-related macros and packages are not supported by MATHJAX, including \ensuremath and \bigdelim, along with occasionally-used macros such as \relax. lwarp emulates footnotes, units, and nicefrac for MathJax.

9.6.6 Catcode changes

preamble macros with math

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

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

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

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

9.6.7 Complicated inline math objects

<code>\inlinemathnormal</code>	An inline math expression is usually converted to a reusable hashed SVG math image, or a MathJax expression. The hash or expression depends on the contents of the math expression. In most cases this math expression is static, such as <code>\$x+1\$</code> , so the image can be reused for multiple instances of the same expression. In some cases, the math expression includes a counter or other object which may change between uses. Another problem is complicated contents which do not expand well in an <code>alt</code> tag. The macro <code>\inlinemathother</code> may be used before a dynamic math expression, and <code>\inlinemathnormal</code> after. Doing so tells lwarf to use an unhashed SVG math image, even if MathJax is in use. See section 44.
changing contents complicated alt tag	

9.6.8 Complicated display math objects

<code>\displaymathnormal</code>	By default, or when selecting <code>\displaymathnormal</code> , MATHJAX math display environments print their contents as text into HTML, and SVG display math environments render their contents as SVG images and use their contents as the <code>alt</code> tag of HTML output. To do so, the contents are loaded into a macro for reuse. In some cases, such as complicated Tikz pictures, compilation will fail.
MathJax unsupported complicated alt tag	When selecting <code>\displaymathother</code> , it is assumed that the contents are more complicated than “pure” math. An example is an elaborate Tikz picture, which will not render in MATHJAX and will not make sense as an HTML <code>alt</code> tag. In this mode, MATHJAX is turned off, math display environments become SVG images, even if MATHJAX is selected, and the HTML <code>alt</code> tags become simple messages. The contents are internally processed as an environment instead of a macro argument, so complicated objects such as Tikz pictures are more likely to compile successfully.

9.6.9 ntheorem package

Pkg <code>ntheorem</code>	This conversion is not total. Font control is via <code>css</code> , and the custom LATEX font settings are ignored.
⚠️ <code>Font control</code>	
⚠️ <code>Equation numbering</code>	<code>ntheorem</code> has a bug with equation numbering in <code>\AMS</code> environments when the option <code>thref</code> is used. lwarf does not share this bug, so equations with <code>\split</code> , etc, are

numbered correctly with lwarp's HTML output, but not with the print output. It is recommended to use cleveref instead of ntheorem's thref option.

9.6.10 siunitx package

Pkg siunitx Due to *pdftotext* limitations, fraction output is replaced by symbol output for per-mode and quotient-mode.

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

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

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

9.6.11 units and nicefrac packages

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

9.6.12 newtxmath package

Pkg newtxmath The proper load order is:

 **loading sequence**

```

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

```

9.7 Graphics

Pkg graphics For \includegraphics with .pdf or .eps files, the user must provide a .pdf or .eps image file for use in print mode, and also a .svg, .png, or .jpg version of the same image for use in HTML.

Pkg graphicx \includegraphics file formats

\includegraphics{filename} % print:.pdf/.eps HTML:.svg, etc.

For print output, l warp will automatically choose the .pdf or .eps format if available, or some other format otherwise. For HTML, one of the other formats is used instead.

If a .pdf or .eps image is referred to with its file extension, the extension will be changed to .svg for HTML:

```
\includegraphics{filename.pdf} % uses .svg in HTML
\includegraphics{filename.eps} % uses .svg in HTML
```

Prog pdftocairo To convert a PDF image to SVG, use the utility *pdftocairo*:

[PDF to SVG](#)

Enter ⇒ pdftocairo -svg filename.pdf

Prog l warpmk pdftosvg For a large number of images, use *l warpmk*:

Enter ⇒ l warpmk pdftosvg *.pdf (or a list of filenames)

Prog l warpmk epstopdf For EPS images converted to PDF using the package *epstopdf*, use

[epstopdf package](#)

to convert to SVG images.

[DVI latex](#) When using DVI *latex*, it is necessary to convert EPS to PDF and then to SVG:

Enter ⇒ l warpmk epstopdf *.eps (or a list of filenames)

Enter ⇒ l warpmk pdftosvg *.pdf (or a list of filenames)

[PNG and JPG](#)

For PNG or JPG while using *pdflatex*, *lualatex*, or *xelatex*, the same file may be used in both print or HTML versions, and may be used with a file extension, but will also be used without the file extension if it is the only file of its base name.

[GIF](#)

GIF files may be used for HTML, but another format must also be provided for print output.

[file extension priorities](#)

If a file extension is not used, for HTML the file extension priorities are: SVG, GIF, PNG, then JPG.

⚠ [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 l warp — the entire image will be shown.

⚠ [viewport](#)

⚠ [viewport units](#)

For *\includegraphics*, avoid px and % units for width and height, or enclose them inside *warpHTML* environments. For font-proportional image sizes, use ex or em. For fixed-sized images, use cm, mm, in, pt, or pc. Use the keys *width=.5\linewidth*, or similar for *\textwidth* or *\textheight* to give fixed-sized images proportional to a 6 by 9 inch text area. Do not use the *scale* option, since it is not well supported by HTML browsers.

options \includegraphics accepts width and height, origin, rotate and scale, plus new class and alt keys.

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.

HTML alt tags Likewise, the \includegraphics alt key adds an HTML alt tag to an image, and is ignored for print output. If not assigned, each image is given an alt tag of “(image)”.

\rotatebox \rotatebox accepts the optional origin key.

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

9.7.1 tikz package

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

⚠ displaymath and matrices

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

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

9.7.2 grffile package

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

9.7.3 color package

Pkg color color is superceded by xcolor, and lwarf requires several of the features of xcolor. When color is requested, xcolor is loaded as well.

9.7.4 xcolor package

Pkg xcolor \colorboxBlock and \fcolorboxBlock are provided for increased HTML compatibility.

\colorboxBlock and \fcolorboxBlock

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

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

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

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

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

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

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

9.7.5 epstopdf package

Pkg `epstopdf` Images with an `.eps` extension will be converted to `.pdf`. The HTML output uses the `.svg` version, so use

Enter ⇒ `lwarfmk pdftosvg <listofPDFfiles>`

to generate `.svg` versions.

9.7.6 pstricks package

Pkg `pstricks` All `pstricks` content should be contained inside a `pspicture` environment.

⚠ use `pspicture`

9.7.7 pdftricks package

Pkg `pdftricks` The `pdftricks` image files `<jobname>-fig*.pdf` must be converted to `.svg`, or else a missing file error will occur. The image files must also be converted again whenever they change. To convert the images:

Enter ⇒ `lwarfmk pdftosvg <jobname>-fig*.pdf`

9.7.8 psfrag package

- Pkg **psfrag** The `psfrags` environment is modified to use `lateximage` to encapsulate the image.
⚠ use psfrags Always use a `psfrags` environment to contain any local `\psfrag` macros and the associated `\includegraphics` or `\epsfig` calls. Outside of a `psfrags` environment, `psfrags` adjustments will not be seen by `l warp`.
- ⚠** Tip: Use a mono-spaced font for the tags in the EPS file.

9.7.9 pstool package

- Pkg **pstool** `\graphicspath` is ignored, and the file directory must be stated.
⚠ path and filename The filename must not have a file extension.

Use

Enter ⇒ `l warpmk html`

followed by

Enter ⇒ `l warpmk limages`

9.7.10 asymptote package

- Pkg **asymptote** To compile:

```
pdflatex project.tex
asy project-*.asy
pdflatex project.tex
```

```
l warpmk print
asy project-*.asy
l warpmk print1
l warpmk print1
```

```
l warpmk html
asy project_html-*.asy
l warpmk html1
l warpmk html1
l warpmk limages
```

9.7.11 overpic package

- Pkg **overpic** The macros `\overpicfontsize` and `\overpicfontskip` are used during HTML generation. These are sent to `\fontsize` to adjust the font size for scaling differences between the print and HTML versions of the document. Renew these macros before using the `overpic` and `Overpic` environments.

9.8 Tabbing

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

- ⚠️ math in tabbing** If math is used inside tabbing, place tabbing inside a `latexitimage` environment, which will render the entire environment as a single `svg` image.

9.9 Tabular

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

Defining environments:

- ⚠️ Misplaced alignment tab character &**

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

`\StartDefiningTabulars`

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

`\StopDefiningTabulars`

- ⚠️ floatrow**

This includes before and after defining any macro which used `\ttabbox` from `floatrow`.

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

- ⚠️ tabular inside another environment**

```
\StartDefiningTabulars % because & is used in a
definition
\newenvironment{outerenvironment}
{
\begin{array}{c|c}
left & right \\
\end{array}
}
\begin{array}{c|c}
\TabularMacro\ResumeTabular
left & right \\
\end{array}
\end{array}
\end{array}
\StopDefiningTabulars
```

Cell contents:**⚠ \multirow**

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

```
... & \multirow{2}{.5in}{text} & ...
... & \mrowcell & ...
```

vposn

Note that recent versions of multirow include a new optional vposn argument.

- The multirow documentation regarding colored cells recommends using a negative number of rows. This will not work with lwarp, so \warpprintonly and \warpHTMLonly must be used to make versions for print and HTML.

- See section 299.2 for \multicolumnrow.

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

```
\multicolumnrow{2}{c}{c}{3}{0}{1in}{Opt}{Text}
```

The two arguments for \multicolumn come first, followed by the five arguments for \multirow, many of which are optional, followed by the contents.

As per \multirow, skipped cells to the right of the \multicolumnrow statement are not included in the source code on the same line. On the following lines, \mcolrowcell must be used for each cell of each column and each row to be skipped:

```
... & \multicolumnrow{2}{c}{c}{3}{0}{1in}{Opt}{Text} & ...
... & \mcolrowcell & \mcolrowcell & ...
... & \mcolrowcell & \mcolrowcell & ...
```

vposn

Note that recent versions of multirow include a new optional vposn argument.

⚠ **macro in a table**

- Using a custom macro inside a tabular data cell may result in an extra HTML data cell tag, corrupting the HTML table. To avoid this, use `\TabularMacro` just before the macro. This is ignored in print mode.

`\TabularMacro\somemacro & more row contents \\`

Column specifiers:

⚠ *** column specification**

`@ and !`

`\multirow`

⚠ **\newcolumntype**

- * in a column specification is not used (so far). Repeat the column type the correct number of times.
- Only one each of @ and ! is used at each column, and they are used in that order.
- In \multirow cells, the print version may have extra instances of <, >, @, and ! cells on the second and later rows in the \multirow which do not appear in the HTML version.
- \newcolumntype is ignored; unknown column types are set to l.

Rules:

`vertical rules`

`width and trim`

`full-width rules`

`combined rules`

- Doubled \hlines, \midrules, and vertical rules are supported.
- Vertical rules next to either side of an @ or ! column are displayed on both sides of the column.
- Width options are honored. Trim options are converted to rounded top corners. Trim corners are not rounded with @ or ! columns, and full-width rules ignore trim.
- \toprule, \midrule, \bottomrule, and \hline ignore trim. When given an optional width, each cell is styled to create the custom border. Without an optional width, the entire row is given a class to assign the standard border.
- If you wish to use \cmidrule followed by \bottomrule, it may be necessary to use:

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

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

- 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.

colortbl:

⚠ **row/cell color**

Only use \rowcolor and \cellcolor at the start of a row, in that order. colortbl ignores the overhang arguments.

Other:

- tabularx ignores the width, but X columns do produce paragraph columns or multicolumns.

longtable headings

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

 **S columns**

- For S columns (from the siunitx package), while producing print output, anything non-numeric must be placed inside {} braces, including commands such as \multirow. While producing HTML output, though, anything placed inside braces is not seen by lwarf's tabular handling algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:

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

9.9.1 longtable package

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

`\warpprintonly{row contents}`

instead of

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

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

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

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

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

or place it inside \warpingprintonly.

 **lateximage** longtable is not supported inside a lateximage.

9.9.2 supertabular and xtab packages

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

Pkg xtab

 **Misplaced alignment tab character &**

```
\StartDefiningTabulars
\tablefirsthead
...
\StopDefiningTabulars
```

See section 9.9.

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

9.9.3 bigdelim package

Pkg **bigdelim** \ldelem and \rdelem use \multirow, so \mrowcell must be used in the proper number of empty cells in the same column below \ldelem or \rdelem, but not in cells which are above or below the delimiter:

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

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

9.10 Floats

9.10.1 Float contents alignment

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

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

9.10.2 float, trivfloat, and/or algorithmicx together

Pkg **float** If using \newfloat, trivfloat, and/or algorithmicx together, see section 429.1.

Pkg **trivfloat**

Pkg **algorithmx**

 **package conflicts**

9.10.3 caption and subcaption packages

Pkg `caption` To pass options to `caption`, select the options before loading `l warp`:

```
Pkg   subcaption      \documentclass{article}
      △  options        ...
          \PassOptionsToPackage{options_list}{caption}
          ...
          \usepackage{l warp}
          ...
          \usepackage{caption}
```

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

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

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

9.10.4 subfig package

Pkg `subfig`

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

In the document source, use `\hfill` and `\hspace*` subfig>inline between subfigures to spread them apart horizontally. The use of other forms of whitespace may cause paragraph tags to be generated, resulting in subfigures appearing on the following lines instead of all on a single line.

9.10.5 floatrow package

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

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

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

- Use fixed lengths. `l warp` will scale the HTML lengths appropriately.
- Use `warpprint` and `warpHTML` environments to select appropriate values for each case.

- Inside a `warpHTML` environment, manually change `\FBwidth` or `\FBheight` before the `\ffigbox` or `\ttabbox`. Use `\FBwidth` or `\FBheight` normally afterwards; it will be used as expected in print output, and will use your custom-selected value in HTML output. This custom value will be used repeatedly, until it is manually changed to a new value.

9.10.6 keyfloat package

Pkg `keyfloat` If placing a `\keyfig[H]` inside a `keywrap`, use an absolute width for `\keyfig`, instead of `1w`-proportional widths. (The `[H]` option forces the use of a `minipage`, which internally adjusts for a virtual 6-inch wide `minipage`, which then corrupts the `1w` option.)

⚠ `keywrap` If placing a `\keyfig[H]` inside a `keywrap`, use an absolute width for `\keyfig`, instead of `1w`-proportional widths. (The `[H]` option forces the use of a `minipage`, which internally adjusts for a virtual 6-inch wide `minipage`, which then corrupts the `1w` option.)

9.11 KOMA-SCRIPT classes

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

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

⚠ Not fully tested! Please send bug reports!

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

9.12 MEMOIR class

Cls `memoir` While emulating `memoir`, `l warp` pre-loads a number of packages (section 469.1). This can cause an options clash when the user's document later loads the same packages with options. To fix this problem, specify the options before loading `l warp`:

```
\documentclass{memoir}
...
\PassOptionsToPackage{options_list}{package_name}
...
\usepackage{l warp}
...
\usepackage{package_name}

\verb|footnote| is not supported.

\verb|newfootnoteseries|, etc. are not supported.
```

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

Poem numbering is not supported.

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

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

9.13 Miscellaneous packages

9.13.1 verse and memoir

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

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

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

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

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

9.13.2 newclude package

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

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

9.13.3 babel package

Pkg **babel**

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

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

punctuation spaces Also when French is used, l warp creates fixed-width space around punctuation by patching \FBcolonspace, \FBthinspace, \FBguillspace, \FBmedkern, \FBthickkern, \FBtextellipsis, and the tilde. If the user's document also changes these parameters, the user's changes should be placed inside a warpprint environment so that the user's changes do not affect the HTML output.

⚠ customized spacing

9.13.4 polyglossia package

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

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

To test compatibility, add

```
\usepackage{cleveref}
```

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

```
\setdefaultlanguage{english}
```

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

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

9.13.5 `todonotes` and `luatodonotes` packages

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

9.13.6 `fixme`

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

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

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

9.13.7 `chemfig` package

If using `\polymerdelim` to add delimiters to a `\chemfig`, wrap both inside a single `lateximage`:

```
\begin{lateximage}[(-chemfig-~\packagediagramname)]
\chemfig{...}
\polymerdelim[...]{...}
\end{lateximage}
```

9.13.8 `chemformula` package

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

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

9.13.9 mhchem package

See section 290.

9.13.10 xparse

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

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

10 Compiling using custom shell commands

lwarf and *lwarpmk* try to make it easy to process print and HTML compilation tasks in most situations. Depending on the operating system, command-line options, TEX engine, and *lwarf* options, the commands *lwarpmk print* and *lwarpmk html* are automatically set up to correctly recompile the project. These actions may be overridden using *lwarf* options, thus allowing the use of packages such as *perltx* and *pythontex*.

10.1 Command options

Opt `PrintLatexCmd` The *lwarf* options `PrintLatexCmd` and `HTMLLatexCmd` are used to set customized commands to be executed by *lwarpmk print* and *lwarpmk html*.

`PrintLatexCmd` should be set to shell commands which take `project.tex` and generate `project.pdf`.

`HTMLLatexCmd` should be set to take `project_html.tex` and generate `project_html.pdf`. *lwarpmk* will then take `project_html.pdf` and automatically convert it and generate `project.html`.

10.2 Literal character macros

The *lwarf* package options are parsed by TEX, and so some characters require the use of a special macro to represent them. See table 7. `\LWRopquote` and `\LWRopseq` may be used to increase operating-system portability. `\jobname` must have `_html` appended for processing HTML. `\space` may be necessary between other macros.

 **macro not found** To use these macros, either `kvoptions-patch` must be loaded before *lwarf*:

```
\usepackage{kvoptions-patch}
\usepackage[
    PrintLatexCmd={ ... } ,
    HTMLLatexCmd={ ... }
]{lwarf}
```

Table 7: Literal character macros

Character	Macro	Comment
%	\LWRpercent	
\$	\LWRdollar	
&	\LWRamp	
%	\LWRhash	
\	\LWRbackslash	
' or "	\LWRopquote	Depends on the operating system.
& or &&	\LWRopseq	Depends on the operating system.
(space)	\space	Forces an extra space.
(jobname)	\jobname	Without file extension.

or \lwarfsetup must be used to set PrintLatexCmd and HTMLLatexCmd:

```
\usepackage[...]{lwarf}
\lwarfsetup{
    PrintLatexCmd=
    {
        latex tm \LWRopseq
        dvips -o tm-pics.ps tm.dvi \LWRopseq
        ps2pdf tm-pics.ps \LWRopseq
        pdflatex tm.tex
    } ,
    HTMLLatexCmd=
    {
        latex tm_html \LWRopseq
        dvips -o tm_html-pics.ps tm_html.dvi \LWRopseq
        ps2pdf tm_html-pics.ps \LWRopseq
        pdflatex tm_html.tex
    }
}
```

10.3 *latexmk*

Pkg `latexmk` If *latexmk* is used for a project, it may be easiest to continue using it.

`latexmk project.tex` would create `project.pdf` as normal.

`latexmk project_html.tex` would create `project_html.pdf`, then

`l warpmk pdftohtml project_html.pdf` would take `project_html.pdf` and convert it to `project.html`.

Pkg `sagetex` *latexmk* may simplify the use of packages such as `sagetex`.

10.4 *perltx* package

Pkg `perltx` The l warp package option settings to use perltx would be similar to:

```
\usepackage[
    ...
    PrintLatexCmd={perltx -latex=pdflatex project.tex} ,
    HTMLLatexCmd={perltx -latex=pdflatex project_html.tex} ,
    ...
]{l warp}
```

⚠ “impure” math Place perltx math expressions between `\displaymathother` and `\displaymathnormal`, or `\inlinemathother` and `\inlinemathnormal`. See section 9.6.8.

10.5 *pythontex* package

Pkg `pythontex` An example using pythontex:

```
\usepackage[
    ...
    PrintLatexCmd={
        pdflatex project.tex \LWRopseq
        pythontex project \LWRopseq
        pdflatex project.tex
    } ,
    HTMLLatexCmd={
        pdflatex project_html.tex \LWRopseq
        pythontex project_html \LWRopseq
        pdflatex project_html.tex
    } ,
    ...
]{l warp}
```

Another possibility is to use *latexmk*, placing the `latexmk ...` commands in the `PrintLatexCmd` and `HTMLLatexCmd` options. While using these options, the `lwarp` option `latexmk` would not be used.

- ⚠ “impure” math
 - ⚠ HTML look-alike
- No attempt has yet been made to make `pythontex` robust with HTML output. Some math objects must be surrounded by `\displaymathother ... \displaymathnormal`, or `\inlinemathother ... \inlinemathnormal`. Displays of code may have to be enclosed inside a `lateximage` environment to prevent <, > and similar from being interpreted by the browser as HTML entities.

10.6 Other packages

- Pkg `sympytex` Other packages such as `sympytex` and `rterface` would be set up similar to `pythontex`, and the same warnings would apply.
Pkg `rterface`

10.7 *make* program

- Prog `make` To use `lwarp` with the *make* program, have the `makefile` take `project.tex` and generate the print version `project.pdf`, as normal. `\usepackage{lwarp}` must be used, and it generates `lwarpmk.conf` when the print version is created.

To generate HTML, first have `project_html.tex` be compiled to generate `project_html.pdf`. This must be in PDF format. Finally, have `project_html.pdf` be converted to HTML using `lwarpmk pdftohtml project_html.pdf`, and convert SVG math with `lwarpmk limages`.

10.8 UTF-8 locale

- ⚠ UTF-8 locale
- lwarpmk* uses the `texlua` program, which sets the “locale” to “C”, including for external operating-system calls such as when executing `lwarpmk html`. In some cases, an external program called from the user’s document may require the use of a UTF-8 “locale”. For UNIX-related operating systems, it may be required to use `lwarp`’s custom compilation options to add a locale change:

```
\usepackage{lwarp}[
  PrintLatexCmd={
    env LC_CTYPE=en_US.UTF-8
    xelatex -shell-escape project.tex
  }
  HTMLLatexCmd={
    env LC_CTYPE=en_US.UTF-8
    xelatex -shell-escape project_html.tex
  }
]
```

- Pkg ditaa The only example seen so far where this is required is the ditaa package, where the locale change allows the use of UTF-8 with Xe^LA_ET_X and ditaa. To use Lu^AI_ET_X instead, the locale change would have to be made inside the ditaa package where its calls the *ditaa* program.

11 EPUB conversion

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

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

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

FormatEPUB

Bool FormatEPUB
Default: false

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

To help convert lwarp HTML output to EPUB, add

\booltrue{FormatEPUB}

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

Enter ⇒ lwarpmk cleanall

Enter ⇒ lwarpmk html

Enter ⇒ lwarpmk limages

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

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

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

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

Preferences → Plugins → File type plugins → HTML to Zip

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

 **section breaks** The EPUB-conversion program must also know where the section breaks are located. For a list of lwarp's section headings, see table 9. For example, an article class document would break at \section, which is mapped to HTML heading level <h4>, whereas a book class document would break at \chapter, which is HTML heading level <h3>. For *Calibre*, this option is found in

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

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

Match HTML tags with tag name:

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

//h:h4 — or — //h:h3

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

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

MATHJAX support

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

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

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

12 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.

12.1 Activating word-processor conversion

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

FormatWP

Bool FormatWP
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. Additionally, honors the booleans WPMarkFloats, WPMarkMinipages, WPMarkTOC, and WPMarkLOFT.

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

`\booltrue{FormatWP}`

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

- If using a class without chapters, `\section` and lower are shifted up in level for the HTML heading tags. The css has not been changed, so the section heading formats will not match the normal HTML output, but when imported to *LibreOffice Writer* the higher section headings will import as **Heading 1** for the title, **Heading 2** for `\section`, etc.
- Headers, footers, and navigation are removed at file splits.
- Any accumulated footnotes are printed at the bottom of each section.
- Forces single-file output.
- Turns off HTML debugging comments. These are comments appearing inside the HTML code, marking the opening/closing of sections and `<div>`s, but they are no longer useful when the document has been imported into a word processor.

- An additional `<div>` with an `id` encapsulates each float and `minipage`, which on import into *LibreOffice Writer* causes a thin frame to appear around the text block for each.
- Float captions are given an explicit italic formatting.
- Tabular rule borders are made explicit for *LibreOffice Writer*. LIBREOFFICE displays a light border around each cell while editing, even those which have no border when printed, and lwarp also uses a light border for thin rules, so it will be best to judge the results using the print preview instead of while editing in LIBREOFFICE.
- `\includegraphics` and `svg` math width and height are made explicit for LIBREOFFICE.
- `\hspace` is approximated by a number of `\quads`, and rules are approximated by a number of underscores.
- Explicit HTML styles are given to:
 - `\textsc`, etc.
 - `\underline`, `soul` and `ulem` markup.
 - `center`, `flushleft`, `flushright`.
 - `\marginpar`, `keyfloat`, `sidenotes`, `floatflt`, and `wrapfig`.
 - `fancybox` `\shadowbox`, etc.
 - The L^AT_EX and T_EX logos.

- Honors several booleans:

WPMarkFloats: Marks the begin and end of floats.

WPMarkMinipages: Marks the begin and end of minipages.

WPMarkTOC: Marks the location of the Table of Contents.

WPMarkLOFT: Marks the locations of the List of Figures/Tables.

WPMarkMath: Prints L^AT_EX math instead of using images.

WPTitleHeading: Adjusts title and section headings.

Several of these may be used to add markers to the HTML text which help determine where to adjust the word processor document after import.

12.2 Additional modifications

WPMarkFloats

Adds

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

or

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

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

WPMarkMinipages

Adds

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

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

WPMarkTOC

While formatting for word processors, adds

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

Bool WPMarkTOC

Default: true

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

If set false, the actual toc is printed instead.

WPMarkLOFT

While formatting for word processors, adds

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

Bool WPMarkLOFT

Default: false

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

If set false, the actual lists are printed instead.

WPMarkMath

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

Bool WPMarkMath

Default: false

Prog TeXMaths

siunitx

When using the siunitx package, enter

```
\usepackage{siunitx}
```

in the *TeXMaths* preamble. Equation numbering is problematic for *AMS* math environments.

WPTitleHeading

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

See table 8 on page 190.

12.3 Recommendations

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

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

Table 8: Section HTML headings for word-processor conversion

Section	HTML headings*			
	With \chapter		Without \chapter	
	WPTitleHeading	WPTitleHeading	WPTitleHeading	WPTitleHeading
Title	true	plain	<h1>	plain
\part	<h2>	<h1>	<h2>	<h1>
\chapter	<h3>	<h2>	—	—
\section	<h4>	<h3>	<h3>	<h2>
\subsection	<h5>	<h4>	<h4>	<h3>
\subsubsection	<h6>	<h5>	<h5>	<h4>
\paragraph	span	<h6>	<h6>	<h5>

* For default depths when not FormatWP, see table 9 on page 204.

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

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

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

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

12.4 Limitations

FLOATS AND CAPTIONS ARE NOT EXPLICITLY CONVERTED TO LIBREOFFICE FLOATS WITH THEIR OWN CAPTIONS. FLOATS ARE SURROUNDED BY A THIN FRAME IN THE LIBREOFFICE EDITOR, AND MAY BE MARKED WITH WPMARKFLOATS, BUT ARE NOT GIVEN A PROPER LIBREOFFICE OBJECT FRAME. CAPTIONS ARE GIVEN AN EXPLICIT ITALIC FORMATTING, BUT NOT A PROPER LIBREOFFICE PARAGRAPH STYLE.

CROSS REFERENCES ARE NOT ACTUAL LIBREOFFICE LINKED CROSS REFERENCES.

The List of Figures and List of Tables are not linked. The pasted pseudo LOF and LOT match the numbering of the L^AT_EX and HTML versions.

Equation numbering is not automatic, but the equation numbers in SVG math will match the L^AT_EX and HTML output. SVG math is recommended when using the *AMS* environments, which may have multiple numbered equations per object.

As of when last checked, LIBREOFFICE ignores the following:

- Minipage alignment.
- Tabular cell vertical alignment.
- Image rotation and scaling.
- Rounded border corners, which are also used by:
 - \textcircled
 - booktabs trim
- \hspace and rules, also used by algorithmic.
- Coloring of text decorations, used by soul and ulem.
- Overline text decoration, used by romanbar.

LIBREOFFICE also has limitations with frames and backgrounds:

- Multiple lines in an object are framed individually instead of as a whole.
- Nested frames are not handled correctly.
- Images inside boxes are not framed correctly.
- Spans with background colors and frames are not displayed correctly.

13 Modifying lwarp

- locating something** To quickly find the source for a package in `lwarp.dtx`, search for `*packagename`, such as `*siunitx`. Likewise, to quickly find the source for a file in `lwarp.dtx`, search for `*filename`, such as `*lwarp.css`. Purely text-based packages probably will work as-is when generating `HTML`. Look to existing code for ideas on how to expand into new code.
- image of TeX output** An environment may be converted to a `lateximage` then displayed with an image of the resulting `LATEX` output. See section 87 for an example of the `picture` environment.
- css classes** To create a custom `HTML` block or inline `css` class, see section 52.8.
- print/HTML macros** To create print and `HTML` versions of the same macro or environment, see section 36.
- ⚠ TEX boxes** Any `TEX` boxes must be undone, as `SVG` math or `lateximages` require `\newpage`, which will not work in a `TEX` box.
- index recreation** To recreate the index for the `lwarp` documentation:

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

13.1 Creating a development system

The following creates a local development system for `lwarp` on a `TeXLive` system in a `UNIX`-like environment. Doing so allows anything requesting `lwarp` to use the development version instead of whichever version is installed in `TeXLive`.

Create a development directory:

Place into this directory `lwarp.dtx` and `lwarp.ins`.

To create `lwarp.sty`, execute

Enter ⇒ `pdflatex lwarp.ins`

which creates `lwarp.sty` and several hundred additional `lwarp-*.sty` files for the various packages which are supported.

To create the documentation `lwarp.pdf`, execute

Enter ⇒ pdflatex lwarf.dtx

To make the development files visible to other projects:

Create the directory

/usr/local/texlive/texmf-local/tex/latex/local/lwarf

Inside this directory, create the file update, containing:

```
ln -s /path_to_dev_directory/lwarf*.sty .
ln -s /path_to_dev_directory/lwarf_baseline_marker.png .
ln -s /path_to_dev_directory/lwarf_baseline_marker.eps .
mktextsls
```

Run ./update now, and whenever a new lwarf-* package is added.

To make the development version of lwarpmk visible to other projects:

```
cd /opt
ln -s /usr/local/texlive/texmf-local/bin/x86_64-linux texbin_local
cd texbin_local
ln -s ../../scripts/lwarf/lwarpmk.lua lwarpmk
cd /usr/local/texlive/texmf-local/scripts/
mkdir lwarf
cd lwarf
ln -s /path_to_dev_directory/lwarpmk.lua lwarpmk
```

Verify that the correct version is found with

Enter ⇒ which lwarpmk

To make the local versions visible to the shell:

Paths must be set by the shell startup, such as in .bashrc and .cshrc:

In .bashrc:

PATH=/opt/texbin_local:/opt/texbin:\$PATH

In .cshrc:

setenv PATH \${HOME}/bin:/opt/texbin_local:/opt/texbin:\${PATH}

13.2 Modifying a package for l warp

If a class loads additional packages, it will be required to modify the class for l warp, since l warp must be loaded before most other packages.

To work with l warp, a class must first set up anything which replicates the functions of the basic L^AT_EX classes, load any required fonts, then load l warp, then finally load and adjust any other required packages.

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

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

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

Each l warp-* package should first call either of:

```
\LWR@ProvidesPackageDrop  
-OR-  
\LWR@ProvidesPackagePass
```

If “Drop”ped, the original print-version package is ignored, and only the l warp- version is used. Use this where the original print version is useless for HTML. If “Pass”ed, the original package is loaded first, with the user-supplied options, then the l warp- version continues loading as well. See section 313 (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 l warp- version.

13.2.1 Adding a package to the lwarp.dtx file

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

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

to `lwarp.dtx` in section 21.1. This causes `lwarp` to stop with an error if `packagename` is loaded before `lwarp`. Finally, add an entry in table 2, [Supported packages and features](#), and also the Updates section.

13.3 Modifying a class for lwarp

If a class loads additional packages, it will be required to modify the class for `lwarp`, since `lwarp` must be loaded before most other packages.

To work with `lwarp`, a class must first set up anything which replicates the functions of the basic L^AT_EX classes, load any required fonts, then load `lwarp`, then finally load and adjust any other required packages.

13.4 Testing lwarp

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

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

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

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

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

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

13.5 Modifying *lwarpmk*

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

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

1. Add the `lwarpmk` option to the `lwarp` 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`.

14 Troubleshooting

14.1 Using the `lwarp.sty` package

Also see:

Section 8.8: Commands to be placed into the `warpprint` environment

Section 9: Special cases and limitations

Text is not converting correctly / corrupted HTML tags:

- Font-related UTF-8 information must be embedded in the PDF file. See section 8.2 regarding bitmapped vs. vector fonts.
- See section 9.2.1 regarding HTML entities and the characters &, <, and >.

Undefined HTML settings:

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

Tabular problems: See section 9.9.

Obscure error messages:

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

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

Options clash: If using memoir, see section 9.12.

“No room for a new `\write`.”: Before `\usepackage{lwarp}`, add:

```
\usepackage{morewrites}  
\morewritessetup{allocate=10}
```

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

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

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

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

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

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

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

```
\LetLtxMacro{\ensuredmath}{\LWR@origensuredmath}
```

inside a group first.

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

If using BibTeX, see section 9.5.9.

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

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

```
\begin{equation}
...
\end{equation}
```

“Misplaced \omit”: If using \LWR@formatted to define new macros for print and HTML modes, see section 36 regarding \LWR@expandableformatted.

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

```
\displaymathother -or- \inlinemathother  
before the math, and then
```

```
\displaymathnormal -or- \inlinemathnormal  
when displaying “normal” math. See section 9.6.8.
```

Slow compilation of math objects: Complicated math objects can also cause problems with alt tags, resulting in very slow compilation, large alt tags, and possible crashes. Use \inlinemathother ... \displaymathnormal or \displaymathother ... \displaymathnormal around the math expression.

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

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

Misnumbered footnotes from section headings: See section 9.4.4.

Missing HTML files:

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

Missing / incorrect cross-references:

- Use lwarfpmk again followed by lwarfpmk html or lwarfpmk print to compile the document one more time.
- Labels with special characters may be a problem. It is best to stick with alpha-numeric, hyphen, underscore, and perhaps the colon (if not French).
`\nameref` refers to the most recently-used section where the `\label` was defined. If no section has been defined before the `\label`, the link will be empty. Index entries also use `\nameref` and have the same limitation.
- cleveref and varioref are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used for `\cpageref` and `\cpagerefrange`. This phrase includes `\cpagerefFor`, which defaults to "for".

Ex:

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

See `\cpagerefFor` at page 576 to redefine the message which is printed for page number references.

BibTeX errors with \etalchar: See section 9.5.9.

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

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

Use X_ELATEX 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.

Images are appearing in strange places:

- Enter `lwarfpmk limages` to refresh the lateximage images.

SVG images:**⚠️ adding/removing**

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

Before attempting to create the `svg` image files, `l warpmk` verifies that the `HTML` version of the document exists and has correct internal image references.¹⁵ If it is necessary to recompile the document's `HTML` version, `l warpmk` will inform so with an error message.

If `HTML` appears where an `svg` image should be, recompile the document one more time to get the page numbers back in sync, then remake the images one more time. `l warpmk` attempts to detect this situation and print a warning.

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

⚠️ page counter

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

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

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

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 `l warpmk clean`, recompile, then start by reloading the home page. You may have been looking at an older version of the document. If you changed a section name, you may have been looking at the file for the old name.
- See the warning regarding changes to the `HTML` settings at section 8.4.
- Verify that the proper `css` is actually being used.

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

- 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.

14.1.1 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.

14.2 Compiling the lwarp.dtx file

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

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

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

Un-nested environments:

Be sure to properly nest:

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

15 Trademarks

- TEX is a trademark of American Mathematical Society.
- ADOBE® and ADOBE *Framemaker*® are either registered trademarks or trademarks of ADOBE SYSTEMS INCORPORATED in the United States and/or other countries.
- LINUX® is the registered trademark of Linus Torvalds in the U.S. and other countries.
- MAC OS® is a trademark of APPLE INC.
- MADCAP FLARE™ is the property of MADCAP SOFTWARE, INC.
- MATHJAX is copyright 2009 and later. The MATHJAX CONSORTIUM is a joint venture of the AMERICAN MATHEMATICAL SOCIETY (AMS) and the SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS (SIAM) to advance mathematical and scientific content on the web.
- MICROSOFT®, ENCARTA, MSN, and WINDOWS® are either registered trademarks or trademarks of MICROSOFT CORPORATION in the United States and/or other countries.
- UNIX® is a registered trademark of THE OPEN GROUP.

File 1 **lwarf.sty**

16 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 haeks* clever tricks. Reference sources are given for many of the solutions, and a quick internet search will provide additional possibilities.

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

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

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

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

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

Table 9: 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	
subparagraph	5	
listitem	7	new for this package, used for list items

* If FormatWP is true, section headings may be adjusted, depending on WPTitleHeading. See table 8 on page 190.

17 Section depths and HTML headings

Stacks are created to track depth inside the LATEX document structure. This depth is translated to HTML headings as shown in table 9. “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.

18 Source code

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

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

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

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

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

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

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

for HTML output: Green-colored tags in the left margin show which sections of source code apply to

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

for HTML & PRINT:

19 Detecting the TeX Engine — *pdflatex*, *lualatex*, *xelatex*

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

Detects XeTeX and LuaTeX:

```

1 \RequirePackage{ifutex}
2 \newif\ifxetexorluatex
3 \ifXeTeX
4   \xetexorluatextrue
5 \else
6   \ifLuaTeX
7     \xetexorluatextrue
8   \else
9     \xetexorluatexfalse
10  \fi
11 \fi
12
13 \ifLuaTeX
14 \RequirePackage{luatex85}% until the geometry package is updated
15 \fi
16
17 \RequirePackage{ifpdf}

18 \RequirePackage{ifptex}
```

20 Early package requirements

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

Pkg `xpatch` Patches macros with optional arguments.

```

19 \RequirePackage{etoolbox}[2011/01/03]%
20 \RequirePackage{xpatch}
```

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

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

Pkg `letltxmacro`

```
22 \RequirePackage{letltxmacro}
```

21 Package load order

Several packages must never be used with l warp, others should only be loaded before l warp, and others should only be loaded after. The l warp core checks most of these cases. In some l warp-* packages, \LWR@loadbefore is used to trigger an error if they are loaded after l warp, while additional code provides necessary patches for when they are loaded before.

Packages which must be loaded after l warp are enforced by a large number of \LWR@loadafter statements, below. Some packages are emulated by memoir, and so these are tested by \LWR@notmemoirloadafter, which does not cause an error if memoir is used.

\LWR@checkloadfilename is used to check each filename to see if it must never be loaded, or must always be loaded before l warp.

21.1 Tests of package load order

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

```

23 \newcommand*{\LWR@loadafter}[1]{%
24 \@ifpackageloaded{#1}%
25 {%
26   \PackageError{l warp}%
27     {Package #1, or one which uses #1, must be loaded after l warp}%
28     {Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{l warp}.}%
29     Package #1 may also be loaded by something else, which must also be moved%
30     after l warp.}%
31 }%
32 {}%
33 }
```

\LWR@notmemoirloadafter {*packagename*} Error if not memoir class and this package was loaded before l warp.

memoir emulates many packages, and pretends that they have already been loaded.

```

34 \@ifclassloaded{memoir}%
35 {\newcommand*{\LWR@notmemoirloadafter}[1]{}}
36 {\LetLtxMacro{\LWR@notmemoirloadafter}{\LWR@loadafter}}
```

\LWR@notltjloadafter {*packagename*} Error if not a ltjs* class and this package was loaded before l warp.

```

37 \LetLtxMacro{\LWR@notltjloadafter}{\LWR@loadafter}
38
39 \@ifclassloaded{ltjarticle}{\renewcommand*{\LWR@notltjloadafter}[1]{}{}}
40 \@ifclassloaded{ltjbook}{\renewcommand*{\LWR@notltjloadafter}[1]{}{}}
41 \@ifclassloaded{ltjreport}{\renewcommand*{\LWR@notltjloadafter}[1]{}{}}
42 \@ifclassloaded{ltjsarticle}{\renewcommand*{\LWR@notltjloadafter}[1]{}{}}
43 \@ifclassloaded{ltjsbook}{\renewcommand*{\LWR@notltjloadafter}[1]{}{}}
44 \@ifclassloaded{ltjsreport}{\renewcommand*{\LWR@notltjloadafter}[1]{}{}}
45 \@ifclassloaded{ltjspf}{\renewcommand*{\LWR@notltjloadafter}[1]{}{}}
46 \@ifclassloaded{ltjskiyou}{\renewcommand*{\LWR@notltjloadafter}[1]{}{}}
47 \@ifclassloaded{ltjarticle}{\renewcommand*{\LWR@notltjloadafter}[1]{}{}}
48 \@ifclassloaded{ltjtbook}{\renewcommand*{\LWR@notltjloadafter}[1]{}{}}
49 \@ifclassloaded{ltjtreport}{\renewcommand*{\LWR@notltjloadafter}[1]{}{}}

```

\LWR@loadbefore {*packagename*} Error if this package is loaded after lwarf.

```

50 \newcommand*{\LWR@loadbefore}[1]{%
51 \@ifpackageloaded{#1}%
52 {}%
53 {%
54 \PackageError{lwarf}%
55 {Package #1 must be loaded before lwarf}%
56 {Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{lwarf}.}%
57 }%
58 }

```

\LWR@checkloadbefore {*thispackagename*} {*packagename*}

If package names match, error if it is loaded after lwarf.

```

59 \newcommand*{\LWR@checkloadbefore}[2]{%
60     \edef\LWR@tempone{#1}%
61     \ifdefstring{\LWR@tempone}{#2}{%
62         \LWR@loadbefore{#1}%
63     }{}%
64 }

```

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

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

```

65 \newcommand*{\LWR@loadnever}[2]{%
66 \PackageError{lwarf}%
67 {Package #1 is not supported by lwarf's HTML conversion.}%
68 {Package(s) #2 may be useful instead}%
69 {Package #1 might conflict with lwarf in some way,}%
70 {or is superceded by another package.}

```

```
71 For a possible alternative, see package(s) #2.}
72 }
```

\LWR@checkloadnever {\⟨thispackagename⟩} {\⟨badpackagename⟩} {\⟨replacementpkgnames⟩}

If this package name is the bad packagename, suggest the replacements instead.

```
73 \newcommand*{\LWR@checkloadnever}[3]{%
74     \edef\LWR@tempone{\#1}%
75     \ifdefstring{\LWR@tempone}{\#2}{%
76         \LWR@loadnever{\#2}{\#3}%
77     }{}%
78 }
```

\LWR@earlyloadnever {\⟨badpackagename⟩} {\⟨replacementpkgname⟩}

The first package is not supported, so tell the user to use the second instead. This version checks immediately for packages which may have been loaded before lwarp.

```
79 \newcommand*{\LWR@earlyloadnever}[2]{%
80 \@ifpackageloaded{\#1}{%
81 \PackageError{lwarp}{%
82 {Package #1 is not supported by lwarp's HTML conversion.}%
83 {Package(s) #2 may be useful instead}%
84 {Package #1 might conflict with lwarp in some way,}%
85 {or is superceded by another package.}%
86 {For a possible alternative, see package(s) #2.}%
87 }{}%
88 }
```

\LWR@earlyclassloadnever {\⟨badclassname⟩} {\⟨replacementclassname⟩}

The first class is not supported, so tell the user to use the second instead. This version checks immediately for classes which may have been loaded before lwarp.

```
89 \newcommand*{\LWR@earlyclassloadnever}[2]{%
90 \@ifclassloaded{\#1}{%
91 \PackageError{lwarp}{%
92 {Class #1 is not supported by lwarp's HTML conversion.}%
93 {Class(es) #2 may be useful instead}%
94 {Class #1 might conflict with lwarp in some way,}%
95 {or is superceded by another class.}%
96 {For a possible alternative, see class(es) #2.}%
97 }{}%
98 }
```

21.2 Error for disallowed packages and classes loaded before l warp

```

99 \LWR@earlyclassloadnever{jarticle}{ujarticle}
100 \LWR@earlyclassloadnever{jbook}{ujbook}
101 \LWR@earlyclassloadnever{jreport}{ujreport}
102 \LWR@earlyclassloadnever{tarticle}{utarticle}
103 \LWR@earlyclassloadnever{tbook}{utbook}
104 \LWR@earlyclassloadnever{treport}{utreport}
105 \LWR@earlyloadnever{ae}{cm-super, lmodern}
106 \LWR@earlyloadnever{aecmpl}{cm-super, lmodern}
107 \LWR@earlyloadnever{aecc}{cm-super, lmodern}
108 \LWR@earlyloadnever{boxedminipage}{boxedminipage2e}
109 \LWR@earlyloadnever{caption2}{caption}
110 % \LWR@earlyloadnever{ccaption}{caption} might be preloaded by memoir

```

The older CJK and CJKutf8 only work with xeCJK:

```

111 \cifpackage{xeCJK}{}{
112     \LWR@earlyloadnever{CJK}{ctex, xeCJK}
113     \LWR@earlyloadnever{CJKutf8}{ctex, xeCJK}
114 }

```

bxcjkjatype is based on CJK:

```
115 \LWR@earlyloadnever{bxcjkjatype}{upLaTeX, bxjsarticle, ujarticle, utarticle}
```

hangul is not in TeXLive, and is not tested:

```
116 \LWR@earlyloadnever{hangul}{kotex, xetexko, luatexko}
```

Others:

```

117 \LWR@earlyloadnever{fancyheadings}{fancyhdr}
118 \LWR@earlyloadnever{glossary}{glossaries}
119 \LWR@earlyloadnever{t1enc}{fontenc, inputenc, inputenx}
120 \LWR@earlyloadnever{wasysym}{textcomp, amssymb, amsfonts, mnsymbol, fdsymbol}

```

21.3 Enforcing package loading after l warp

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

The following packages must be loaded after l warp:

```

121 \LWR@loadafter{2in1}
122 \LWR@loadafter{2up}
123 \LWR@loadafter{a4}
124 \LWR@loadafter{a4wide}

```

```
125 \LWR@loadafter{a5comb}
126 \LWR@notmemoirloadafter{abstract}
127 \LWR@loadafter{accsupp}
128 \LWR@loadafter{acro}
129 \LWR@loadafter{acronym}
130 \LWR@loadafter{adjmulticol}
131 \LWR@loadafter{addlines}
132 \LWR@loadafter{ae}
133 \LWR@loadafter{aecc}
134 \LWR@loadafter{afterpage}
135 \LWR@loadafter{algorithm2e}
136 \LWR@loadafter{algorithmicx}
137 \LWR@loadafter{alltt}
138 \LWR@loadafter{amsmath}
139 \LWR@loadafter{amsthm}
140 \LWR@loadafter{anonchap}
141 \LWR@loadafter{anysize}
142 \LWR@notmemoirloadafter{appendix}
143 \LWR@loadafter{arabicfront}
144 \LWR@notmemoirloadafter{array}
145 \LWR@loadafter{arydshln}
146 \LWR@loadafter{asymptote}
147 % \LWR@loadafter{atbegshi} % used by morewrites
148 \LWR@loadafter{attachfile}
149 \LWR@loadafter{attachfile2}
150 \LWR@loadafter{authblk}
151 \LWR@loadafter{axessibility}
152 \LWR@loadafter{axodraw2}
153 \LWR@loadafter{backref}
154 \LWR@loadafter{balance}
155 \LWR@loadafter{bigdelim}
156 \LWR@loadafter{bigstrut}
157 \LWR@loadafter{bitpattern}
158 \LWR@loadafter{blowup}
159 \LWR@loadafter{booklet}
160 \LWR@loadafter{bookmark}
161 \LWR@notmemoirloadafter{booktabs}
162 \LWR@loadafter{bophook}
163 \LWR@loadafter{bounddvi}
164 \LWR@loadafter{boxedminipage}
165 \LWR@loadafter{boxedminipage2e}
166 \LWR@loadafter{breakurl}
167 \LWR@loadafter{breqn}
168 \LWR@loadafter{bsheaders}
169 \LWR@loadafter{bxpaper-size}
170 \LWR@loadafter{bytefield}
171 \LWR@loadafter{cancel}
172 \LWR@loadafter{canoniclayout}
173 \LWR@loadafter{caption}
174 \LWR@loadafter{caption2}
```

```
175 \LWR@loadafter{cases}
176 % \LWR@loadafter{ccaption} may be preloaded by memoir
177 \LWR@loadafter{changebar}
178 \LWR@notmemoirloadafter{changepage}
179 \LWR@notmemoirloadafter{chngpage}
180 \LWR@loadafter{chappg}
181 \LWR@loadafter{chapterbib}
182 \LWR@loadafter{chemfig}
183 \LWR@loadafter{chemformula}
184 \LWR@loadafter{chemgreek}
185 \LWR@loadafter{chemmacros}
186 \LWR@loadafter{chemnum}
187 \LWR@loadafter{chkfloat}
188 \LWR@loadafter{cite}
189 \LWR@loadafter{cmdtrack}
190 \LWR@loadafter{color}
191 \LWR@loadafter{colortbl}
192 \LWR@loadafter{continue}
193 \LWR@loadafter{copyrightbox}
194 \LWR@notmemoirloadafter{crop}
195 % ctex must be loaded before lwarf
196 \LWR@loadafter{cuted}
197 \LWR@loadafter{cutwin}
198 \LWR@loadafter{dblfloatfix}
199 \LWR@loadafter{dblfnote}
200 \LWR@notmemoirloadafter{dcolumn}
201 \LWR@loadafter{diagbox}
202 \LWR@loadafter{dprogress}
203 \LWR@loadafter{draftcopy}
204 \LWR@loadafter{draftfigure}
205 \LWR@loadafter{draftwatermark}
206 \LWR@loadafter{easy-todo}
207 \LWR@loadafter{ebook}
208 \LWR@loadafter{ellipsis}
209 \LWR@loadafter{embrac}
210 \LWR@loadafter{emptypage}
211 \LWR@loadafter{endfloat}
212 \LWR@loadafter{endheads}
213 \LWR@loadafter{endnotes}
214 \LWR@notmemoirloadafter{enumerate}
215 \LWR@loadafter{enumitem}
216 \LWR@notmemoirloadafter{epigraph}
217 \LWR@loadafter{epsfig}
218 \LWR@loadafter{epstopdf}
219 \LWR@loadafter{epstopdf-base}
220 \LWR@loadafter{errata}
221 \LWR@loadafter{eso-pic}
222 \LWR@loadafter{everypage}
223 \LWR@loadafter{everyshi}
224 \LWR@loadafter{extramarks}
```

```
225 \LWR@loadafter{fancybox}
226 \LWR@loadafter{fancyhdr}
227 \LWR@loadafter{fancyheadings}
228 \LWR@loadafter{fancyref}
229 \LWR@loadafter{fancytabs}
230 \LWR@loadafter{fancyvrb}
231 \LWR@loadafter{figcaps}
232 \LWR@loadafter{figsize}
233 \LWR@loadafter{fix2col}
234 \LWR@loadafter{fixme}
235 \LWR@loadafter{fixmetodonotes}
236 \LWR@loadafter{flafter}
237 \LWR@loadafter{flippdf}
238 \LWR@loadafter{float}
239 \LWR@loadafter{floatflt}
240 \LWR@loadafter{floatpag}
241 \LWR@loadafter{floatrow}
242 \LWR@loadafter{fltrace}
243 \LWR@loadafter{flushend}
244 \LWR@loadafter{fnbreak}
245 \LWR@loadafter{fncychap}
246 \LWR@loadafter{fnlineno}
247 \LWR@loadafter{fnpos}
248 % fontenc must be loaded before lwarf
249 % fontspec must be loaded before lwarf
250 \LWR@loadafter{footmisc}
251 \LWR@loadafter{footnote}
252 \LWR@loadafter{footnotehyper}
253 \LWR@loadafter{footnoterange}
254 \LWR@loadafter{footnpag}
255 \LWR@loadafter{forest}
256 \LWR@loadafter{framed}
257 \LWR@loadafter{ftnright}
258 \LWR@loadafter{fullminipage}
259 \LWR@loadafter{fullpage}
260 \LWR@loadafter{fullwidth}
261 \LWR@loadafter{fwlw}
262 \LWR@loadafter{gentombow}
263 % geometry is always loaded by lwarf, and lwarf-geometry is AtBeginDocument
264 \LWR@loadafter{gmeometric}
265 \LWR@loadafter{glossaries}
266 % \LWR@loadafter{graphics}% pre-loaded by xunicode
267 % \LWR@loadafter{graphicx}% pre-loaded by xunicode
268 \LWR@loadafter{glossary}
269 \LWR@loadafter{grffile}
270 \LWR@loadafter{grid}
271 \LWR@loadafter{grid-system}
272 \LWR@loadafter{gridset}
273 \LWR@loadafter{hang}
274 \LWR@loadafter{hanging}
```

```
275 \LWR@loadafter{hypcap}
276 \LWR@loadafter{hypdestopt}
277 \LWR@loadafter{hypernat}
278 \LWR@loadafter{hyperref}
279 \LWR@loadafter{hyperxmp}
280 \LWR@loadafter{hyphenat}
281 \LWR@loadafter{idxlayout}
282 \LWR@loadafter{ifoddpage}
283 \LWR@loadafter{imakeidx}
284 \LWR@loadafter{indentfirst}
285 \LWR@notmemoirloadafter{index}
286 % inputenc must be loaded before l warp
287 % inputenx must be loaded before l warp
288 \LWR@loadafter{intopdf}
289 \LWR@loadafter{keyfloat}
290 \LWR@loadafter{layaureo}
291 \LWR@loadafter{layout}
292 \LWR@loadafter{leading}
293 \LWR@loadafter{letterspace}
294 \LWR@loadafter{lettrine}
295 \LWR@loadafter{lineno}
296 \LWR@loadafter{lips}
297 \LWR@loadafter{listings}
298 \LWR@loadafter{longtable}
299 \LWR@loadafter{lscape}
300 \LWR@loadafter{ltablex}
301 \LWR@loadafter{ltcaption}
302 \LWR@loadafter{ltxgrid}
303 \LWR@loadafter{ltxtable}
304 \LWR@loadafter{lua-check-hyphen}
305 \LWR@loadafter{lua-visual-debug}
306 \LWR@loadafter{luacolor}
307 \LWR@loadafter{luatodonotes}
308 \LWR@loadafter{magaz}
309 \LWR@notmemoirloadafter{makeidx}
310 \LWR@loadafter{marginfit}
311 \LWR@loadafter{marginfix}
312 \LWR@loadafter{marginnote}
313 \LWR@loadafter{mcaption}
314 \LWR@loadafter{mdframed}
315 \LWR@loadafter{memhfixc}
316 \LWR@loadafter{metalogo}
317 \LWR@loadafter{mhchem}
318 \LWR@loadafter{microtype}
319 \LWR@loadafter{midfloat}
320 \LWR@loadafter{midpage}
321 \LWR@loadafter{morefloats}
322 \LWR@notmemoirloadafter{moreverb}
323 % morewrites must be loaded before l warp
324 \LWR@notmemoirloadafter{mparhack}
```

```
325 \%LWR@loadafter{multicol} loaded by ltxdoc
326 \LWR@loadafter{multicolrule}
327 \LWR@loadafter{multirow}
328 \LWR@loadafter{multitoc}
329 \LWR@loadafter{musicography}
330 \LWR@loadafter{nameref}
331 \LWR@loadafter{natbib}
332 \LWR@notmemoirloadafter{nccfancyhdr}
333 \LWR@notmemoirloadafter{needspace}
334 % newclude must be loaded before lwarf
335 \LWR@loadafter{newtxmath}
336 % newunicodechar must be loaded before lwarf
337 \LWR@notmemoirloadafter{nextpage}
338 \LWR@loadafter{nicefrac}
339 \LWR@loadafter{nonfloat}
340 \LWR@loadafter{nonumonpart}
341 \LWR@loadafter{nopageno}
342 \LWR@loadafter{notespages}
343 \LWR@loadafter{nowidow}
344 \LWR@loadafter{theorem}
345 \LWR@loadafter{octave}
346 \LWR@loadafter{overpic}
347 \LWR@loadafter{pagegrid}
348 \LWR@notmemoirloadafter{pagenote}
349 \LWR@loadafter{pagesel}
350 \LWR@loadafter{paralist}
351 \LWR@loadafter{parnotes}
352 \LWR@notmemoirloadafter{parskip}
353 \LWR@loadafter{pbox}
354 \LWR@loadafter{pdfcomment}
355 \LWR@loadafter{pdflscape}
356 \LWR@loadafter{pdfmarginpar}
357 \LWR@loadafter{pdfpages}
358 \LWR@loadafter{pdfprivacy}
359 \LWR@loadafter{pdfrender}
360 \LWR@loadafter{pdfsync}
361 \LWR@loadafter{pdftricks}
362 \LWR@loadafter{pdfx}
363 \LWR@loadafter{pfnote}
364 \LWR@loadafter{phfqit}
365 \LWR@loadafter{placeins}
366 \LWR@loadafter{plarray}
367 \LWR@loadafter{plarydshln}
368 \LWR@loadafter{plextarray}
369 \LWR@loadafter{plextarydshln}
370 \LWR@loadafter{plcolortbl}
371 \LWR@loadafter{plextdearray}
372 \LWR@loadafter{prelim2e}
373 \LWR@loadafter{prettyref}
374 \LWR@loadafter{preview}
```

```
375 \LWR@loadafter{psfrag}
376 \LWR@loadafter{psfragx}
377 \LWR@loadafter{pst-eps}
378 \LWR@loadafter{pstool}
379 \LWR@loadafter{pstricks}
380 % \LWR@loadafter{pxatbegshi} may be used by morewrites
381 \LWR@loadafter{pxeveryshi}
382 \LWR@loadafter{pxftnright}
383 \LWR@loadafter{pxjahyper}
384 \LWR@loadafter{quotchap}
385 \LWR@loadafter{quoting}
386 \LWR@loadafter{ragged2e}
387 \LWR@loadafter{realscripts}
388 \LWR@loadafter{refcheck}
389 \LWR@loadafter{register}
390 \LWR@loadafter{relsized}
391 \LWR@loadafter{repeatindex}
392 \LWR@loadafter{resizegather}
393 \LWR@loadafter{rmpage}
394 \LWR@loadafter{romanbar}
395 \LWR@loadafter{romanbarpagenumber}
396 \LWR@loadafter{rotating}
397 \LWR@loadafter{rotfloat}
398 \LWR@loadafter{rviewport}
399 \LWR@loadafter{savetrees}
400 % scalefnt is loaded by babel-french
401 \LWR@loadafter{schemata}
402 \LWR@loadafter{scrextrnd}
403 \LWR@loadafter{scrhack}
404 \LWR@loadafter{scrlayer}
405 \LWR@loadafter{scrlayer-notecolumn}
406 \LWR@loadafter{scrlayer-scrpage}
407 \LWR@loadafter{scrpge2}
408 \LWR@loadafter{section}
409 \LWR@loadafter{sectionbreak}
410 \LWR@loadafter{sectsty}
411 \LWR@loadafter{semantic-markup}
412 \LWR@notmemoirloadafter{setspace}
413 \LWR@loadafter{shadow}
414 \LWR@notmemoirloadafter{showidx}
415 \LWR@loadafter{showkeys}
416 \LWR@loadafter{sidecap}
417 \LWR@loadafter{sidenotes}
418 \LWR@loadafter{SIunits}
419 \LWR@loadafter{siunitx}
420 \LWR@loadafter{soul}
421 \LWR@loadafter{soulpos}
422 \LWR@loadafter{soulutf8}
423 \LWR@loadafter{splitidx}
424 \LWR@loadafter{srcltx}
```

```
425 \LWR@loadaftersrcTeX
426 \LWR@loadaftersstabular
427 \LWR@notltjloadaftersstfloats
428 \LWR@loadaftersubfig
429 \LWR@loadaftersubfigure
430 \LWR@loadaftersupertabular
431 \LWR@loadaftert1inc
432 \LWR@loadaftertabls
433 \LWR@notmemoirloadaftertabularx
434 \LWR@loadaftertabulary
435 \LWR@loadaftertascmac
436 \LWR@loadaftertextarea
437 % \LWR@loadaftertextcomp% maybe before lwarf with font packages
438 \LWR@loadaftertextfit
439 \LWR@loadaftertextpos
440 \LWR@loadaftertheorem
441 \LWR@loadafterthinsp
442 \LWR@loadafterthreadcol
443 \LWR@loadaftertreeparttable
444 \LWR@loadaftertthumb
445 \LWR@loadaftertthumbs
446 \LWR@loadaftertikz
447 \LWR@loadaftertitleps
448 \LWR@loadaftertitlesec
449 \LWR@loadaftertitletoc
450 \LWR@notmemoirloadaftertitling
451 % \LWR@loadaftertocbasic% preloaded by koma-script classes
452 \LWR@notmemoirloadaftertocbibind
453 \LWR@loadaftertoccenter
454 \LWR@notmemoirloadaftertocloft
455 \LWR@loadaftertocstyle
456 \LWR@loadaftertodo
457 \LWR@loadaftertodonotes
458 \LWR@loadaftertram
459 \LWR@loadaftertransparent
460 \LWR@loadaftertrimclip
461 \LWR@loadaftertrivfloat
462 \LWR@loadafterturnthepage
463 \LWR@loadaftertwoup

464 % \LWR@loadaftertypearea% preloaded by koma-script classes
465 % \LWR@loadaftertulem% preloaded by ctexart and related classes
466 \LWR@loadafterscore
467 \LWR@loadaftersunits
468 \LWR@loadaftersunitsdef
469 \LWR@loadafterspref
470 \LWR@loadaftersurl
471 \LWR@loadafterspace
472 \LWR@loadaftervarioref% no lwarf package provided
```

```

473 \LWR@notmemoirloadafter{verse}
474 \LWR@loadafter{versonotes}
475 \LWR@loadafter{vertbars}
476 \LWR@loadafter{vmargin}
477 \LWR@loadafter{vowel}
478 \LWR@loadafter{vpe}
479 \LWR@loadafter{vwcol}
480 \LWR@loadafter{wallpaper}
481 \LWR@loadafter{watermark}
482 \LWR@loadafter{widows-and-orphans}
483 \LWR@loadafter{wrapfig}
484 \LWR@loadafter{xbmks}
485 \LWR@loadafter{xcolor}
486 \LWR@loadafter{xchangebar}
487 \LWR@loadafter{xellipsis}
488 % xetexko-vertical must be loaded before l warp
489 \LWR@loadafter{xfrac}
490 \LWR@loadafter{xltabular}
491 \LWR@loadafter{xltextra}
492 \LWR@loadafter{xmpincl}
493 \LWR@loadafter{xpiano}
494 \LWR@loadafter{xpinyin}
495 \LWR@loadafter{xtab}
496 % xunicode must be loaded before l warp
497 \LWR@loadafter{xurl}
498 \LWR@loadafter{xy}
499 \LWR@loadafter{zwpagelayout}

```

22 MD5 hashing

The MD5 hash is used for `latextimage` filenames for SVG math.

```

500 \newcommand{\LWR@mdfive}[1]{%
501 \PackageError{l warp}%
502 {No MD5 macro was found.}%
503 {L warp must find the macros pdfmdfivesum or mdfivesum.}%
504 }

```

The default for `pdflATEX`, `DVILATEX`, `upLATEX`, etc:

```
505 \let\LWR@mdfive\pdfmdfivesum
```

For `LuaLATEX`:

```

506 \ifLuaTeX
507 \RequirePackage{pdftexcmds}
```

```
508 \let\LWR@mdfive\pdf@mdfivesum
509 \fi
```

For X_ELATEX:

```
510 \ifXeTeX
511 \@ifundefined{pdffivesum}{}{
512     \let\LWR@mdfive\pdfmdfivesum
513 \else
514     \let\LWR@mdfive\mdfivesum
515 \fi
```

23 pdfLATEX T1 and UTF-8 encoding

When using pdflATEX, l warp requires T1 encoding, and recommends UTF-8 encoding.

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

X_ELATEX and LuatATEX are both UTF-8 by nature.

\LWR@pdfencoding Sets T1, and also utf8 if not already set.

```
516 \newcommand*\LWR@pdfencoding{%
517     \RequirePackage[T1]{fontenc}
518
519     \@ifpackageloaded{inputenc}{}{
520         \@ifpackageloaded{inputenx}{}{
521             \RequirePackage[utf8]{inputenc}
522         }
523     }
524 }

525 \ifPDFTeX% pdflatex or dvi latex
526     \LWR@pdfencoding
527 \fi
528
529 \ifpTeX
530     \LWR@pdfencoding
531 \fi
```

24 Unicode input characters

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

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

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

```

532 \ifpTeX
533 \else
534 \RequirePackage{newunicodechar}
535
536 \newunicodechar{x}{\texttimes}
537
538 \ifPDFTeX% pdflatex or dvi latex
539 \newunicodechar{ff}{ff}% the first arguments are ligatures
540 \newunicodechar{fi}{fi}
541 \newunicodechar{fl}{fl}
542 \newunicodechar{ffi}{ffi}
543 \newunicodechar{ffl}{ffl}
544 \newunicodechar{--}{--}
545 \newunicodechar{-}{-}
546 \fi
547
548 \fi

```

25 Upright quotes

In PDFTEX, preserve upright quotes in verbatim text. `upquote` also loads `textcomp`.

```

549 \ifPDFTeX
550 \RequirePackage{upquote}
551 \fi
552
553 \ifpTeX
554     \RequirePackage{upquote}
555 \fi

```

26 Miscellaneous tools

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

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

```
556 \newcommand*\LWR@providelength[1]{%
557   \ifdefined\length{#1}{}{\newlength{#1}}%
558 }
```

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

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

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

\LWR@patcherror {*packagename*} {*macroname*}

Prints an error if could not patch a macro.

```
560 \newcommand*{\LWR@patcherror}[2]{%
561 \PackageError{l warp}%
562 {Unable to patch package #1, macro #2}%
563 {Please contact the author of the l warp package.}%
564 }
```

\LWR@isolate {*text*} Isolates Chinese characters from the surrounding text. This is required to avoid extra spaces on either side of the Chinese characters, especially when written to a file.

```
565 \newcommand{\LWR@isolate}[1]{#1}%
566
567 \@ifpackageloaded{ctexpatch}{%
568   \renewcommand{\LWR@isolate}[1]{\null#1\null}%
569 }{}%
570
571 \@ifpackageloaded{xeCJK}{%
572   \renewcommand{\LWR@isolate}[1]{\null#1\null}%
573 }{}}
```

27 Operating-System portability

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

27.1 Literal characters

Literal characters to be used in PrintLatexCmd and HTMLLatexCmd. These are defined without @ to easily allow their inclusion in the user’s document.

The literal % character:

```
574 \let\LWRpercent\@percentchar
```

The literal \$ character:

```
575 \catcode`\$=12
576 \def\LWRdollar{\$}
577 \catcode`\$=3
```

The literal & character:

```
578 \catcode`\&=12
579 \def\LWRamp{\&}
580 \catcode`\&=4
```

The literal \ character. The ampersand is temporarily set to the escape character during the definition of the backslash macro.

```
581 \catcode`\&=0
582 &\catcode`\&=12
583 &\def&\LWRbackslash{\}
584 &\catcode`\&=0
585 \catcode`\&=4
```

The literal # character:

```
586 \catcode`\#=12
```

```
587 \def\LWRhash{#}
588 \catcode`#=6
```

\LWRopquote The operating system's quote mark, UNIX default. For WINDOWS, see \LWR@setOSWindows, below.

```
589 \def\LWRopquote{`}
```

\LWRopseq The operating system's sequential execution command, UNIX default. For WINDOWS, see \LWR@setOSWindows, below.

```
590 \def\LWRopseq{\space\LWRamp\LWRamp\space\space}
```

27.2 Common portability code

Bool usingOSWindows Set if the OSWindows option is used, or if WINDOWS is automatically detected.

```
591 \newbool{usingOSWindows}
592 \boolfalse{usingOSWindows}
```

27.3 UNIX, LINUX, and MAC OS

\OSPathSymbol Symbol used to separate directories in a path.

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

27.4 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.

```
594 \newcommand*{\LWR@setOSWindows}
595 {
596 \booltrue{usingOSWindows}
597 \renewcommand*{\OSPathSymbol}{\@backslashchar}
598 \def\LWRopquote{"}
```

```
599 \def\LWRopseq{\space\WRamp\space\space}
600 }
```

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

```
601 \ifwindows
602 \LWR@setOSWindows
603 \fi
```

28 Package options

`Pkg kvoptions` Allows key/value package options.

```
604 \RequirePackage{kvoptions}
605 \SetupKeyvalOptions{family=LWR,prefix=LWR@}
```

`\l warpsetup` A user interface to set the keys:

```
606 \newcommand{\l warpsetup}[1]{\setkeys{LWR}{#1}}
```

`Bool warpingprint`

`Bool warpingHTML`

`Bool mathjax`

`Bool LWR@origmathjax`

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

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

```
607 \newbool{warpingprint}
608 \newbool{warpingHTML}
609 \newbool{mathjax}
610 \newbool{LWR@origmathjax}
```

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

```
611 \booltrue{warpingprint}%
612 \boolfalse{warpingHTML}%
613 \boolfalse{mathjax}%
```

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

```
614 \DeclareVoidOption{warpprint}{%
```

```

615 \PackageInfo{lwarp}{Using option 'warpprint'}
616 \booltrue{warpingprint}%
617 \boolfalse{warpingHTML}%
618 }

```

- Opt **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.

```

619 \DeclareVoidOption{warpHTML}{%
620 \PackageInfo{lwarp}{Using option 'warpHTML'}%
621 \booltrue{warpingHTML}%
622 \boolfalse{warpingprint}%
623 }

```

- 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.

```

624 \DeclareVoidOption{mathsvg}{%
625 \PackageInfo{lwarp}{Using option 'mathsvg'}%
626 \boolfalse{mathjax}%
627 \boolfalse{LWR@origmathjax}%
628 }

```

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

```

629 \DeclareVoidOption{mathjax}{%
630 \PackageInfo{lwarp}{Using option 'mathjax'}%
631 \booltrue{mathjax}%
632 \booltrue{LWR@origmathjax}%
633 }

```

- 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.

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

- Opt **makeindexStyle** Selects a custom `.ist` file. The default is `lwarp.ist`. A customized file should be based on `lwarp.ist`, and must retain the lines related to `\hyperindexref`.

635 \DeclareStringOption[lwarf.ist]{makeindexStyle}

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

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

636 \DeclareStringOption[lwarf.xdy]{xindyStyle}

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

637 \DeclareStringOption[english]{xindyLanguage}

Opt `xindyCodepage` Sets the `xindy` codepage to be assigned in `lwarpmk`'s configuration files. This is then used by `lwarpmk` while processing the index.

638 \DeclareStringOption[utf8]{xindyCodepage}

Opt `pdftotextEnc` The option `pdftotextEnc` sets the encoding used by `pdftotext`. This is passed to `pdftotext` using its `-enc` option, and is used when converting L^AT_EX PDF output with HTML tags into a plain-text file with HTML tags.
 Default: `UTF-8`

639 \DeclareStringOption[UTF-8]{pdftotextEnc}

Opt `lwarpmk` Tells `lwarf` 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`:

```
640 \newbool{LWR@creatinglwarpmk}
641 \boolfalse{LWR@creatinglwarpmk}
642
643 \DeclareVoidOption{lwarpmk}{%
644 \PackageInfo{lwarf}{Using option 'lwarpmk'}%
645 \booltrue{LWR@creatinglwarpmk}%
646 }
```

Opt `OSWindows` Tells `lwarf` 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.

647 \DeclareVoidOption{OSWindows}{}

```
648 \PackageInfo{lwarp}{Using option 'OSWindows'}
649 \LWR@setOSWindows
650 }
```

Opt `HomeHTMLFilename`
 Default: `\lwarp` 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.

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

Opt `HTMLFilename`
 Default: `<empty>` 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.

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

Opt `PrintLatexCmd` The shell commands to use to compile the print document.
 Default: `<automatic>`

```
653 \DeclareStringOption[] {PrintLatexCmd}
```

Opt `HTMLLatexCmd` The shell commands to use to compile the HTML document.
 Default: `<automatic>`

```
654 \DeclareStringOption[] {HTMLLatexCmd}
```

Opt `PrintIndexCmd` The shell commands to use to compile the print indexes.
 Default: `<empty>`

```
655 \DeclareStringOption[] {PrintIndexCmd}
```

Opt `HTMLIndexCmd` The shell commands to use to compile the HTML indexes.
 Default: `<empty>`

```
656 \DeclareStringOption[] {HTMLIndexCmd}
```

Opt `LatexmkIndexCmd` The shell commands to be used by *latexmk* to compile the print indexes. Unlike `PrintIndexCmd` and `HTMLIndexCmd`, `LatexmkIndexCmd` does not include the filename, which will be provided by *latexmk*.

```
657 \DeclareStringOption[] {LatexmkIndexCmd}
```

Opt `makeindex` Tells `lwarp` to use `makeindex` for index generation. When `lwarpmk.conf` and `*.lwarpmkconf` are generated, `PrintIndexCmd` and `HTMLIndexCmd` will be set for `makeindex` with a single index file.

```
658 \DeclareBoolOption[false] {makeindex}
```

Opt `xindy` Tells *lwarf* to use *xindy* for index generation. When `lwarfmk.conf` and `*.lwarfmkconf` are generated, `PrintIndexCmd` and `HTMLIndexCmd` will be set for *xindy* with a single index file.

```
659 \DeclareBoolOption[false]{xindy}
```

Opt `GlossaryCmd` The shell command to use to compile the glossary. The print or HTML version of the glossary filename will be appended to this command.
Default: `makeglossaries`

```
660 \DeclareStringOption[makeglossaries]{GlossaryCmd}
```

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

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

Opt `dvips` Option *dvips* tells *lwarfmk* to use *dvips* when compiling DVI *latex* documents.

```
662 \DeclareBoolOption[false]{dvips}
```

Opt `dvipdfm` Option *dvipdfm* tells *lwarfmk* to use *dvipdfm* when compiling DVI *latex* documents.

```
663 \DeclareBoolOption[false]{dvipdfm}
```

Opt `dvipdfmx` Option *dvipdfmx* tells *lwarfmk* to use *dvipdfmx* when compiling DVI *latex* documents.

```
664 \DeclareBoolOption[false]{dvipdfmx}
```

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

```
665 \ProcessKeyvalOptions*\relax
```

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

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

Defaults unless already over-ridden by the user:

```
667 \ifcsempty{\LWR@HomeHTMLFilename}{
668 \newcommand*\{\HomeHTMLFilename}{\BaseJobname}
669 }{
670 \csedef{\HomeHTMLFilename}{\LWR@HomeHTMLFilename}
671 }
672
673 \csedef{\HTMLFilename}{\LWR@HTMLFilename}
```

\LWR@PrintIndexCmd and \LWR@HTMLIndexCmd are tested to see if they are empty. If so, they are set to a reasonable defaults for a single index using *makeindex*, then possibly set to defaults for *xindy* if the lwarp xindy option was selected.

```
674 \ifdefempty{\LWR@PrintIndexCmd}{%
675     \renewcommand{\LWR@PrintIndexCmd}{%
676         makeindex -s \LWR@makeindexStyle \space \jobname.idx}%
677     }%
678     \ifbool{\LWR@xindy}{%
679         \renewcommand{\LWR@PrintIndexCmd}{%
680             xindy
681             -M \LWR@xindyStyle \space
682             -L \LWR@xindyLanguage \space
683             -C \LWR@xindyCodepage \space
684             \jobname.idx}%
685         }%
686     }{}%
687 }{}%
688
689 \ifdefempty{\LWR@HTMLIndexCmd}{%
690     \renewcommand{\LWR@HTMLIndexCmd}{%
691         makeindex -s \LWR@makeindexStyle \space \jobname_html.idx}%
692     }%
693     \ifbool{\LWR@xindy}{%
694         \renewcommand{\LWR@HTMLIndexCmd}{%
695             xindy
696             -M \LWR@xindyStyle \space
697             -L \LWR@xindyLanguage \space
698             -C \LWR@xindyCodepage \space
699             \jobname_html.idx}%
700         }%
701     }{}%
702 }{}%
703
704 \ifdefempty{\LWR@LatexmkIndexCmd}{%
705     \renewcommand{\LWR@LatexmkIndexCmd}{%
706         makeindex -s \LWR@makeindexStyle}%
707     }%
708     \ifbool{\LWR@xindy}{%
709         \renewcommand{\LWR@LatexmkIndexCmd}{%
710             xindy
711             -M \LWR@xindyStyle \space
712             -L \LWR@xindyLanguage \space
713             -C \LWR@xindyCodepage}%
714         }%
715     }{}%
716 }{}
```

28.1 Conditional compilation

\warpprintonly {*contents*}

Only process the contents if producing printed output.

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

\warpHTMLonly {*contents*}

Only process the contents if producing HTML output.

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

Pkg comment Provides conditional code blocks.

Attempts to use versions or verbatim fail in some cases, and do not provide much of a speed benefit even when they do work.

```
719 \RequirePackage{comment}
```

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

```
720 \ifbool{warpingHTML}{  
721 \def\DefaultCutFileName{\def\CommentCutFile{comment_html.cut}}}  
722 }{}  
723  
724 \ifbool{warpingprint}{  
725 \def\DefaultCutFileName{\def\CommentCutFile{comment_print.cut}}}  
726 }{}
```

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

```
727 \includecomment{warpall}
```

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

Env warpHTML For HTML output:

```
728 \ifbool{warpingHTML}{  
729 {\includecomment{warpHTML}}}  
730 {\excludecomment{warpHTML}}%
```

```
731 \ifbool{warpingprint}{
```

```
732 {\includecomment{warpprint}}
733 {\excludecomment{warpprint}}
```

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

```
734 \ifbool{LWR@creatingswarp}{}
735 {\includecomment{LWR@createlswarp}}
736 {\excludecomment{LWR@createlswarp}}
```

29 Required packages

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

for HTML output: 737 \begin{warpHTML}

Load fontspec if necessary:

```
738 \ifxetexorluatex
739 @ifpackageloaded{fontspec}{}{
740     \usepackage[no-math]{fontspec}
741 }
```

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

```
742 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
743 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
744 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
745 \else
```

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

Pkg **microtype**

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

```
746 \RequirePackage{microtype}
747
748 \microtypesetup{
749     protrusion=false,
750     expansion=false,
```

```

751     tracking=false,
752     kerning=false,
753     spacing=false}
754
755 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}

756 \fi

757 \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: 758 `\begin{warpHTML}`

```
759 \RequirePackage{geometry}
```

Avoid class and option conflict by changing settings after package load:

```

760 \geometry{
761     paperheight=190in,%
762     paperwidth=20in,%
763     left=2in,right=6in,%
764     top=1in,bottom=1in,%
765 }

```

`\tjarticle` and other classes may require these to be reset by `lwarp`:

```

766 \setlength{\textheight}{0.8\paperheight}
767 \setlength{\textwidth}{0.7\paperwidth}
768
769 \twosidefalse
770 \mparswitchfalse
771
772 \end{warpHTML}

```

for HTML & PRINT: 773 `\begin{warpall}`

Pkg **xparse**

$\text{\LaTeX}3$ command argument parsing

774 \RequirePackage{xparse}

Pkg **calc**

 \RequirePackage{calc}

776 \end{warpall}

for HTML output: 777 \begin{warpHTML}

Pkg **expl3**

$\text{\LaTeX}3$ programming

778 \RequirePackage{expl3}

Pkg **getttitlestring**

 Used to emulate \nameref.

779 \RequirePackage{getttitlestring}

Pkg **everyhook**

 everyhook is used to patch paragraph handling.

780 \@ifundefined{bxjs@everypar}{}{\let\everypar\bxjs@everypar}
781
782 \RequirePackage{everyhook}
783 \end{warpHTML}

for HTML & PRINT: 784 \begin{warpall}

Pkg **filecontents**

 Used to write helper files, done in print mode.

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

785 \RequirePackage{filecontents}
786

```
787 \@ifpackagelater{filecontents}{2011/10/09}%
788 {}
789 {
790 \newwrite\fcwrite
791 \let\LWR@origfilec@ntents\filec@ntents
792 \def\filec@ntents{\def\chardef##1{\write{\let\reserved@c\fcwrite}\LWR@origfilec@ntents}}
793 }

794 \end{warpall}
```

for HTML output: 795 \begin{warpHTML}

Pkg xifthen

```
796 \RequirePackage{xifthen}
```

Pkg verbatim

```
797 \RequirePackage{verbatim}
```

Pkg refcount

Provides \setcounterref, \setcounterpageref, etc.

```
798 \RequirePackage{refcount}
```

Pkg newfloat

```
799 \RequirePackage{newfloat}
```

```
800 \end{warpHTML}
```

for HTML & PRINT: 801 \begin{warpall}

Pkg xstring

```
802 \RequirePackage{xstring}
```

Pkg environ Used to encapsulate math environments for re-use in HTML <alt> text.

```
803 \RequirePackage{environ}
```

```
804 \end{warpall}
```

for HTML output: 805 \begin{warpHTML}

Pkg **zref** Used for cross-references.

```
806 \RequirePackage{zref}
```

Pkg **printlen** Used to convert lengths for image width/height options.

```
807 \RequirePackage{printlen}
```

\LWR@printlength {<length>}

Prints a length using a locally-controlled unit and space. Rounding is used unless the length is small.

```
808 \newrobustcmd*\LWR@printlength[1]{%
809 \begingroup%
810 \uselengthunit{PT}%
811 \renewcommand*{\unitspace}{ }%
812 \ifdimless{#1}{10pt}{%
813   \printlength{#1}%
814 }{%
815   \rndprintlength{#1}%
816 }%
817 \endgroup%
818 }
```

```
819 \end{warpHTML}
```

for PRINT output: 820 \begin{warpprint}

Pkg **varwidth** Used for print-mode `lateximage`:

```
821 \RequirePackage{varwidth}
```

```
822 \end{warpprint}
```

30 Loading packages

for HTML output: 823 \begin{warpHTML}

Remember the original \RequirePackage:

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

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

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

\LWR@parsedrequirepackagenames Stores the parsed list of required package names after spaces are removed and lwarp- is prepended.

```
826 \newcommand*{\LWR@parsedrequirepackagenames}{}{}
```

\LWR@checkloadfilename {<filename>} Checks if this filename should be loaded after lwarp, or never at all.

The following should never be loaded:

```
827 \newcommand*{\LWR@checkloadfilename}[1]{%
828 \LWR@checkloadnever{\#1}{ae}{latinmodern}%
829 \LWR@checkloadnever{\#1}{aecc}{latinmodern}%
830 \LWR@checkloadnever{\#1}{bitfield}{bytefield}%
831 \LWR@checkloadnever{\#1}{boxedminipage}{boxedminipage2e}%
832 \LWR@checkloadnever{\#1}{caption2}{caption}%
833 \LWR@checkloadnever{\#1}{ccaption}{caption}%
834 \LWR@checkloadnever{\#1}{doublespace}{setspace}%
835 \LWR@checkloadnever{\#1}{fancyheadings}{fancyhdr}%
836 \LWR@checkloadnever{\#1}{glossary}{glossaries}%
837 \LWR@checkloadnever{\#1}{newthm}{ntheorem}%
838 \LWR@checkloadnever{\#1}{rplain}{fancyhdr}%
839 \LWR@checkloadnever{\#1}{si}{siunitx}%
840 \LWR@checkloadnever{\#1}{t1enc}{fontenc, inputenc, inputenx}%
841 \LWR@checkloadnever{\#1}{wasysym}{textcomp, amssymb, amsfonts, mnsymbol, fdsymbol}}
```

The following should only be loaded before lwarp:

```
842 \LWR@checkloadbefore{\#1}{ctex}%
843 \LWR@checkloadbefore{\#1}{fontspec}%
844 \LWR@checkloadbefore{\#1}{inputenc}%
845 \LWR@checkloadbefore{\#1}{inputenx}%
846 \LWR@checkloadbefore{\#1}{kotex}%
847 \LWR@checkloadbefore{\#1}{luatexja}%
848 \LWR@checkloadbefore{\#1}{luatexja-fontspec}%
849 \LWR@checkloadbefore{\#1}{luatexko}%
850 \LWR@checkloadbefore{\#1}{morewrites}%
851 \LWR@checkloadbefore{\#1}{newclclude}%
852 \LWR@checkloadbefore{\#1}{newunicodechar}%
853 \LWR@checkloadbefore{\#1}{plex}%
854 \LWR@checkloadbefore{\#1}{xeCJK}%
855 \LWR@checkloadbefore{\#1}{xetexko}%
856 \LWR@checkloadbefore{\#1}{zxjatype}%
857 }
```

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

Note that argument 4 is passed directly to \StrBetween.

```
858 \newcommand*\LWR@findword[3][,]{%
859     \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
860 }
```

```
\LWR@lookforpackagename {<index>}
```

If this is an lwarp-supported package name, re-direct it to the lwarp version by renaming it `lwarp-` followed by the original name.

Looks `index` deep into the list of package names, `\LWR@requirepackagenames`, and builds `\LWR@parsedrequirepackagenames` which is the modified list of names.

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

Find the `index`'th package name from the list:

```
862 \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`.

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

See if the package name was found:

```
864 \IfStrEq{\LWR@strresulttwo}{}%
865 {}% no filename
866 {}% yes filename was found
```

Check if the package should be loaded before lwarp, or never at all:

```
867 \LWR@checkloadfilename{\LWR@strresulttwo}%
```

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

```
868 \IfExists{lwarp-\LWR@strresulttwo.sty}%
869 {}% lwarp-* file found
870 \ifdefvoid{\LWR@parsedrequirepackagenames}{%
871     \edef\LWR@parsedrequirepackagenames{lwarp-\LWR@strresulttwo}%
872 }{}%
873 \edef\LWR@parsedrequirepackagenames{%
874     \LWR@parsedrequirepackagenames,lwarp-\LWR@strresulttwo%
875 }%
```

```

876      }%
877      }%
878      {%

879      \ifdefvoid{\LWR@parsedrequirepackagenames}{%
880          \edef\LWR@parsedrequirepackagenames{\LWR@strresulttwo}%
881      }{%
882          \edef\LWR@parsedrequirepackagenames{%
883              \LWR@parsedrequirepackagenames,\LWR@strresulttwo}%
884      }%
885      }%
886      }% no lwarf-* file
887 }% yes filename
888 }

```

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

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

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

Redirect up to twenty names:¹⁶

```

890 \renewcommand*{\LWR@requirepackagenames}{#2}%
891 \renewcommand*{\LWR@parsedrequirepackagenames}{}%
892 \LWR@lookforpackagename{1}%
893 \LWR@lookforpackagename{2}%
894 \LWR@lookforpackagename{3}%
895 \LWR@lookforpackagename{4}%
896 \LWR@lookforpackagename{5}%
897 \LWR@lookforpackagename{6}%
898 \LWR@lookforpackagename{7}%
899 \LWR@lookforpackagename{8}%
900 \LWR@lookforpackagename{9}%
901 \LWR@lookforpackagename{10}%
902 \LWR@lookforpackagename{11}%
903 \LWR@lookforpackagename{12}%
904 \LWR@lookforpackagename{13}%
905 \LWR@lookforpackagename{14}%
906 \LWR@lookforpackagename{15}%
907 \LWR@lookforpackagename{16}%
908 \LWR@lookforpackagename{17}%
909 \LWR@lookforpackagename{18}%
910 \LWR@lookforpackagename{19}%
911 \LWR@lookforpackagename{20}%

```

¹⁶This was originally nine names, but then I came across a package which used twelve...

\RequirePackage depending on the options and version:

```

912 \IfValueTF{#1}%
913 {%
914   \IfValueTF{#3}%
915   { \LWR@origRequirePackage[#1]{\LWR@parsedrequirepackagenames}[#3] }%
916   { \LWR@origRequirePackage[#1]{\LWR@parsedrequirepackagenames} }%
917 }%
918 {%
919   \IfValueTF{#3}%
920   { \LWR@origRequirePackage{\LWR@parsedrequirepackagenames}[#3] }%
921   { \LWR@origRequirePackage{\LWR@parsedrequirepackagenames} }%
922 }%
923 }
924 \LetLtxMacro\usepackage\RequirePackage

```

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

Uses the original package, including options.

```

925 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{%
926   \PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
927   \IfValueTF{#2}{%
928     {\ProvidesPackage{lwarp-#1}[#2]}%
929     {\ProvidesPackage{lwarp-#1}}%
930     \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}%
931     \ProcessOptions\relax
932   \IfValueTF{#2}{%
933     {\LWR@origRequirePackage[#1][#2]}%
934     {\LWR@origRequirePackage[#1]}%
935   \DeclareOption*{}%
936   \ProcessOptions\relax
937 }

```

In some cases, the following seems to be required to avoid an “unknown option” error, such as when loading xcolor with options.

```

935 \DeclareOption*{}%
936 \ProcessOptions\relax
937 }

```

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

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

```

938 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{%
939   \PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
940   \IfValueTF{#2}{%
941     {\ProvidesPackage{lwarp-#1}[#2]}%
942     {\ProvidesPackage{lwarp-#1}}%
943   }

```

Ignore all options.

```
943 \DeclareOption*{}
```

Nullifies then processes the options. Seems to be required when options contain curly braces, which were causing “Missing `\begin{document}`”.

```
944 % \ProcessOptions\relax% original LaTeX code
945 \let\ds@\emptyset%
946 \edef\@curroptions{}% lwarf modification to \ProcessOptions
947 \@process@options\relax% from the original \ProcessOptions
948 }
```

```
949 \end{warpHTML}
```

31 Additional required packages

for HTML output: 950 `\begin{warpHTML}`

Pkg caption

```
951 \RequirePackage{caption}%
952 \end{warpHTML}
```

32 File handles

Defines file handles for writes.

for HTML & PRINT: 953 `\begin{warpall}`

`\LWR@quickfile` For quick temporary use only. This is reused in several places.

```
954 \newwrite\LWR@quickfile%
```

```
955 \end{warpall}
```

for HTML output: 956 `\begin{warpHTML}`

`\LWR@lateximagesfile` For `lateximages.txt`.

```
957 \newwrite\LWR@lateximagesfile
```

```
958 \end{warpHTML}
```

33 Include a file

During HTML output, `\include{<filename>}` causes the following to occur:

1. l warp creates `<filename>_html_inc.tex` whose contents are:

```
\input <filename>.tex
```

2. `<filename>_html_inc.tex` is then `\included` instead of `<filename>.tex`.

3. `<filename>_html_inc.aux` is automatically generated and used by L^AT_EX.

for HTML output: 959 `\begin{warpHTML}`

```
\include {<filename>}
```

`\@include {<filename>}` Modified to load _html_inc files.

```
960 \def\@include#1 {%
961   \immediate\openout\LWR@quickfile #1_html_inc.tex% l warp
962   \immediate\write\LWR@quickfile{\string\input{#1.tex}}% l warp
963   \immediate\closeout\LWR@quickfile% l warp
964   \LWR@origclearpage% \changed
965   \if@files
966     \immediate\write\@mainaux{\string\@input{#1_html_inc.aux}}% changed
967   \fi
968   \tempswattrue
969   \if@partsw
970     \tempswafalse
971     \edef\reserved@b{#1}%
972     \for\reserved@a:=\partlist\do
973       {\ifx\reserved@a\reserved@b\tempswattrue\fi}%
974   \fi
975   \if@tempswa
976     \let\@auxout\@partaux
977     \if@files
978       \immediate\openout\@partaux #1_html_inc.aux % changed
979       \immediate\write\@partaux{\relax}%
980     \fi
981     \input{#1_html_inc.tex}% changed
982     \LWR@origclearpage% changed
983     \writeckpt{#1}%
984 }
```

```

984     \if@filesw
985         \immediate\closeout\partaux
986     \fi
987 \else
988     \deadcycles{z@}
989     \cnameuse{cp@#1}%
990 \fi
991 \let\auxout\mainaux%
992 }

993 \end{warpHTML}

```

34 Copying a file

for HTML output: 994 \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.

```

995 \newwrite\LWR@copyoutfile % open the file to write to
996 \newread\LWR@copyinfile % open the file to read from
997
998 \newcommand*\LWR@copyfile[2]{%
999 \LWR@traceinfo{\LWR@copyfile: copying #1 to #2}
1000
1001 \immediate\openout\LWR@copyoutfile=#2
1002 \openin\LWR@copyinfile=#1
1003 \begingroup\endlinechar=-1
1004 \makeatletter
1005
1006 \LWR@traceinfo{\LWR@copyfile: about to loop}
1007
1008 \loop\unless\ifeof\LWR@copyinfile
1009   \LWR@traceinfo{\LWR@copyfile: one line}
1010   \read\LWR@copyinfile to\LWR@fileline % Read one line and store it into \LWR@fileline
1011 %   \LWR@fileline\par % print the content into the pdf
1012 % print the content:
1013   \immediate\write\LWR@copyoutfile{\unexpanded\expandafter{\LWR@fileline}}%
1014 \repeat
1015 \immediate\closeout\LWR@copyoutfile
1016 \LWR@traceinfo{\LWR@copyfile: done}
1017 \endgroup
1018 }

```

1019 \end{warpHTML}

35 Debugging messages

HTML comments To have the HTML output include additional HTML comments, such as which `<div>` is closing, use

`\booltrue{HTMLDebugComments}`

debugging information To have debug information written to the log, use

`\tracinglwarp`

for HTML & PRINT: 1020 `\begin{warpall}`

Bool `LWR@tracinglwarp` True if tracing is turned on.

1021 `\newbool{LWR@tracinglwarp}`

`\tracinglwarp` Turns on the debug tracing messages.

1022 `\newcommand{\tracinglwarp}{\booltrue{LWR@tracinglwarp}}`

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

```
1023 \newcommand{\LWR@traceinfo}[1]{%
1024 \ifbool{LWR@tracinglwarp}{%
1025 {%
1026     \typeout{*** lwarp: #1}%
1027 }%
1028 {}}%
1029 }
```

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

Default: `false`

1030 `\newbool{HTMLDebugComments}`
1031 `\boolfalse{HTMLDebugComments}`

If `\tracinglwarp`, show where preamble hooks occur:

```
1032 \AfterEndPreamble{%
1033 \LWR@traceinfo{AfterEndPreamble}%
1034 }
```

```
1035
1036 \AtBeginDocument{
1037 \LWR@traceinfo{AtBeginDocument}
1038 }

1039 \end{warpall}
```

36 Defining print and HTML versions of macros and environments

The following refers to defining objects inside l warp, and is not for the user's document.

Many macros and environments must be provided as both print and HTML versions.

While generating the print version of a document, the original macros as defined by L^AT_EX and its packages are used as-is.

While generating the HTML version of a document, the original macro or environment is redefined to call a new HTML version or a copy of the original print version. The new HTML versions of macros and environments are used most of the time. Copies of the print versions are used inside a `lateximage` environment, which draws and remembers an image of the printed output, and also several other places. The copies of the print versions may also be used by the HTML versions, such as when the HTML version merely encloses the print version inside HTML tags.

The general structure for providing print and HTML versions of a macro or environment is as follows:

For a preexisting macro, not defined with xparse: An HTML version is provided with a special name, inside a `warpHTML` environment, then `\LWR@formatted` is used to redefine and patch various macros:

```
\begin{warpHTML}
\newcommand{\LWR@HTML@name}{...}%
% may also use xparse

\LWR@formatted{name}
\end{warpHTML}
```

`\LWR@formatted{name}` copies the original print version, then redefines `\name` to use either the print or HTML version depending on which mode l warp is using. xparse may be used to define the new HTML version, even if the original did not use xparse.

For a preexisting environment, not defined with xparse: The process is similar.
Note the use of `\LWR@formattedenv` instead of `\LWR@formatted`.

```
\begin{warpHTML}
\newenvironment{\LWR@HTML@name}{...}%
may also use xparse

\LWR@formattedenv{name}
\end{warpHTML}
```

If the original used xparse: A copy must be made using a new name:

```
\begin{warpHTML}
\NewDocumentCommand{\LWR@print@name}{...}{...}%
copy the original

\NewDocumentCommand{\LWR@HTML@name}{...}{...}%
or use \newcommand

\LWR@formatted{name}
\end{warpHTML}
```

Similar for an environment, using `\LWR@formattedenv`. (`\LWR@formatted` and `\LWR@formattedenv` use `\LetLtxMacro` to copy the original print definition, which may not work with macros and environments created by xparse, so the print version must be manually recreated in the lwarp source.)

For a new macro or environment, not using xparse for the print version:

```
\begin{warpall}
\newcommand{\name}{...}%
NOT xparse!

\begin{warpHTML}
\newcommand{\LWR@HTML@name}{...}%
may use xparse for HTML

\LWR@formatted{name}
\end{warpHTML}
```

Similar for an environment. The plain `\name` or environment `name` is used for the printed version, and is placed inside `warpall`. xparse may be used for the `\LWR@HTML@<name>` version.

For a new macro or environment, using xparse: It is possible to use xparse for an entirely new macro or environment by defining the `\LWR@print@<name>` version with xparse, along with `\name` defined without xparse to refer directly to the `\LWR@print` version:

```
\begin{warpall}
\NewDocumentCommand{\LWR@print@name}{...} {...}%
\NewDocumentEnvironment{\LWR@print@name}{...} {...} %

% Simply a call to \LWR@print@name:
\newcommand{\name}{\LWR@print@name}%
\newenvironment{name}{\LWR@print@name}{\end\LWR@print@name}
\end{warpall}

\begin{warpHTML}
\NewDocumentCommand{\LWR@HTML@name}{...} {...}%
\NewDocumentEnvironment{\LWR@HTML@name}{...} {...} %

\LWR@formatted{name}%
\LWR@formattedenv{name}
\end{warpHTML}
```

In general, `\LWR@formatted` or `\LWR@formattedenv` are placed inside a `warpHTML` environment, and while producing an `HTML` document they do the following:

- Macros are modified:
 1. The pre-existing print version `\name` is saved as `\LWR@print@<name>`, unless `\LWR@print@<name>` is already defined.
 2. The original `\name` is redefined to call either the print or `HTML` version depending on which format is in use at the moment, as set by `\LWR@formatting`, which is defined as either “print” or “HTML”.
- When `lwarp` is producing a print document, the original definitions are used, as well as any new definitions defined in `warpall` above.
- When `lwarp` is generating `HTML` output, `\LWR@formatting` is set to “HTML”, and `\name` is directed to `\LWR@HTML@<name>`.
- When `lwarp` is generating `HTML` output but enters a `lateximage` environment, or for some other reason needs to draw images using the original print definitions, `\LWR@formatting` is changed to “print” and `\name` is then redirected to `\LWR@print@<name>`, which was the original `\name`.

Since arguments are not handled by the new `\name`, any star and other arguments are processed by the print or `HTML` version.

Expandable versions are also provided as well. These usually are necessary for anything which could appear inside a `tabular`, without which a “Misplaced `\omit`” error may occur.

 Misplaced `\omit` error

```
\LWR@expandableformatted
\LWR@expandableformattedenv
```

(Older versions of lwarp used \LetLtxMacro for everything, but this could fail when using macros defined by xparse. This older system is still in use for many definitions.)

for HTML output: 1040 \begin{warpHTML}

\LWR@formatting Remembers if selected print/HTML formatting.

Used while \LWR@restoreorigformatting, such as in an `lateximage`. May be set to either “print” or “HTML”.

```
1041 \newcommand*{\LWR@formatting}{HTML}
```

\LWR@formatted {*macroname*} No backslash in the macro name.

If not yet defined, defines \LWR@print@<name> as the original print-mode \<name>. Also redefines \<name> to use \LWR@<format>@<name>, where <format> is set by \LWR@formatting, and is print or HTML.

```
1042 \newcommand*{\LWR@formatted}[1]{%
1043     \ifcsundef{\LWR@print@#1}{%
1044         \expandafter\LetLtxMacro\csname LWR@print@#1\expandafter\endcsname%
1045         \csname#1\endcsname%
1046     }{%
1047     \ifcsundef{#1}{%
1048         \expandafter\newrobustcmd\csname #1\endcsname{%
1049             @nameuse{\LWR@LWR@formatting @#1}%
1050         }%
1051     }{%
1052         \expandafter\renewrobustcmd\csname #1\endcsname{%
1053             @nameuse{\LWR@LWR@formatting @#1}%
1054         }%
1055     }%
1056 }
```

\LWR@expandableformatted {*macroname*} No backslash in the macro name.

An expandable version of \LWR@formatted.

```
1057 \newcommand*{\LWR@expandableformatted}[1]{%
1058     \ifcsundef{\LWR@print@#1}{%
1059         \expandafter\LetLtxMacro\csname LWR@print@#1\expandafter\endcsname%
1060         \csname#1\endcsname%
1061     }{%
1062     \ifcsundef{#1}{%
1063         \expandafter\newcommand\csname #1\endcsname{%
```

```

1064           \nameuse{LWR@LWR@formatting @#1}%
1065       }%
1066   }{%
1067     \expandafter\renewcommand\csname #1\endcsname{%
1068       \nameuse{LWR@LWR@formatting @#1}%
1069     }%
1070   }%
1071 }

```

\LWR@formattedenv {*<environmentname>*}

If not yet defined, defines the environment LWR@print@<name> as the original print-mode <name>. Also redefines the environment <name> to use environment LWR@<format>@<name>, where <format> is set by \LWR@formatting, and is print or HTML.

```

1072 \newcommand*\LWR@formattedenv[1]{%
1073   \ifcsundef{LWR@print@#1}{%
1074     \expandafter\LetLtxMacro\csname LWR@print@#1\expandafter\endcsname{%
1075       \csname#1\endcsname}%
1076     \csletcs{endLWR@print@#1}{end#1}%
1077   }{%
1078     \DeclareDocumentEnvironment{#1}{}{%
1079       \nameuse{LWR@LWR@formatting @#1}%
1080       \nameuse{endLWR@LWR@formatting @#1}%
1081     }

```

\LWR@expandableformattedenv {*<environmentname>*}

An expandable version of LWR@formattedenv.

```

1082 \newcommand*\LWR@expandableformattedenv[1]{%
1083   \ifcsundef{LWR@print@#1}{%
1084     \expandafter\LetLtxMacro\csname LWR@print@#1\expandafter\endcsname{%
1085       \csname#1\endcsname}%
1086     \csletcs{endLWR@print@#1}{end#1}%
1087   }{%
1088     \DeclareExpandableDocumentEnvironment{#1}{}{%
1089       \nameuse{LWR@LWR@formatting @#1}%
1090       \nameuse{endLWR@LWR@formatting @#1}%
1091     }

```

1092 \end{warpHTML}

37 HTML-conversion output modifications

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

for HTML & PRINT: 1093 \begin{warpall}

37.1 User-level controls

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

1094 \newbool{FormatEPUB}
1095 \boolfalse{FormatEPUB}

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

1096 \newbool{FormatWP}
1097 \boolfalse{FormatWP}

Bool WPMarkFloats Adds
Default: false

```
==== begin table ====
...
==== end ====
or
==== begin figure ====
...
==== end ====

```

around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames and captions.¹⁷

1098 \newbool{WPMarkFloats}
1099 \boolfalse{WPMarkFloats}

Bool WPMarkMinipages Adds
Default: false

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

```

¹⁷Perhaps some day word processors will have HTML import options for identifying <figure> and caption tags for figures and tables.

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

```
1100 \newbool{WPMarkMinipages}
1101 \boolfalse{WPMarkMinipages}
```

Bool **WPMarkTOC** While formatting for word processors, adds
Default: true
`==== table of contents ===`

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

If set false, the actual TOC is printed instead.

```
1102 \newbool{WPMarkTOC}
1103 \booltrue{WPMarkTOC}
```

Bool **WPMarkLOFT** While formatting for word processors, adds
Default: false
`==== list of figures === and/or`
`==== list of tables ===`

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

If set false, the actual lists are printed instead.

```
1104 \newbool{WPMarkLOFT}
1105 \boolfalse{WPMarkLOFT}
```

Bool **WPMarkMath** While formatting for word processors, prints math as L^AT_EX code instead of creating
Default: false SVG images or MATHJAX. This is useful for cut/paste into the *LibreOffice Writer TeXMaths* extension.

```
1106 \newbool{WPMarkMath}
1107 \boolfalse{WPMarkMath}
```

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

See table 8 on page 190.

```
1108 \newbool{WPTitleHeading}
1109 \boolfalse{WPTitleHeading}
```

```
1110 \end{warpall}
```

37.2 Heading adjustments

If formatting the HTML for a word processor, adjust heading levels.

If `WPTitleHeading` is true, adjust so that part is **Heading 1**.

If `WPTitleHeading` is false, use `<h1>` for the title, and set part to **Heading 2**.

for HTML output: 1111 \begin{warpHTML}

```
1112 \AtBeginDocument{
1113 \ifbool{FormatWP}{
1114 \@ifundefined{chapter}{
1115 \ifbool{WPTitleHeading}{% part and section starting at h2
1116 \renewcommand*\{\LWR@tagtitle\}{h1}
1117 \renewcommand*\{\LWR@tagtitleend\}{/h1}
1118 \renewcommand*\{\LWR@tagpart\}{h2}
1119 \renewcommand*\{\LWR@tagpartend\}{/h2}
1120 \renewcommand*\{\LWR@tagsection\}{h3}
1121 \renewcommand*\{\LWR@tagsectionend\}{/h3}
1122 \renewcommand*\{\LWR@tagsubsection\}{h4}
1123 \renewcommand*\{\LWR@tagsubsectionend\}{/h4}
1124 \renewcommand*\{\LWR@tagsubsubsection\}{h5}
1125 \renewcommand*\{\LWR@tagsubsubsectionend\}{/h5}
1126 \renewcommand*\{\LWR@tagparagraph\}{h6}
1127 \renewcommand*\{\LWR@tagparagraphend\}{/h6}
1128 \renewcommand*\{\LWR@tagsubparagraph\}{span class="subparagraph"}
1129 \renewcommand*\{\LWR@tagsubparagraphend\}{/span}
1130 }% WPTitleHeading
1131 {% not WPTitleHeading, part and section starting at h1
1132 \renewcommand*\{\LWR@tagtitle\}{div class="title"}
1133 \renewcommand*\{\LWR@tagtitleend\}{/div}
1134 \renewcommand*\{\LWR@tagpart\}{h1}
1135 \renewcommand*\{\LWR@tagpartend\}{/h1}
1136 \renewcommand*\{\LWR@tagsection\}{h2}
1137 \renewcommand*\{\LWR@tagsectionend\}{/h2}
1138 \renewcommand*\{\LWR@tagsubsection\}{h3}
1139 \renewcommand*\{\LWR@tagsubsectionend\}{/h3}
1140 \renewcommand*\{\LWR@tagsubsubsection\}{h4}
1141 \renewcommand*\{\LWR@tagsubsubsectionend\}{/h4}
1142 \renewcommand*\{\LWR@tagparagraph\}{h5}
1143 \renewcommand*\{\LWR@tagparagraphend\}{/h5}
1144 \renewcommand*\{\LWR@tagsubparagraph\}{h6}
1145 \renewcommand*\{\LWR@tagsubparagraphend\}{/h6}
1146 }% not WPTitleHeading
1147 }% chapter undefined
```

```

1148 {%
1149 \ifbool{WPTitleHeading}{}
1150 {%
1151 \renewcommand*{\LWR@tagtitle}{\div class="title"}}
1152 \renewcommand*{\LWR@tagtitleend}{/div}
1153 \renewcommand*{\LWR@tagpart}{h1}
1154 \renewcommand*{\LWR@tagpartend}{/h1}
1155 \renewcommand*{\LWR@tagchapter}{h2}
1156 \renewcommand*{\LWR@tagchapterend}{/h2}
1157 \renewcommand*{\LWR@tagsection}{h3}
1158 \renewcommand*{\LWR@tagsectionend}{/h3}
1159 \renewcommand*{\LWR@tagsubsection}{h4}
1160 \renewcommand*{\LWR@tagsubsectionend}{/h4}
1161 \renewcommand*{\LWR@tagsubsubsection}{h5}
1162 \renewcommand*{\LWR@tagsubsubsectionend}{/h5}
1163 \renewcommand*{\LWR@tagparagraph}{h6}
1164 \renewcommand*{\LWR@tagparagraphend}{/h6}
1165 \renewcommand*{\LWR@tagsubparagraph}{span class="subparagraph"}}
1166 \renewcommand*{\LWR@tagsubparagraphend}{/span}
1167 }%
1168 }%
1169 }%
1170 }%
1171 \end{warpHTML}

```

38 Remembering original formatting macros

for HTML output: 1172 \begin{warpHTML}

Remember original definitions of formatting commands. Will be changed to HTML commands for most uses. Will be temporarily restored to original meaning inside any `lateximage` environment. Also nullify unused commands.

Some packages redefine `\#`, which is used to generate HTML, so the original must be remembered here.

```

1173 \chardef\LWR@origpound='\#
1174 \let\LWR@origcomma\
1175 \let\LWR@origthinspace\thinspace
1176 \let\LWR@orignegthinspace\negthinspace
1177 \let\LWR@origtilde~
1178 \let\LWR@origenskip\enskip
1179 \let\LWR@origquad\quad
1180 \let\LWR@origqquad\qquad

```

```
1181 \let\LWR@orighfil\hfil
1182 \let\LWR@orighss\hss
1183 \let\LWR@origllap\llap
1184 \let\LWR@origrlap\rlap
1185 \let\LWR@origfilneg\hfilneg
1186
1187 \let\LWR@origmedskip\medskip
1188 \let\LWR@origbigskip\bigskip
1189
1190 \let\LWR@origtextellipsis\textellipsis
1191
1192 \LetLtxMacro{\LWR@origrmfamily\rmfamily}
1193 \LetLtxMacro{\LWR@origsffamily\sffamily}
1194 \LetLtxMacro{\LWR@origttfamily\ttfamily}
1195 \LetLtxMacro{\LWR@origbfseries\bfseries}
1196 \LetLtxMacro{\LWR@origmdseries\mdseries}
1197 \LetLtxMacro{\LWR@origupshape\upshape}
1198 \LetLtxMacro{\LWR@origslshape\slshape}
1199 \LetLtxMacro{\LWR@origscshape\scshape}

1200 \@ifundefined{sishape}{
1201     \LetLtxMacro{\LWR@origsishape\scshape}
1202 }{
1203     \LetLtxMacro{\LWR@origsishape\sishape}
1204 }

1205 \LetLtxMacro{\LWR@origitshape\itshape}
1206 \LetLtxMacro{\LWR@origem\em}
1207 \LetLtxMacro{\LWR@orignormalfont\normalfont}
1208
1209 \let\LWR@origonecolumn\onecolumn
1210
1211 \let\LWR@origsp\sp
1212 \let\LWR@origsb\sb
1213 \LetLtxMacro{\LWR@origtextsuperscript\textsuperscript}
1214 \LetLtxMacro{\LWR@orig@textsuperscript@\textsuperscript}
1215
1216 \AtBeginDocument{
1217 \LetLtxMacro{\LWR@origtextsubscript\textsubscript}
1218 \LetLtxMacro{\LWR@orig@textsubscript@\textsubscript}
1219 }
1220
1221 \LetLtxMacro{\LWR@origunderline\underline}

1222 \let\LWR@orignewpage\newpage
1223
1224 \let\LWR@origpagestyle\pagestyle
1225 \let\LWR@origthispagestyle>thispagestyle
1226 \LetLtxMacro{\LWR@origpagenumbering\pagetitle
```

```
1227
1228 \let\LWR@orignewline\newline
1229
1230
1231 \AtBeginDocument{%
1232   \let\LWR@orig@trivlist@\trivlist
1233   \let\LWR@origtrivlist\trivlist
1234   \let\LWR@origendtrivlist\endtrivlist
1235   \LetLtxMacro{\LWR@origitem}{\item}
1236   \LetLtxMacro{\LWR@origitemize}{\itemize}
1237   \LetLtxMacro{\LWR@endorigitemize}{\enditemize}
1238   \LetLtxMacro{\LWR@origenumerate}{\enumerate}
1239   \LetLtxMacro{\LWR@endorigenumerate}{\endenumerate}
1240   \LetLtxMacro{\LWR@origdescription}{\description}
1241   \LetLtxMacro{\LWR@endorigdescription}{\enddescription}
1242   \let\LWR@orig@mklabel@\mklabel
1243   \let\LWR@origmakelabel\makelabel
1244   \let\LWR@orig@donoparitem@\donoparitem
1245   \LetLtxMacro{\LWR@orig@item}{\@item}
1246   \let\LWR@orig@nbitem@\nbitem
1247 }
1248
1249 \let\LWR@origpar\par
1250
1251 \LetLtxMacro{\LWR@origfootnote}{\footnote}
1252 \let\LWR@orig@mpfootnotetext@\mpfootnotetext
1253
1254 \let\LWR@origclearpage\clearpage
1255
1256
1257 \AtBeginDocument{%
1258   \LetLtxMacro{\LWR@orighline}{\hline}
1259   \LetLtxMacro{\LWR@origcline}{\cline}
1260 }
```

1261 \end{warpHTML}

39 Accents

Native L^AT_EX accents such as \" will work, but many more kinds of accents are available when using Unicode-aware X_HL^AT_EX and LuaL^AT_EX.

for HTML output: 1262 \begin{warpHTML}

Without \AtBeginDocument, \t was being re-defined somewhere.

```
1263 \AtBeginDocument{
```

The following are restored for print when inside a `\teximage`.

For Unicode engines, only `\t` needs to be redefined:

```
1264 \LetLtxMacro{\LWR@origt}{\t}
```

For pdfLATEX, additional work is required:

```
1265 \ifPDFTeX% pdflatex or dvi latex
1266 \LetLtxMacro{\LWR@origequalaccent}{%
1267 \LetLtxMacro{\LWR@origdotaccent}{%
1268 \LetLtxMacro{\LWR@origu}{\u}
1269 \LetLtxMacro{\LWR@origv}{\v}
1270 \LetLtxMacro{\LWR@origc}{\c}
1271 \LetLtxMacro{\LWR@origd}{\d}
1272 \LetLtxMacro{\LWR@origb}{\b}
```

The HTML redefinitions follow.

For pdfLATEX, Unicode diacritical marks are used:

```
1273 \renewcommand*{\=}[1]{\#1\HTMLunicode{0305}}
1274 \renewcommand*{\.}[1]{\#1\HTMLunicode{0307}}
1275 \renewcommand*{\u}[1]{\#1\HTMLunicode{0306}}
1276 \renewcommand*{\v}[1]{\#1\HTMLunicode{030C}}
1277 \renewcommand*{\c}[1]{\#1\HTMLunicode{0327}}
1278 \renewcommand*{\d}[1]{\#1\HTMLunicode{0323}}
1279 \renewcommand*{\b}[1]{\#1\HTMLunicode{0331}}
1280 \fi
```

For all engines, a Unicode diacritical tie is used:

```
1281 \def{\LWR@t}{\#1\HTMLunicode{0361}\#2}
1282 \renewcommand*{\t}[1]{\LWR@t}
```

`\LWR@restoreorigaccents` Called from `\restoreoriginalformatting` when a `\teximage` is begun.

```
1283 \ifPDFTeX% pdflatex or dvi latex
1284 \newcommand*{\LWR@restoreorigaccents}{%
1285 \LetLtxMacro{\=}{\LWR@origequalaccent}%
1286 \LetLtxMacro{\.}{\LWR@origdotaccent}%
1287 \LetLtxMacro{\u}{\LWR@origu}%
1288 \LetLtxMacro{\v}{\LWR@origv}%
1289 \LetLtxMacro{\t}{\LWR@origt}%
1290 \LetLtxMacro{\c}{\LWR@origc}%
1291 \LetLtxMacro{\d}{\LWR@origd}%
```

```

1292 \LetLtxMacro\b{\LWR@origb%
1293 }%
1294 \else% XeLaTeX, LuaLaTeX:
1295 \newcommand*{\LWR@restoreorigaccents}{%
1296 \LetLtxMacro\t{\LWR@origt}%
1297 }%
1298 \fi%
1299 }% AtBeginDocument

1300 \end{warpHTML}

```

40 Configuration Files

40.1 Decide whether to generate configuration files

Configuration files are only written if processing the print version of the document, and not processing a pstool image. pstool uses an additional compile for each image using the original document's preamble, which includes lwarp, so the lwarp configuration files are turned off if -pstool is part of the \jobname.

Default to no configuration files:

```
1301 \excludecomment{LWRwriteconf}
```

Generate configuration files if print mode and not -pstool:

```

for PRINT output: 1302 \begin{warpprint}
1303 \fullexpandarg%
1304 \IfSubStr*{\jobname}{-pstool}%
1305 {
1306     \typeout{lwarp: jobname with -pstool is found,}%
1307     \typeout{lwarp: \space\space not generating configuration files.}%
1308 }
1309 {
1310     \typeout{lwarp: generating configuration files}
1311     \includecomment{LWRwriteconf}
1312 }
1313 \end{warpprint}

```

40.2 project_html.tex

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

```
Config file: 1314 \begin{LWRwriteconf}
1315 \immediate\openout\LWR@quickfile=\jobname_html.tex
1316 \immediate\write\LWR@quickfile{%
1317 \detokenize{\PassOptionsToPackage}{%
1318 {warpHTML,BaseJobname=\jobname}{lwarp}}%
1319 }%
1320 \immediate\write\LWR@quickfile{%
1321 \detokenize{\input}\string{\jobname.tex}\string }%
1322 }%
1323 \immediate\closeout\LWR@quickfile
1324 \end{LWRwriteconf}
```

40.3 *lwarpmk* configuration files

Config file: 1325 \begin{LWRwriteconf}

\LWR@l warpconfversion The version number of the configuration file, allowing *lwarpmk* to detect an obsolete configuration file format. Incremented by one each time the configuration file format changes. (This is NOT the same as the lwarp version number.)

1326 \newcommand*\LWR@l warpconfversion{1} also in lwarpmk.lua

40.3.1 Helper macros

\LWR@shellescapecmd The LaTeX compile option for shell escape, if used.

```
1327 \ifshellescape
1328     \def\LWR@shellescapecmd{--shell-escape }
1329 \else
1330     \def\LWR@shellescapecmd{}
1331 \fi
```

\LWR@compilecmd {\langle engine\rangle} {\langle suffix\rangle}

Used to form the basic compilation command for a document, adding the optional shell escape.

Engine is *pdflatex*, etc. Suffix is empty or _html

```
1332 \newcommand*\LWR@compilecmd[2]{%
1333     #1 \LWR@shellescapecmd \jobname#2%
1334 }
```

\LWR@addcompilecmd {\langle cmd\rangle} {\langle suffix\rangle}

Adds to the compilation command.

Cmd is *dvipdfmx*, etc. Suffix is empty or _html

```
1335 \newcommand*{\LWR@addcompilecmd}[2]{%
1336     \LWRopseq
1337     #1 \jobname#2%
1338 }
```

\LWR@unknownengine Error message if not sure which L^AT_EX engine is being used.

```
1339 \newcommand*{\LWR@unknownengine}{%
1340     \PackageError{l warp}%
1341     {Unknown LATEX engine.}%
1342     {L warp only knows about pdflatex, dvi latex, xelatex, and lualatex.}%
1343 }
```

\LWR@latexmkvar {\langle varname\rangle} {\langle value\rangle}

Adds a *latexmk* variable assignment.

```
1344 \newcommand*{\LWR@latexmkvar}[2]{%
1345     -e
1346     \LWRopquote%
1347     \LWRdollar #1=q/#2/%
1348     \LWRopquote
1349 }
```

\LWR@latexmkcmd {\langle latexmk options\rangle}

Sets a call to *latexmk* with the given options, possibly adding --shell-escape, and also adding the indexing program.

```
1350 \newcommand*{\LWR@latexmkcmd}[1]{%
1351     latexmk \space \LWR@shellescapecmd \space #1 \space
1352     -recorder \space
1353     \LWR@latexmkvar{makeindex}{\LWR@LatexmkIndexCmd}%
1354 }
```

\LWR@latexmkdvipdfm {\langle dvipdfm or dvipdfmx\rangle}

Adds the options settings for *dvipdfm* or *dvipdfmx*.

```
1355 \newcommand*{\LWR@latexmkdvipdfm}[1]{%
1356     -pdfdvi \space
1357     \LWR@latexmkvar{dvipdf}{%
1358         #1
```

```

1359      \@percentchar 0
1360      -o \@percentchar D
1361      \@percentchar S%
1362  }
1363 }
```

\LWR@compileuplatex Sets compile options for upLATEX with uarticle or related classes.

```

1364 \newcommand*\LWR@compileuplatex{%
1365   \def\LWR@tempprintlatexcmd{%
1366     \LWR@compilecmd{uplatex}{}%
1367     \LWR@addcompilecmd{dvipdfmx}{}%
1368   }%
1369   \def\LWR@tempHTMLlatexcmd{%
1370     \LWR@compilecmd{uplatex}{_html}%
1371     \LWR@addcompilecmd{dvipdfmx}{_html}%
1372   }%
1373 }
```

\LWR@PrintLatexCmd If not set by the user, the following sets the command to use to compile the source to PDF form.
 \LWR@HTMLLatexCmd

If using *latexmk*, a complicated string is created, eventually resulting in something such as:

For *xelatex* with --shell-escape:

```
[[latexmk -xelatex --shell-escape -recorder
-e '$makeindex = q/makeindex -s lwarp.ist/' <jobname>_html]]
```

For *dvipdfmx*:

```
[[latexmk -pdfdvi -e '$dvipdf=q/dvipdfmx %0 -o %D %S /'
-recorder
-e '$makeindex=q/makeindex -s lwarp.ist/' <jobname>_html]]
```

For the following, temporary values are computed, but the permanent values are only set if the originals were not assigned by the user.

```
1374 \ifbool{\LWR@latexmk}{
```

For *latexmk* with *pdflatex* or *lualatex*:

```
1375 \ifpdf
```

For *latexmk* with *pdflatex*:

```
1376      \ifPDFTeX
1377          \def\LWR@latexcmd{\LWR@latexmkcmd{-pdf -dvi- -ps-}}
1378      \else
```

For *latexmk* with *lualatex*:

```
1379      \ifLuaTeX
1380          \def\LWR@latexcmd{\LWR@latexmkcmd{-lualatex}}
1381      \else
1382          \LWR@unknownengine
1383      \fi
1384      \fi
1385  \else% \ifpdf
```

For *latexmk* with *xelatex* or *DVI latex*:

```
1386      \ifXeTeX
```

For *latexmk* with *xelatex*:

```
1387      \def\LWR@latexcmd{\LWR@latexmkcmd{-xelatex}}
1388  \else% \ifXeTeX
```

For *latexmk* with *DVI latex*:

```
1389      \ifbool{LWR@dvipdfm}{
1390          \def\LWR@latexcmd{%
1391              \LWR@latexmkcmd{%
1392                  \LWR@latexmkdvipdfm{dvipdfm}%
1393              }%
1394          }%
1395      }{%
1396          \ifbool{LWR@dvipdfmx}{
1397              \def\LWR@latexcmd{%
1398                  \LWR@latexmkcmd{%
1399                      \LWR@latexmkdvipdfm{dvipdfmx}%
1400                  }%
1401          }%
1402      }{%
1403          \def\LWR@latexcmd{\LWR@latexmkcmd{-pdfps}}%
1404      }%
1405  }%
1406  \fi%
1407 \ifpdf
```

The final assignment if *latexmk*:

```

1408      \def\LWR@tempprintlatexcmd{\LWR@latexcmd \space \jobname}
1409      \def\LWR@tempHTMLlatexcmd{\LWR@latexcmd \space \jobname_html}
1410 }% latexmk

```

Without *latexmk*, the compiling command is simply the compiler name and the optional shell escape:

```

1411 {%
1412     \ifpdf

```

For *pdflatex* or *lualatex*:

```
1413         \ifPDFTeX
```

For *pdflatex*:

```

1414         \def\LWR@tempprintlatexcmd{\LWR@compilecmd{pdflatex}{}}
1415         \def\LWR@tempHTMLlatexcmd{\LWR@compilecmd{pdflatex}{_html}}
1416     \else
1417         \ifLuaTeX

```

For *lualatex*:

```

1418         \def\LWR@tempprintlatexcmd{\LWR@compilecmd{lualatex}{}}
1419         \def\LWR@tempHTMLlatexcmd{\LWR@compilecmd{lualatex}{_html}}
1420     \else
1421         \LWR@unknownengine
1422     \fi
1423     \fi
1424 \else% \ifpdf

```

For DVI *latex* or *xelatex*:

```
1425         \ifXeTeX
```

For *xelatex*:

```

1426         \def\LWR@tempprintlatexcmd{\LWR@compilecmd{xelatex}{}}
1427         \def\LWR@tempHTMLlatexcmd{\LWR@compilecmd{xelatex}{_html}}
1428     \else

```

For DVI *latex*. Default to *dvips*, unless told to use *dvipdfm* or *dvipdfmx*:

```
1429         \ifbool{\LWR@dvipdfm}{%
```

For DVI *latex* with *dvipdfm*:

```

1430          \def\LWR@tempprintlatexcmd{%
1431              \LWR@compilecmd{latex}{}%
1432              \LWR@addcompilecmd{dvipdfm}{}%
1433          }%
1434          \def\LWR@tempHTMLlatexcmd{%
1435              \LWR@compilecmd{latex}{_html}%
1436              \LWR@addcompilecmd{dvipdfm}{_html}%
1437          }%
1438      }{%
1439          \ifbool{\LWR@dvipdfmx}{%

```

For DVI *latex* with *dvipdfmx*:

```

1440          \def\LWR@tempprintlatexcmd{%
1441              \LWR@compilecmd{latex}{}%
1442              \LWR@addcompilecmd{dvipdfmx}{}%
1443          }%
1444          \def\LWR@tempHTMLlatexcmd{%
1445              \LWR@compilecmd{latex}{_html}%
1446              \LWR@addcompilecmd{dvipdfmx}{_html}%
1447          }%
1448      }% dvips

```

For DVI *latex* with *dvips* and *ps2pdf*:

```

1449          \def\LWR@tempprintlatexcmd{%
1450              \LWR@compilecmd{latex}{}%
1451              \LWR@addcompilecmd{dvips}{}%
1452              \LWR@addcompilecmd{ps2pdf}{} .ps
1453          }%
1454          \def\LWR@tempHTMLlatexcmd{%
1455              \LWR@compilecmd{latex}{_html}%
1456              \LWR@addcompilecmd{dvips}{_html}%
1457              \LWR@addcompilecmd{ps2pdf}{_html} .ps
1458          }%
1459      }%
1460  }
1461  \fi% \ifXeTeX
1462  \fi% \ifpdf
1463 }% latexmk

```

For *ujarticle*, *utarticle*, and related, using upLATEX and *dvipdfmx*:

```

1464 \@ifclassloaded{ujarticle}{\LWR@compileuplateX}{}%
1465 \@ifclassloaded{ujbook}{\LWR@compileuplateX}{}%
1466 \@ifclassloaded{ujreport}{\LWR@compileuplateX}{}%
1467 \@ifclassloaded{utarticle}{\LWR@compileuplateX}{}%
1468 \@ifclassloaded{utbook}{\LWR@compileuplateX}{}%
1469 \@ifclassloaded{utreport}{\LWR@compileuplateX}{}%

```

Only make the setting permanent if the original was empty:

```

1470 \ifdefempty{\LWR@PrintLatexCmd}{%
1471     \def\LWR@PrintLatexCmd{\LWR@tempprintlatexcmd}%
1472 }{}%
1473 \ifdefempty{\LWR@HTMLLatexCmd}{%
1474     \def\LWR@HTMLLatexCmd{\LWR@tempHTMLlatexcmd}%
1475 }{}%
\LWR@writeconf {<filename>}

```

Common code for each of `lwarpmk.conf` and `<project>.lwarpmkconf`. Each entry is a variable name, the equal sign, and a quoted string inside `[[and]]`, which are *lua*'s long quote characters, allowing the use of single and double quotes inside.

```

1476 \newcommand{\LWR@writeconf}[1]{%
1477 \ifcsdef{LWR@quickfile}{}{\newwrite{\LWR@quickfile}}%
1478 \immediate\openout\LWR@quickfile=\#1%
1479 \immediate\write\LWR@quickfile{confversion = [[\LWR@lwarpconfversion]]}%
1480 \ifbool{usingOSWindows}{%
1481     \immediate\write\LWR@quickfile{opsystem = [[Windows]]}%
1482 }{%
1483     \immediate\write\LWR@quickfile{opsystem = [[Unix]]}%
1484 }%
1485 \immediate\write\LWR@quickfile{sourcename = [[\jobname]]}%
1486 \immediate\write\LWR@quickfile{homehtmlfilename = [[\HomeHTMLFilename]]}%
1487 \immediate\write\LWR@quickfile{htmlfilename = [[\HTMLFilename]]}%
1488 \immediate\write\LWR@quickfile{latexmk = [[\ifbool{LWR@latexmk}{true}{false}]]}%
1489 \immediate\write\LWR@quickfile{printlatexcmd = [[\LWR@PrintLatexCmd]]}%
1490 \immediate\write\LWR@quickfile{HTMLlatexcmd = [[\LWR@HTMLLatexCmd]]}%
1491 \immediate\write\LWR@quickfile{printindexcmd = [[\LWR@PrintIndexCmd]]}%
1492 \immediate\write\LWR@quickfile{HTMLindexcmd = [[\LWR@HTMLIndexCmd]]}%
1493 \immediate\write\LWR@quickfile{latexmkindexcmd = [[\LWR@LatexmkIndexCmd]]}%
1494 \immediate\write\LWR@quickfile{glossarycmd = [[\LWR@GlossaryCmd]]}%
1495 \immediate\write\LWR@quickfile{pdftotextenc = [[\LWR@pdftotextEnc]]}%
1496 \immediate\closeout\LWR@quickfile%
1497 }%
1498
1499 \end{LWRwriteconf}

```

40.3.2 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`.

Config file: 1500 `\begin{LWRwriteconf}`
1501
1502 `\AtBeginDocument{\LWR@writeconf{lwarpmk.conf}}`
1503
1504 `\end{LWRwriteconf}`

40.3.3 <project>.lwarpmkconf

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

The `makeindex` and `xindy` options have already been handled for `lwarf.conf`.

Config file: 1505 `\begin{LWRwriteconf}`
1506
1507 `\AtBeginDocument{\LWR@writeconf{\jobname.lwarpmkconf}}`
1508
1509 `\end{LWRwriteconf}`

40.4 lwarf.css

File `lwarf.css` This is the base css layer used by `lwarf`.

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

Config file: 1510 `\begin{LWRwriteconf}`
1511 `\begin{filecontents*}{lwarf.css}`
1512 `/*`
1513 CSS stylesheet for the LaTeX lwarf package
1514 Copyright 2016-2018 Brian Dunn -- BD Tech Concepts LLC
1515 `*/`
1516
1517 `/* a fix for older browsers: */`
1518 `header, section, footer, aside, nav, main,`
1519 `article, figure { display: block; }`
1520
1521
1522
1523 `A:link {color:#000080 ; text-decoration: none ; }`
1524 `A:visited {color:#800000 ; }`
1525 `A:hover {color:#000080 ; text-decoration: underline ;}`
1526 `A:active {color:#800000 ; }`
1527

```
1528 a.tocpart {display: inline-block ; margin-left: 0em ;
1529     font-weight: bold ;}
1530 a.tocchapter {display: inline-block ; margin-left: 0em ;
1531     font-weight: bold ;}
1532 a.tocsection {display: inline-block ; margin-left: 1em ;
1533     text-indent: -.5em ; font-weight: bold ;}
1534 a.tocsubsection {display: inline-block ; margin-left: 2em ;
1535     text-indent: -.5em ;}
1536 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
1537     text-indent: -.5em ;}
1538 a.tocparagraph {display: inline-block ; margin-left: 4em ;
1539     text-indent: -.5em ;}
1540 a.toc subparagraph {display: inline-block ; margin-left: 5em ;
1541     text-indent: -.5em ;}
1542 a.tocfigure {margin-left: 0em}
1543 a.tocsubfigure {margin-left: 2em}
1544 a.toctable {margin-left: 0em}
1545 a.tocstable {margin-left: 2em}
1546 a.toctheorem {margin-left: 0em}
1547 a.toclstlisting {margin-left: 0em}
1548
1549 body {
1550     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1551         "Lucida Bright", Georgia, serif;
1552     background: #FAF7F4 ;
1553     color: black ;
1554     margin:0em ;
1555     padding:0em ;
1556     font-size: 100% ;
1557     line-height: 1.2 ;
1558 }
1559
1560 p {margin: 1.5ex 0em 1.5ex 0em ;}
1561 table p {margin: .5ex 0em .5ex 0em ;}
1562
1563 /* Holds a section number */
1564 span.sectionnumber { margin-right: 0em }
1565
1566 /* Inserted in front of index lines */
1567 span.indexitem {margin-left: 0em}
1568 span.indexsubitem {margin-left: 2em}
1569 span.indexsubsubitem {margin-left: 4em}
1570
1571 div.hidden, span.hidden { display: none ; }
1572
1573 kbd, span.texttt {
1574     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1575         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1576         "Courier New", monospace;
1577     font-size: 100% ;
```

```
1578 }
1579
1580 pre { padding: 3pt ; }
1581
1582 span.strong, span.textbf, div.strong, div.textbf { font-weight: bold; }
1583
1584 span.textit, div.textit { font-style: italic; }
1585
1586 span.textmd, div.textmd { font-weight: normal; }
1587
1588 span.textsc, div.textsc {
1589     font-variant: small-caps;
1590     font-variant-numeric: oldstyle-nums ;
1591 }
1592
1593 span.textsi, div.textsi {
1594     font-style: italic ;
1595     font-variant: small-caps;
1596     font-variant-numeric: oldstyle-nums ;
1597 }
1598
1599 span.textsl, div.textsl { font-style: oblique; }
1600
1601 span.textup, div.textup {
1602     font-style: normal;
1603     font-variant: normal;
1604     font-variant-numeric: normal ;
1605 }
1606
1607 span.textrm, div.textrm {
1608     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1609     "Lucida Bright", Georgia, serif;
1610 }
1611
1612 span.textsf, div.textsf {
1613     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1614     Geneva, Verdana, sans-serif ;
1615 }
1616
1617 span.textcircled { border: 1px solid black ; border-radius: 1ex ; }
1618
1619 span.underline {
1620     text-decoration: underline ;
1621     text-decoration-skip: auto ;
1622 }
1623
1624 span.overline {
1625     text-decoration: overline ;
1626     text-decoration-skip: auto ;
1627 }
```

```
1628
1629
1630 /* for vertical text: */
1631 div.verticalrl { writing-mode: vertical-rl }
1632 div.horizontaltb { writing-mode: horizontal-tb }
1633
1634
1635 /* for diagbox */
1636 div.diagboxtitleN { border-bottom: 1px solid gray }
1637 div.diagboxtitleS { border-top: 1px solid gray }
1638
1639 div.diagboxE {
1640     padding-left: 2em ;
1641     text-align: right ;
1642 }
1643
1644 div.diagboxW {
1645     padding-right: 2em ;
1646     text-align: left ;
1647 }
1648
1649
1650
1651 /* For realscripts */
1652 .supsubscript {
1653     display: inline-block;
1654     text-align:left ;
1655 }
1656
1657 .supsubscript sup,
1658 .supsubscript sub {
1659     position: relative;
1660     display: block;
1661     font-size: .5em;
1662     line-height: 1;
1663 }
1664
1665 .supsubscript sup {
1666     top: .5em;
1667 }
1668
1669 .supsubscript sub {
1670     top: .5em;
1671 }
1672
1673 div.attribution p {
1674     text-align: right ;
1675     font-size: 80%
1676 }
1677
```

```
1678 span.poemtitle {
1679   font-size: 120% ; font-weight: bold;
1680 }
1681
1682 pre.tabbing {
1683   font-family: "Linux Libertine Mono O", "Lucida Console",
1684   "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1685   "Liberation Mono", "FreeMono", "Andale Mono",
1686   "Nimbus Mono L", "Courier New", monospace;
1687 }
1688
1689 blockquote {
1690   display: block ;
1691   margin-left: 2em ;
1692   margin-right: 2em ;
1693 }
1694
1695 blockquotation {
1696   display: block ;
1697   margin-left: 2em ;
1698   margin-right: 2em ;
1699 }
1700
1701 /* quotchap is for the quotchap package */
1702 div.quotchap {
1703   display: block ;
1704   font-style: oblique ;
1705   overflow-x: auto ;
1706   margin-left: 2em ;
1707   margin-right: 2em ;
1708 }
1709
1710 blockquote p, blockquotation p, div.quotchap p {
1711   line-height: 1.5;
1712   text-align: left ;
1713   font-size: .85em ;
1714 }
1715
1716 /* qauthor is for the quotchap package */
1717 div.qauthor {
1718   display: block ;
1719   text-align: right ;
1720   margin-left: auto ;
1721   margin-right: 2em ;
1722   font-size: 80% ;
1723   font-variant: small-caps;
1724 }
1725
1726 div.qauthor p {
1727   text-align: right ;
```

```
1728 }
1729
1730 div.epigraph, div.dictum {
1731   line-height: 1.2;
1732   text-align: left ;
1733   padding: 3ex 1em 0ex 1em ;
1734 /*     margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
1735   margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
1736 /*     margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
1737   font-size: .85em ;
1738   max-width: 27em ;
1739 }
1740
1741 div.epigraphsource, div.dictumauthor {
1742   text-align:right ;
1743   margin-left:auto ;
1744 /*     max-width: 50% ; */
1745   border-top: 1px solid #AOAOAO ;
1746   padding-bottom: 3ex ;
1747   line-height: 1.2;
1748 }
1749
1750 div.epigraph p, div.dictum p { padding: .5ex ; margin: 0ex ;}
1751 div.epigraphsource p, div.dictumauthor p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
1752 div.dictumauthor { font-style:italic }
1753
1754
1755 /* copyrightbox package */
1756 div.copyrightbox { margin: .5ex .5em }
1757 div.copyrightbox p {margin: 0px .5em ; padding: 0px}
1758 div.copyrightboxnote {text-align: left ; font-size: 60%}
1759
1760
1761 /* lettrine package: */
1762 span.lettrine { font-size: 4ex ; float: left ; }
1763 span.lettrinetext { font-variant: small-caps ; }
1764
1765 /* ulem and soul packages: */
1766 span.uline {
1767   text-decoration: underline ;
1768   text-decoration-skip: auto ;
1769 }
1770
1771 span.uline {
1772   text-decoration: underline ;
1773   text-decoration-skip: auto ;
1774   text-decoration-style: double ;
1775 }
1776
1777 span.uwave {
```

```
1778     text-decoration: underline ;
1779     text-decoration-skip: auto ;
1780     text-decoration-style: wavy ;
1781 }
1782
1783 span.sout {
1784     text-decoration: line-through ;
1785 }
1786
1787 span.xout {
1788     text-decoration: line-through ;
1789 }
1790
1791 span.dashuline {
1792     text-decoration: underline ;
1793     text-decoration-skip: auto ;
1794     text-decoration-style: dashed ;
1795 }
1796
1797 span.dotuline {
1798     text-decoration: underline ;
1799     text-decoration-skip: auto ;
1800     text-decoration-style: dotted ;
1801 }
1802
1803 span.letterspacing { letter-spacing: .2ex ; }
1804
1805 span.capsspacing {
1806     font-variant: small-caps ;
1807     letter-spacing: .1ex ;
1808 }
1809
1810 span.highlight { background: #F8E800 ; }
1811
1812
1813
1814
1815 html body {
1816     margin: 0 ;
1817     line-height: 1.2;
1818 }
1819
1820
1821 body div {
1822     margin: 0ex;
1823 }
1824
1825
1826 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1827 {
```

```
1828     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1829         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1830         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1831         "Times New Roman", serif;
1832     font-style: normal ;
1833     font-weight: bold ;
1834     text-align: left ;
1835 }
1836
1837 h1 { /* title of the entire website, used on each page */
1838     text-align: center ;
1839     font-size: 2.5em ;
1840     padding: .4ex 0em 0ex 0em ;
1841 }
1842 h2 { font-size: 2.25em }
1843 h3 { font-size: 2em }
1844 h4 { font-size: 1.75em }
1845 h5 { font-size: 1.5em }
1846 h6 { font-size: 1.25em }
1847 span.paragraph {font-size: 1em ; font-variant: normal ;
1848     margin-right: 1em ; }
1849 span.subparagraph {font-size: 1em ; font-variant: normal ;
1850     margin-right: 1em ; }
1851
1852 div.minisec {
1853     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1854         Geneva, Verdana, sans-serif ;
1855     font-style: normal ;
1856     font-weight: bold ;
1857     text-align: left ;
1858 }
1859
1860 h1 {
1861     margin: 0ex 0em 0ex 0em ;
1862     line-height: 1.3;
1863     text-align: center ;
1864 }
1865
1866 h2 {
1867     margin: 1ex 0em 1ex 0em ;
1868     line-height: 1.3;
1869     text-align: center ;
1870 }
1871
1872 h3 {
1873     margin: 3ex 0em 1ex 0em ;
1874     line-height: 1.3;
1875 }
1876
1877 h4 {
```

```
1878   margin: 3ex 0em 1ex 0em ;
1879   line-height: 1.3;
1880 }
1881
1882 h5 {
1883   margin: 3ex 0em 1ex 0em ;
1884   line-height: 1.3;
1885 }
1886
1887 h6 {
1888   margin: 3ex 0em 1ex 0em ;
1889   line-height: 1.3;
1890 }
1891
1892
1893 div.titlepage {
1894   text-align: center ;
1895 }
1896
1897 .footnotes {
1898   text-align: left ;
1899   font-size: .85em ;
1900   margin: 3ex 2em 0ex 2em ;
1901   border-top: 1px solid silver ;
1902 }
1903
1904 .marginpar, .marginparblock {
1905   max-width: 50%;
1906   float: right ;
1907   clear: both ;
1908   text-align: left ;
1909   margin: 1ex 0.5em 1ex 1em ;
1910   padding: 1ex 0.5em 1ex 0.5em ;
1911   font-size: 85% ;
1912   border-top: 1px solid silver ;
1913   border-bottom: 1px solid silver ;
1914   overflow-x: auto ;
1915 }
1916
1917 .marginpar br { margin-bottom: 2ex ; }
1918
1919 div.marginblock, div.marginparblock {
1920   max-width:50%;
1921   float:right;
1922   text-align:left;
1923   margin: 1ex 0.5em 1ex 1em ;
1924   padding: 1ex 0.5em 1ex 0.5em ;
1925   overflow-x: auto;
1926 }
1927
```

```
1928 div.marginblock div.minipage,
1929 div.marginparblock div.minipage {
1930     display: block ;
1931     margin: Opt auto Opt auto ;
1932 }
1933
1934 div.marginblock div.minipage p ,
1935 div.marginparblock div.minipage p
1936     { font-size: 85%}
1937
1938 div.marginblock br ,
1939 div.marginparblock br
1940     { margin-bottom: 2ex ; }
1941
1942 div.bodycontainer {
1943     float: left ;
1944     width: 80% ;
1945 }
1946
1947 div.bodywithoutsidetoc div.bodycontainer {
1948     float: none ;
1949     width: 100% ;
1950 }
1951
1952 section.textbody div.footnotes{
1953     margin: 3ex 2em 0ex 2em ;
1954     border-bottom: 2px solid silver ;
1955 }
1956
1957 .footnoteheader {
1958     border-top: 2px solid silver ;
1959     margin-top: 3ex ;
1960     padding-top: 1ex ;
1961     font-weight: bold ;
1962 }
1963
1964 .mpfootnotes {
1965     text-align: left ;
1966     font-size: .85em ;
1967     margin-left: 1em ;
1968     border-top: 1px solid silver ;
1969 }
1970
1971 /* Remove footnote top border in the title page. */
1972 div.titlepage div.mpfootnotes {
1973     border-top: none ;
1974 }
1975
1976
1977
```

```
1978 ol {
1979   margin: 1ex 1em 1ex 0em;
1980   line-height: 1.2;
1981 }
1982
1983 ul, body dir, body menu {
1984   margin: 3ex 1em 3ex 0em;
1985   line-height: 1.2;
1986 }
1987
1988 li { margin: 0ex 0em 1ex 0em; }
1989
1990 html {
1991   margin: 0;
1992   padding: 0;
1993 }
1994
1995 .programlisting {
1996   font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1997     "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1998     "Courier New", monospace;
1999   margin: 1ex 0ex 1ex 0ex ;
2000   padding: .5ex 0pt .5ex 0pt ;
2001   overflow-x: auto;
2002 }
2003
2004 section.textbody>pre.programlisting {
2005 border-top: 1px solid silver ;
2006 border-bottom: 1px solid silver ;
2007 }
2008
2009
2010 div.displaymath {
2011   text-align: center ;
2012 }
2013
2014 div.displaymathnumbered {
2015   text-align: right ;
2016   margin-left: 5% ;
2017   margin-right: 5% ;
2018   min-width: 2.5in ;
2019 }
2020
2021 @media all and (min-width: 400px) {
2022   div.displaymathnumbered {
2023     margin-left: 10% ;
2024     margin-right: 10% ;
2025   }
2026 }
2027
```

```
2028 @media all and (min-width: 800px) {  
2029     div.displaymathnumbered {  
2030         margin-right: 20% ;  
2031     }  
2032 }  
2033  
2034 @media all and (min-width: 1200px) {  
2035     div.displaymathnumbered {  
2036         margin-right: 30% ;  
2037     }  
2038 }  
2039  
2040  
2041 .inlineprogramlisting {  
2042     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",  
2043             "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",  
2044             "Courier New", monospace;  
2045     overflow-x: auto;  
2046 }  
2047  
2048 span.listinglabel {  
2049     display: inline-block ;  
2050     font-size: 70% ;  
2051     width: 4em ;  
2052     text-align: right ;  
2053     margin-right: 2em ;  
2054 }  
2055  
2056 div.abstract {  
2057     margin: 2em 5% 2em 5% ;  
2058     padding: 1ex 1em 1ex 1em ;  
2059 /* font-weight: bold ; */  
2060     font-size: 90% ;  
2061     text-align: left ;  
2062 }  
2063  
2064 div.abstract dl {line-height:1.5;}  
2065 div.abstract dt {color:#304070;}  
2066  
2067 div.abstracttitle{  
2068     font-family: "URW Classico", Optima, "Linux Biolinum 0",  
2069             "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",  
2070             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;  
2071     font-weight:bold;  
2072     font-size:1.25em;  
2073     text-align: center ;  
2074 }  
2075  
2076 span.abstractrunintitle{  
2077     font-family: "URW Classico", Optima, "Linux Biolinum 0",
```

```
2078     "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2079     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2080     font-weight:bold;
2081 }
2082
2083 .verbatim {
2084     overflow-x: auto ;
2085 }
2086
2087 .alltt {
2088     overflow-x: auto ;
2089 }
2090
2091
2092 .bverbatim {
2093     margin: 1ex 0pt 1ex 0pt ;
2094     padding: .5ex 0pt .5ex 0pt ;
2095     overflow-x: auto ;
2096 }
2097
2098 .lverbatim {
2099     margin: 1ex 0pt 1ex 0pt ;
2100     padding: .5ex 0pt .5ex 0pt ;
2101     overflow-x: auto ;
2102 }
2103
2104
2105 .fancyvrb {
2106     font-size:.85em ;
2107     margin: 3ex 0pt 3ex 0pt
2108 }
2109
2110 .fancyvrblabel {
2111     font-weight:bold;
2112     text-align: center ;
2113 }
2114
2115
2116 .verse {
2117     font-family: "Linux Libertine Mono O", "Lucida Console",
2118     "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
2119     "Liberation Mono", "FreeMono", "Andale Mono",
2120     "Nimbus Mono L", "Courier New", monospace;
2121     margin-left: 1em ;
2122 }
2123
2124
2125 div.singlespace { line-height: 1.2 ; }
2126 div.onehalfspace { line-height: 1.5 ; }
2127 div.doublespace { line-height: 2 ; }
```

```
2128
2129
2130 /* Word processor format output: */
2131 div.wpfigure { border: 1px solid red ; margin: .5ex ; padding: .5ex ; }
2132 div.wptable { border: 1px solid blue ; margin: .5ex ; padding: .5ex ; }
2133 div.wpmminipage { border: 1px solid green ; margin: .5ex ; padding: .5ex ; }
2134
2135
2136
2137
2138 /* Minipage environments, vertically aligned to top, center, bottom: */
2139 .minipage, .fminipage, .fcolorminipage {
2140     /* display: inline-block ; */
2141     /* Mini pages which follow each other will be tiled. */
2142     margin: .25em .25em .25em;
2143     padding: .25em .25em .25em .25em;
2144     display: inline-flex;
2145     flex-direction: column ;
2146     overflow: auto;
2147 }
2148
2149 /* Paragraphs in the flexbox did not collapse their margins. */
2150 /* Have not yet researched this. */
2151 .minipage p {margin: .75ex 0em .75ex 0em ;}
2152
2153 .fboxBlock .minipage, .colorbox .minipage, .colorboxBlock .minipage,
2154 .fcolorbox .minipage, .fcolorboxBlock .minipage
2155     {border: none ; background: none;}
2156
2157 .fbox, .fboxBlock { border: 1px solid black ; }
2158
2159 .fbox, .fboxBlock, .fcolorbox, .fcolorboxBlock, .colorbox, .colorboxBlock,
2160 .fminipage, .fcolorminipage
2161     {display: inline-block}
2162
2163 .shadowbox, .shabox {
2164     border: 1px solid black;
2165     box-shadow: 3px 3px 3px #808080 ;
2166     border-radius: 0px ;
2167     padding: .4ex .3em .4ex .3em ;
2168     margin: 0pt .3ex 0pt .3ex ;
2169     display: inline-block ;
2170 }
2171
2172 .doublebox {
2173     border: 3px double black;
2174     border-radius: 0px ;
2175     padding: .4ex .3em .4ex .3em ;
2176     margin: 0pt .3ex 0pt .3ex ;
2177     display: inline-block ;
```

```
2178 }
2179
2180 .ovalbox, .Ovalbox {
2181   border: 1px solid black;
2182   border-radius: 1ex ;
2183   padding: .4ex .3em .4ex .3em ;
2184   margin: 0pt .3ex 0pt .3ex ;
2185   display: inline-block ;
2186 }
2187
2188 .Ovalbox { border-width: 2px ; }
2189
2190 .framebox {
2191   border: 1px solid black;
2192   border-radius: 0px ;
2193   padding: .3ex .2em 0ex .2em ;
2194   margin: 0pt .1ex 0pt .1ex ;
2195   display: inline-block ;
2196 }
2197
2198
2199 .mdframed {
2200   padding: 0ex ;
2201   margin: 2ex 0em 2ex 0em ;
2202 }
2203
2204 .mdframed p { padding: 0ex .5em 0ex .5em ; }
2205
2206 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
2207
2208 .mdframedtitle {
2209   padding: .5ex 0pt 0pt 0pt ;
2210   border-radius: 10pt 10pt 0pt 0pt ;
2211   display: block ;
2212   margin-bottom: 1ex ;
2213 }
2214
2215 .mdframedsubtitle {
2216   display: block ;
2217 }
2218
2219 .mdframedsubsubtitle {
2220   display: block ;
2221 }
2222
2223 .mdtheorem {
2224   padding: 0ex .5em 0ex .5em ;
2225   margin: 3ex 5% 3ex 5% ;
2226 }
2227
```

```
2228
2229 /* framed package */
2230 .framed, pre.boxedverbatim, fcolorbox {
2231     margin: 3ex 0em 3ex 0em ;
2232     border: 1px solid black;
2233     border-radius: 0px ;
2234     padding: .3ex 1em 0ex 1em ;
2235     display: block ;
2236 }
2237
2238 .shaded {
2239     margin: 3ex 0em 3ex 0em ;
2240     padding: .3ex 1em .3ex 1em ;
2241     display: block ;
2242 }
2243
2244 .snugframed {
2245     margin: 3ex 0em 3ex 0em ;
2246     border: 1px solid black;
2247     border-radius: 0px ;
2248     display: block ;
2249 }
2250
2251 .framedleftbar {
2252     margin: 3ex 0em 3ex 0em ;
2253     border-left: 3pt solid black;
2254     border-radius: 0px ;
2255     padding: .3ex .2em .3ex 1em ;
2256     display: block ;
2257 }
2258
2259 .framedtitle {
2260     margin: 0em ;
2261     padding: 0em ;
2262     font-size: 130%
2263 }
2264
2265 .framedtitle p { padding: .3em }
2266
2267
2268
2269 dl {
2270     margin: 1ex 2em 1ex 0em;
2271     line-height: 1.3;
2272 }
2273
2274 dl dt {
2275     margin-top: 1ex;
2276     margin-left: 1em ;
2277     font-weight: bold;
```

```
2278 }
2279
2280 dl dd p { margin-top: 0em; }
2281
2282
2283 nav {
2284     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2285             "DejaVu Sans", "Bitstream Vera Sans",
2286             Geneva, Verdana, sans-serif ;
2287     margin-bottom: 4ex ;
2288 }
2289
2290 nav p {
2291     line-height: 1.2 ;
2292     margin-top:.5ex ;
2293     margin-bottom:.5ex;
2294     font-size: .9em ;
2295 }
2296
2297
2298
2299 img, img.hyperimage, img.borderimage {
2300     max-width: 600px;
2301     border: 1px solid silver;
2302     box-shadow: 3px 3px 3px #808080 ;
2303     padding: .5% ;
2304     margin: .5% ;
2305     background: none ;
2306 }
2307
2308 img.inlineimage{
2309     padding: 0px ;
2310     box-shadow: none ;
2311     border: none ;
2312     background: none ;
2313     margin: 0px ;
2314     display: inline-block ;
2315     border-radius: 0px ;
2316 }
2317
2318 img.logoimage{
2319     max-width: 300px ;
2320     box-shadow: 3px 3px 3px #808080 ;
2321     border: 1px solid black ;
2322     background:none ;
2323     padding:0 ;
2324     margin:.5ex ;
2325     border-radius: 10px ;
2326 }
2327
```

```
2328
2329 .section {
2330 /*
2331     To have each section float relative to each other:
2332 */
2333 /*
2334     display: block ;
2335     float: left ;
2336     position: relative ;
2337     background: white ;
2338     border: 1px solid silver ;
2339     padding: .5em ;
2340 */
2341     margin: 0ex .5em 0ex .5em ;
2342     padding: 0 ;
2343 }
2344
2345
2346 figure {
2347     margin: 5ex auto 5ex auto ;
2348     padding: 1ex 1em 1ex 1em ;
2349     overflow-x: auto ;
2350 }
2351
2352
2353 /* To automatically center images in figures: */
2354 /*
2355 figure img.inlineimage {
2356     margin: 0ex auto 0ex auto ;
2357     display: block ;
2358 }
2359 */
2360
2361 /* To automatically center minipages in figures: */
2362 /*
2363 figure div.minipage, figure div.minipage div.minipage {
2364     margin: 1ex auto 1ex auto ;
2365     display: block ;
2366 }
2367 */
2368
2369 figure figure { margin: 0pt }
2370
2371 figure div.minipage p { font-size: 85% ; }
2372
2373 figure.subfigure, figure.subtable {
2374     display: inline-block ; margin: 3ex 1em 3ex 1em ;
2375 }
2376
2377 div.figurecaption .minipage { margin:0 ; padding: 0 }
```

```
2378
2379 div.minipage figure { border: none ; box-shadow: none ; }
2380 div.minipage figure.table { margin: 0ex }
2381 div.minipage div.footnotes { margin: 1ex 2em 0ex 2em }
2382
2383 div.floatrow { text-align: center; }
2384
2385 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
2386
2387 div.floatfoot { font-size: .85em ;
2388     border-top: 1px solid silver ; line-height: 1.2 ; }
2389
2390 div.figurecaption , .lstlistingtitle {
2391     font-size: .85em ;
2392     text-align: center ;
2393     font-weight: bold ;
2394     margin-top: 1ex ;
2395     margin-bottom: 1ex ;
2396 }
2397
2398 figure.subfigure div.figurecaption, figure.subtable div.figurecaption {
2399     border-bottom: none ; background: none ;
2400 }
2401
2402 div.nonfloatcaption {
2403     margin: 1ex auto 1ex auto ;
2404     font-size: .85em ;
2405     text-align: center ;
2406     font-weight: bold ;
2407 }
2408
2409 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
2410 figure div.floatrow div.minipage div.figurecaption {
2411     border: none ;
2412     background: none ;
2413 }
2414
2415
2416 /* For packages such as float, rotfloat, and algorithm2e: */
2417
2418 figure.boxed, figure.boxruled {
2419     border: 1px solid black ;
2420 }
2421
2422 figure.ruled {
2423     border-top: 1px solid black ;
2424     border-bottom: 1px solid black ;
2425     border-left: 0px ;
2426     border-right: 0px ;
2427     border-radius: 0px ;
```

```
2428     background: none ;
2429     box-shadow: none ;
2430 }
2431
2432 figure.ruled div.figurecaption, figure.boxruled div.figurecaption {
2433     border-top: 1px solid silver ;
2434     border-bottom: 1px solid silver ;
2435 }
2436
2437
2438 table {
2439     margin: 1ex auto 1ex auto ;
2440     border-collapse: separate ;
2441     border-spacing: 0px ;
2442     line-height: 1.3 ;
2443 }
2444
2445 tr.hline td {border-top: 1px solid #808080 ; margin-top: 0ex ;
2446     margin-bottom: 0ex ; } /* for \hline */
2447
2448 tr.tbrule td {border-top: 1px solid black ; margin-top: 0ex ;
2449     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
2450
2451 td {padding: .5ex .5em .5ex .5em ;}
2452
2453 table td.tdl { text-align: left ; vertical-align: middle ; }
2454 table td.tdc { text-align: center ; vertical-align: middle ; }
2455 table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
2456 table td.tdbang { text-align: center ; vertical-align: middle ; }
2457 table td.tdr { text-align: right ; vertical-align: middle ; }
2458 table td.tdp { text-align: left ; vertical-align: bottom ; }
2459 table td.tdm { text-align: left ; vertical-align: middle ; }
2460 table td.tdb { text-align: left ; vertical-align: top ; }
2461 table td.tdP { text-align: center ; vertical-align: bottom ; }
2462 table td.tdM { text-align: center ; vertical-align: middle ; }
2463 table td.tdB { text-align: center ; vertical-align: top ; }
2464
2465 table td.tvertbarl { border-left: 1px solid black }
2466 table td.tvertbarldouble { border-left: 4px double black }
2467 table td.tvertbarr { border-right: 1px solid black }
2468 table td.tvertbarrdouble { border-right: 4px double black }
2469
2470 table td.tvertbarldash { border-left: 1px dashed black }
2471 table td.tvertbarldoubledash { border-left: 2px dashed black }
2472 table td.tvertbarrdash { border-right: 1px dashed black }
2473 table td.tvertbarrdoubledash { border-right: 2px dashed black }
2474
2475
2476 /* for cmidrules: */
2477 table td.tdrule {
```

```
2478     border-top: 1px solid #AOAOAO ;
2479 }
2480
2481 table td.tdrulel {
2482     border-top-left-radius:.5em ;
2483     border-top: 1px solid #AOAOAO ;
2484 }
2485
2486 table td.tdruler {
2487     border-top-right-radius:.5em ;
2488     border-top: 1px solid #AOAOAO ;
2489 }
2490
2491 table td.tdrulelr {
2492     border-top-left-radius:.5em ;
2493     border-top-right-radius:.5em ;
2494     border-top: 1px solid #AOAOAO ;
2495 }
2496
2497
2498 /* Margins of paragraphs inside table cells: */
2499 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
2500     padding-bottom: 1ex ; margin: 0ex ; }
2501 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
2502     padding-bottom: 1ex ; margin: 0ex ; }
2503 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
2504     padding-bottom: 1ex ; margin: 0ex ; }
2505
2506 td.tdp , td.tdprule , td.tdP , td.tdPrule
2507     { padding: 0ex .5em 0ex .5em ; }
2508 td.tdm , td.tdmrule , td.tdM , td.tdMrule
2509     { padding: 0ex .5em 0ex .5em ; }
2510 td.tdb , td.tdbrule , td.tdB , td.tdBrule
2511     { padding: 0ex .5em 0ex .5em ; }
2512
2513
2514 /* table notes: */
2515 .tnotes {
2516     margin: 0ex 5% 1ex 5% ;
2517     padding: 0.5ex 1em 0.5ex 1em;
2518     font-size:.85em;
2519     text-align: left ;
2520 }
2521
2522 .tnotes dl dt p {margin-bottom:0px;}
2523
2524 .tnoteitemheader {margin-right: 1em;}
2525
2526
2527 /* for colortbl and cell color */
```

```
2528 div.cellcolor {
2529     width: 100% ;
2530     padding: .5ex .5em .5ex .5em ;
2531     margin: -.5ex -.5em -.5ex -.5em ;
2532 }
2533
2534
2535 /* for bigdelim */
2536 .ldelim, .rdelim { font-size: 200% }
2537
2538
2539 /* center, flushleft, flushright environments */
2540 div.center{text-align:center;}
2541 div.center table {margin-left:auto;margin-right:auto;}
2542 div.flushleft{text-align:left;}
2543 div.flushleft table {margin-left:0em ; margin-right:auto;}
2544 div.flushright{text-align:right;}
2545 div.flushright table {margin-left:auto ; margin-right: 0em ;}
2546
2547
2548 /* Fancybox */
2549 div.Btrivlist table tr td {
2550     padding: .2ex 0em ;
2551 }
2552
2553
2554 /* program listing callouts: */
2555 span.callout {
2556     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2557             Geneva, Verdana, sans-serif ;
2558     border-radius: .5em;
2559     background-color:black;
2560     color:white;
2561     padding:0px .25em 0px .25em;
2562     margin: 0 ;
2563     font-weight: bold;
2564     font-size:.72em ;
2565 }
2566
2567 div.programlisting pre.verbatim span.callout{
2568     font-size: .85em ;
2569 }
2570
2571 span.verbatim {
2572     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
2573             "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
2574             "Courier New", monospace;
2575 }
2576
2577
```

```
2578
2579 div.published
2580 {
2581     text-align: center ;
2582     font-variant: normal ;
2583     font-style: italic ;
2584     font-size: 1em ;
2585     margin: 1ex 0em 1ex 0em ;
2586 }
2587
2588 div.subtitle
2589 {
2590     text-align: center ;
2591     font-variant: normal ;
2592     font-style: italic ;
2593     font-size: 1.25em ;
2594     margin: 1ex 0em 1ex 0em ;
2595 }
2596
2597 div.subtitle p { margin: 1ex ; }
2598
2599 div.author
2600 {
2601     font-variant: normal ;
2602     font-style: normal ;
2603     font-size: 1em ;
2604     margin: 1ex 0em 1ex 0em ;
2605 }
2606
2607 div.oneauthor {
2608     display: inline-block ;
2609     margin: 0ex 1em 0ex 1em ;
2610 }
2611
2612 /*
2613 div.author table {
2614     margin: 1ex auto 0ex auto ;
2615     background: none ;
2616 }
2617
2618 div.author table tbody tr td { padding: .25ex ; }
2619 */
2620
2621 span.affiliation {font-size: .85em ; font-variant: small-caps; }
2622
2623 div.titledate {
2624     text-align: center ;
2625     font-size: .85em ;
2626     font-style: italic;
2627     margin: 1ex 0em 1ex 0em ;
```

```
2628 }
2629
2630
2631 nav.topnavigation{
2632     text-align: left ;
2633     padding: 0.5ex 1em 0.5ex 1em ;
2634 /*      margin: 2ex 0em 3ex 0em ; */
2635     margin: 0 ;
2636     border-bottom: 1px solid silver ;
2637     border-top: 1px solid silver ;
2638     clear:both ;
2639 }
2640
2641 nav.botnavigation{
2642     text-align: left ;
2643     padding: 0.5ex 1em 0.5ex 1em ;
2644 /*      margin: 3ex 0em 2ex 0em ; */
2645     margin: 0 ;
2646     border-top: 1px solid silver ;
2647     border-bottom: 1px solid silver ;
2648     clear:both ;
2649 }
2650
2651
2652 header {
2653     line-height: 1.2 ;
2654     font-size: 1em ;
2655     border-bottom: 1px solid silver ;
2656     margin: 0px ;
2657     padding: 2ex 1em 2ex 1em ;
2658     text-align:left ;
2659 }
2660
2661
2662 footer {
2663     font-size: .85em ;
2664     line-height: 1.2 ;
2665     margin-top: 1ex ;
2666     border-top: 1px solid silver ;
2667     padding: 2ex 1em 2ex 1em ;
2668     clear:both ;
2669     text-align:left ;
2670 }
2671
2672
2673 a.linkhome { font-weight:bold ; font-size: 1em ;}
2674
2675
2676 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
2677
```

```
2678 img.lateximage{
2679     padding: 0pt ;
2680     margin: 0pt ;
2681     box-shadow: none ;
2682     border: none ;
2683     background: none ;
2684     max-width: 100% ;
2685     border-radius: 0ex ;
2686     border: none ;
2687 }
2688
2689
2690 /* The -1px right margin compensates for the 1px right border. */
2691 /* Without this -1px margin, the body container appears below instead */
2692 /* of floating to the side. */
2693 div.sidetoccontainer {
2694     font-family: "DejaVu Serif", "Bitstream Vera Serif",
2695                 "Lucida Bright", Georgia, serif;
2696     float: left ;
2697     width: 20%;
2698     margin: 0pt -1px 3ex 0pt ;
2699     border-right: 1px solid silver;
2700     border-bottom: 1px solid silver;
2701     background: #FAF7F4 ;
2702     font-size:.9em ;
2703     border-radius: 0px 0px 20px 0px ;
2704 }
2705
2706 div.sidetoccocontents {
2707     overflow-y: auto ;
2708     width: 100% ;
2709     text-align: left ;
2710 }
2711
2712
2713 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
2714     text-indent: 0 ; }
2715
2716 nav.sidetoc p a {color:black ; font-size: .7em ;}
2717
2718 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center,
2719     border-bottom: 1px solid silver ;      }
2720
2721 nav.sidetoc a:hover {text-decoration: underline ; }
2722
2723
2724
2725 section.textbody { margin: 0ex 1em 0ex 1em ;}
2726
2727
```

```
2728 div.multicolsheading { -webkit-column-span: all;
2729     -moz-column-span: all; column-span: all; }
2730 div.multicols { -webkit-columns: 3 380px ;
2731     -moz-columns: 3 380px ; columns: 3 380px ; }
2732 div.multicols p {margin-top: 0ex}
2733
2734
2735 /* Used for xfrac and nicefrac: */
2736 span.numerator {
2737     font-size: 60% ;
2738     vertical-align: .4em ;
2739 }
2740
2741 span.denominator {
2742     font-size: 60%
2743 }
2744
2745
2746 /* Used for algorithm2e: */
2747 div.alg2evline{
2748     margin-left: 1em ;
2749     padding-left: 1em ;
2750     border-left: 1px solid black ;
2751     border-radius: 0px 0px 0px 1ex ;
2752 }
2753
2754 div.alg2evsline{
2755     margin-left: 1em ;
2756     padding-left: 1em ;
2757     border-left: 1px solid black ;
2758 }
2759
2760 div.alg2enoline{
2761     margin-left: 1em ;
2762     padding-left: 1em ;
2763 }
2764
2765 span.alg2elinenumber{
2766     margin-right: .5em ;
2767     font-size: 50% ;
2768     color: red ;
2769 }
2770
2771
2772 /* Used for algorithmicx: */
2773 span.floatright { float: right ; }
2774
2775
2776
2777
```

```
2778 /* Native LaTeX theorems: */
2779
2780 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
2781 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
2782
2783
2784 /* theorem, amsthm, and ntheorem packages */
2785
2786 span.theoremheader,
2787 span.theoremheaderplain,
2788 span.theoremheaderdefinition,
2789 span.theoremheaderbreak,
2790 span.theoremheadermarginbreak,
2791 span.theoremheaderchangebreak,
2792 span.theoremheaderchange,
2793 span.theoremheadermargin
2794 {
2795     font-style:normal ; font-weight: bold ; margin-right: 1em ;
2796 }
2797
2798 span.amsthmnameplain,
2799 span.amsthmnamedefinition,
2800 span.amsthmnumberplain,
2801 span.amsthmnumberdefinition
2802 {
2803     font-style:normal ; font-weight: bold ;
2804 }
2805
2806
2807 span.amsthmnameremark,
2808 span.amsthmnumberremark
2809 {font-style:italic ; font-weight: normal ; }
2810
2811
2812 span.amsthmnoteplain,
2813 span.amsthmnotedefinition
2814 {font-style:normal ;}
2815
2816
2817 span.theoremheaderremark,
2818 span.theoremheaderproof,
2819 span.amsthmproofname
2820 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
2821
2822 span.theoremheadersc
2823 {
2824     font-style:normal ;
2825     font-variant: small-caps ;
2826     font-weight: normal ;
2827     margin-right: 1em ;
```

```
2828 }
2829
2830 .theoremendmark {float:right}
2831
2832 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
2833 div.theorembodybreak, div.theorembodynonumberbreak,
2834 div.theorembodymarginbreak,
2835 div.theorembodychangebreak,
2836 div.theorembodychange,
2837 div.theorembodymargin
2838 {
2839     font-style:italic;
2840     margin-top: 3ex ; margin-bottom: 3ex ;
2841 }
2842
2843 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
2844 div.theorembodyplainupright, nonumberplainuprightsc,
2845 div.amsthmbodydefinition, div.amsthmbodyremark,
2846 div.amsthmproof
2847 {
2848     font-style: normal ;
2849     margin-top: 3ex ; margin-bottom: 3ex ;
2850 }
2851
2852 span.amsthmnoteremark {}
2853
2854
2855 /*
2856 For CSS LaTeX and related logos:
2857 Based on spacing demonstrated by the metafont package.
2858 */
2859
2860 .latexlogofont {
2861     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2862             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2863     font-variant: normal ;
2864 }
2865
2866 .latexlogo {
2867     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2868             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2869     font-size: 1.1em;
2870 }
2871
2872 .latexlogosup {
2873     text-transform: uppercase;
2874     letter-spacing: .03em ;
2875     font-size: 0.7em;
2876     vertical-align: 0.25em;
2877     margin-left: -0.4em;
```

```
2878   margin-right: -0.15em;
2879 }
2880
2881 .latexlogosub {
2882   text-transform: uppercase;
2883   vertical-align: -0.27ex;
2884   margin-left: -0.08em;
2885   margin-right: -0.07em;
2886   font-size: 1em;
2887 }
2888
2889 .latexlogotwoe {
2890   text-transform: none ;
2891   font-variant-numeric: oldstyle-nums ;
2892 }
2893
2894 .latexlogotwoesub {
2895   font-style:italic ;
2896   vertical-align: -0.27ex;
2897   margin-left: -0.11em;
2898   margin-right: -0.1em;
2899   font-size: 1em;
2900 }
2901
2902 .xelatexlogo {
2903   font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2904   "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2905   letter-spacing: .03em ;
2906   font-size: 1.1em;
2907 }
2908
2909 .xelatexlogosub {
2910   vertical-align: -0.27ex;
2911   margin-left: -0.0667em;
2912   margin-right: -.05em;
2913   font-size: 1em;
2914   letter-spacing: .03em ;
2915 }
2916
2917 .amslogo {
2918   font-family: "TeXGyreChorus", "URW Chancery L",
2919   "Apple Chancery", "ITC Zapf Chancery", "Monotype Corsiva",
2920   "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
2921   "Hoefler Text", Times, "Times New Roman", serif ;
2922   font-style: italic ;
2923 }
2924
2925 .lyxlogo {
2926   font-family: "URW Classico", Optima, "Linux Biolinum O",
2927   "DejaVu Sans", "Bitstream Vera Sans", Geneva,
```

```
2928         Verdana, sans-serif ;
2929 }
2930
2931
2932 /* Only display top and bottom navigation if a small screen: */
2933 /* Hide the sidetoc if a small screen: */
2934 nav.topnavigation { display:none; }
2935 nav.botnavigation { display:none; }
2936
2937 /* Only display the sidetoc's webpage title if a small screen */
2938 span.sidetocthetitle { display: none }
2939
2940 @media screen and (max-width: 50em) {
2941     div.sidetoccontainer {
2942         float: none ;
2943         width: 100% ;
2944         padding: 0 ;
2945         border-radius: 0 ;
2946         border-bottom: 1px solid black ;
2947         border-top: 1px solid black ;
2948         box-shadow: none ;
2949     }
2950     span.sidetocthetitle { display: inline }
2951     nav.botnavigation { display:block }
2952     div.bodycontainer { width: 100% }
2953     .marginpar {
2954         max-width: 100%;
2955         float: none;
2956         display:block ;
2957         margin: 1ex 1em 1ex 1em ;
2958     }
2959 }
2960
2961 @media print {
2962     body {
2963         font-family: "Linux Libertine O",
2964             "DejaVu Serif", "Bitstream Vera Serif",
2965             "Liberation Serif", "Nimbus Roman No 9 L",
2966             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2967     }
2968     div.sidetoccontainer { display:none; }
2969     nav.topnavigation { display: none; }
2970     nav.botnavigation { display: none; }
2971     div.bodycontainer { width: 100% }
2972 }
2973
2974 @media handheld {
2975     div.sidetoccontainer { display:none; }
2976     nav.topnavigation { display:block }
2977     nav.botnavigation { display:block }
```

```

2978     div.bodycontainer { width: 100% }
2979 }
2980
2981 @media projection {
2982     div.sidetoccontainer { display:none; }
2983     nav.topnavigation { display:block }
2984     nav.botnavigation { display:block }
2985     div.bodycontainer { width: 100% }
2986 }
2987 \end{filecontents*}
2988 % \end{Verbatim}%
2989 \end{LWRwriteconf}

```

40.5 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.

Config file:

```

2990 \begin{LWRwriteconf}
2991 \begin{filecontents*}{lwarp_sagebrush.css}
2992 @import url("lwarp.css") ;
2993
2994
2995 A:link {color:#105030 ; text-decoration: none ; }
2996 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
2997 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
2998 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
2999
3000
3001
3002 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
3003 {
3004     font-family: "URW Classico", Optima, "Linux Biolinum 0",
3005             "Linux Libertine O", "Liberation Serif",
3006             "Nimbus Roman No 9 L", "FreeSerif",
3007             "Hoefler Text", Times, "Times New Roman", serif;
3008     font-variant: small-caps ;
3009     font-weight: normal ;
3010     color: #304070 ;
3011     text-shadow: 2px 2px 3px #808080;
3012 }
3013
3014 h1 { /* title of the entire website, used on each page */
3015     font-variant: small-caps ;
3016     color: #304070 ;
3017     text-shadow: 2px 2px 3px #808080;

```

```
3018     background-color: #F7F7F0 ;
3019     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
3020 }
3021
3022 h1 {
3023   border-bottom: 1px solid #304070;
3024 /* border-top: 2px solid #304070; */
3025 }
3026
3027 h2 {
3028   border-bottom: 1px solid #304070;
3029 /* border-top: 2px solid #304070; */
3030   background-color: #F7F7F0 ;
3031   background-image: linear-gradient(to bottom, #F7F7F0, #DADOC0);
3032 }
3033
3034
3035
3036 div.abstract {
3037   background: #f5f5eb ;
3038   background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
3039
3040   border: 1px solid silver;
3041   border-radius: 1em ;
3042 }
3043
3044 div.abstract dl {line-height:1.5;}
3045 div.abstract dt {color:#304070;}
3046
3047 div.abstracttitle{
3048   font-family: "URW Classico", Optima, "Linux Biolinum 0",
3049     "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
3050     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
3051   font-weight:bold;
3052   font-variant: small-caps ;
3053   font-size:1.5em;
3054   border-bottom: 1px solid silver ;
3055   color: #304070 ;
3056   text-align: center ;
3057   text-shadow: 1px 1px 2px #808080;
3058 }
3059
3060 span.abstractrunintitle{
3061   font-family: "URW Classico", Optima, "Linux Biolinum 0",
3062     "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
3063     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
3064   font-weight:bold;
3065 }
3066
3067
```

```
3068 div.epigraph, div.dictum {  
3069     background: #f5f5eb ;  
3070     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);  
3071  
3072     border: 1px solid silver ;  
3073     border-radius: 1ex ;  
3074     box-shadow: 3px 3px 3px #808080 ;  
3075 }  
3076  
3077  
3078 .example {  
3079     background-color: #f5f5eb ;  
3080     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);  
3081  
3082 }  
3083  
3084 div.exampletitle{  
3085     font-family: "URW Classico", Optima, "Linux Biolinum 0",  
3086         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",  
3087         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;  
3088     font-weight:bold;  
3089     font-variant: small-caps ;  
3090     border-bottom: 1px solid silver ;  
3091     color: #304070 ;  
3092     text-align: center ;  
3093     text-shadow: 1px 1px 2px #808080;  
3094 }  
3095  
3096  
3097 .sidebar {  
3098     background-color: #f5f5eb ;  
3099     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);  
3100  
3101 }  
3102  
3103 div.sidebartitle{  
3104     font-family: "URW Classico", Optima, "Linux Biolinum 0",  
3105         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",  
3106         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;  
3107     font-weight:bold;  
3108     font-variant: small-caps ;  
3109     border-bottom: 1px solid silver ;  
3110     color: #304070 ;  
3111     text-align: center ;  
3112     text-shadow: 1px 1px 2px #808080;  
3113 }  
3114  
3115  
3116 .fancyvrblabel {  
3117     font-family: "URW Classico", Optima, "Linux Biolinum 0",
```

```
3118     "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
3119     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
3120     font-weight:bold;
3121     font-variant: small-caps ;
3122     font-size: 1.5em ;
3123     color: #304070 ;
3124     text-align: center ;
3125     text-shadow: 1px 1px 2px #808080;
3126 }
3127
3128 div.minipage {
3129     background-color: #eeeeee7 ;
3130     border: 1px solid silver ;
3131     border-radius: 1ex ;
3132 }
3133
3134 table div.minipage { background: none ; border: none ; }
3135
3136 div.framebox div.minipage {border:none ; background:none}
3137
3138 section.textbody > div.minipage {
3139     box-shadow: 3px 3px 3px #808080 ;
3140 }
3141
3142 div.fboxBlock div.minipage { box-shadow: none ; }
3143
3144 .framed .minipage , .framedleftbar .minipage {
3145     border: none ;
3146     background: none ;
3147     padding: 0ex ;
3148     margin: 0ex ;
3149 }
3150
3151 figure.figure .minipage, div.figurecaption .minipage { border: none; }
3152
3153 div.marginblock div.minipage ,
3154 div.marginparblock div.minipage
3155     { border: none; }
3156
3157 figure , div.marginblock {
3158     background-color: #eeeeee7 ;
3159     border: 1px solid silver ;
3160     border-radius: 1ex ;
3161     box-shadow: 3px 3px 3px #808080 ;
3162 }
3163
3164 figure figure {
3165     border: 1px solid silver ;
3166     margin: 0em ;
3167     box-shadow: none ;
```

```
3168 }
3169
3170 /*
3171 div.figurecaption {
3172     border-top: 1px solid silver ;
3173     border-bottom: 1px solid silver ;
3174     background-color: #e8e8e8 ;
3175 }
3176 */
3177
3178
3179 div.table {
3180     box-shadow: 3px 3px 3px #808080 ;
3181 }
3182
3183 /*
3184 .tnotes {
3185     background: #e8e8e8;
3186     border: 1px solid silver;
3187 }
3188 */
3189
3190
3191 nav.topnavigation{
3192     background-color: #b0b8b0 ;
3193     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
3194 }
3195
3196 nav.botnavigation{
3197     background-color: #b0b8b0 ;
3198     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
3199 }
3200
3201
3202
3203 header{
3204     background-color: #F7F7F0 ;
3205     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
3206 }
3207
3208 footer{
3209     background-color: #F7F7F0 ;
3210     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
3211 }
3212
3213
3214
3215 div.sidetoccontainer {
3216     background-color: #F7F7F0 ;
3217     background-image: linear-gradient(to bottom, #F7F7F0, #COCOCO);
```

```

3218     box-shadow: 3px 3px 3px #808080 ;
3219 }
3220
3221 div.sidetootitle {color: #304070 ; }
3222
3223 nav.sidetoc a:hover {
3224     color:#006000 ;
3225     text-decoration: none ;
3226     text-shadow:0px 0px 2px #a0a0a0;
3227 }
3228
3229
3230 @media screen and (max-width: 45em) {
3231     div.sidetoccontainer { border-radius: 0 ; }
3232 }
3233
3234
3235 \end{filecontents*}
3236 % \end{Verbatim}%
3237 \end{LWRwriteconf}
```

40.6 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.

Config file:

```

3238 \begin{LWRwriteconf}
3239 \begin{filecontents*}{lwarp_formal.css}
3240 @import url("lwarp.css") ;
3241
3242
3243
3244 A:link {color:#802020 ; text-decoration:none; }
3245 A:visited {color:#802020 ; text-shadow:none ;}
3246 A:hover {color:#400000 ; text-shadow:none ;}
3247 A:active {color:#C00000 ; text-shadow:none ;}
3248
3249
3250 body {
3251     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3252             "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3253             "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
3254             "Times New Roman", serif;
3255     background: #ffffcf5;
3256 }
3257
```

```
3258 span.textrm {
3259     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3260         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3261         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
3262         "Times New Roman", serif;
3263 }
3264
3265 span.textsf {
3266     font-family: "DejaVu Sans", "Bitstream Vera Sans",
3267         Geneva, Verdana, sans-serif ;
3268 }
3269
3270
3271
3272 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
3273 {
3274     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3275         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3276         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
3277         "Times New Roman", serif;
3278     color: #800000 ;
3279     text-shadow: none ;
3280 }
3281
3282 h1, h2 {
3283     background-color: #ffffcf5 ;
3284     background-image: none ;
3285     border-bottom: 1px solid #808080;
3286 /*     border-top: 2px solid #808080; */
3287 }
3288
3289 div.abstracttitle {
3290     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3291         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3292         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
3293         "Times New Roman", serif;
3294     color: black ;
3295     text-shadow: none ;
3296 }
3297
3298 span.abstractrunintitle {
3299     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3300         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3301         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
3302         "Times New Roman", serif;
3303     color: black ;
3304     text-shadow: none ;
3305 }
3306
3307 div.abstract { font-size: 100% }
```

```
3308
3309 .sidebar {
3310     background: #fffcf5;
3311     background-image: none ;
3312     margin: 2em 5% 2em 5%;
3313     padding: 0.5em 1em;
3314     border: none ;
3315     border-top : 1px solid silver;
3316     border-bottom : 1px solid silver;
3317     font-size: 90% ;
3318 }
3319
3320 div.sidebartitle{
3321     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3322         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3323         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
3324         "Times New Roman", serif;
3325     color: #800000 ;
3326     text-shadow: none ;
3327     border: none ;
3328 }
3329
3330 .example {
3331     background: #fffcf5;
3332     background-image: none ;
3333     margin: 2em 5% 2em 5%;
3334     padding: 0.5em 1em;
3335     border: none ;
3336     border-top : 1px solid silver;
3337     border-bottom : 1px solid silver;
3338 }
3339
3340 div.exampletitle{
3341     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3342         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3343         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
3344         "Times New Roman", serif;
3345     color: #800000 ;
3346     text-shadow: none ;
3347     border: none ;
3348 }
3349
3350 div.fancyvrblabel{
3351     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3352         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3353         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
3354         "Times New Roman", serif;
3355     color: #800000 ;
3356     text-shadow: none ;
3357     border: none ;
```

```
3358 }
3359
3360
3361
3362 figure {
3363     margin: 5ex 5% 5ex 5% ;
3364     padding: 1ex 1em 1ex 1em ;
3365     background-color: #ffffcf5 ;
3366     overflow-x: auto ;
3367     border: none ;
3368 /*     border-top: 1px solid silver; */
3369 /*     border-bottom: 1px solid silver; */
3370 }
3371
3372
3373 div.figurecaption , .lstlisting {
3374     border: none ;
3375 /*     border-top: 1px solid silver ; */
3376 /*     border-bottom: 1px solid silver ; */
3377     background-color: #ffffcf5 ;
3378 }
3379
3380 .tnotes {
3381     background: #ffffcf5 ;
3382 }
3383
3384 .theorem {
3385     background: none ;
3386 }
3387
3388 .minipage {
3389     background-color: #ffffcf5 ;
3390     border: none ;
3391 }
3392
3393 div.floatrow figure { border: none ; }
3394
3395 figure figure { border: none ; }
3396
3397
3398 nav.toc, nav.lof, nav.lot, nav.lol {
3399     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3400         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3401         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
3402         "Times New Roman", serif;
3403 }
3404
3405 div.sidetoccontainer {
3406     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3407         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
```

```

3408     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
3409     "Times New Roman", serif;
3410     background-image: linear-gradient(to bottom, #ffffcf5, #C0C0C0);
3411 }
3412
3413 div.sidetotitle{
3414     color: #800000 ;
3415 }
3416
3417 header{
3418     background-color: #e0e0e0 ;
3419     background-image: linear-gradient(to top, #ffffcf5, #b0b0b0);
3420     text-align:center ;
3421 }
3422
3423 footer{
3424     background-color: #e0e0e0 ;
3425     background-image: linear-gradient(to bottom, #ffffcf5, #b0b0b0);
3426     padding: 2ex 1em 2ex 1em ;
3427     text-align:left ;
3428 }
3429
3430 nav.botnavigation {
3431     background: #dedcd5 ;
3432     border-top: 1px solid black ;
3433 }
3434 \end{filecontents*}
3435 % \end{Verbatim}% for syntax highlighting
3436 \end{LWRwriteconf}

```

40.7 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.

Config file:

```

3437 \begin{LWRwriteconf}
3438 \begin{filecontents*}{sample_project.css}
3439 /* ( --- Start of project.css --- ) */
3440 /* ( --- A sample project-specific CSS file for lwarp --- ) */
3441
3442 /* Uncomment one of the following: */
3443 @import url("lwarp.css") ;
3444 /* @import url("lwarp_formal.css") ; */
3445 /* @import url("lwarp_sagebrush.css") ; */
3446
3447 /* Project-specific CSS setting follow here. */

```

```

3448 /* . . . */
3449
3450 /* ( --- End of project.css --- ) */
3451 \end{filecontents*}
3452 % \end{Verbatim} for syntax highlighting
3453 \end{LWRwriteconf}
```

40.8 lwarp.ist

File `lwarp.ist` 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.

The page compositor line is for memoir's `\specialindex`.

Config file:

```

3454 \begin{LWRwriteconf}
3455 \begin{filecontents*}{lwarp.ist}
3456 preamble
3457 "\\begin{theindex}
3458   \\providecommand*\\lettergroupDefault[1]{}
3459   \\providecommand*\\lettergroup[1]{%
3460     \\par\\textbf{\#1}\\par
3461     \\nopagebreak
3462   }
3463 "
3464 headings_flag 1
3465 heading_prefix "
3466   \\lettergroup{
3467 heading_suffix "}"
3468 delim_0 ", \\hyperindexref{"
3469 delim_1 ", \\hyperindexref{"
3470 delim_2 ", \\hyperindexref{"
3471 delim_n "}, \\hyperindexref{"
3472 delim_r "} -- \\hyperindexref{"
3473 delim_t "}"
3474 page_compositor "."
3475 \end{filecontents*}
3476 % \end{Verbatim} for syntax highlighting
3477 \end{LWRwriteconf}
```

40.9 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.

See:

[https://tex.stackexchange.com/questions/80300/
how-can-i-convince-hyperref-and-xindy-to-play-together-nicely](https://tex.stackexchange.com/questions/80300/how-can-i-convince-hyperref-and-xindy-to-play-together-nicely)

Config file:

```

3478 \begin{LWRwriteconf}
3479 \begin{filecontents*}{lwarp.xdy}
3480 (require "tex/inputenc/latin.xdy")
3481 (merge-rule "\PS *" "Postscript")
3482 (require "texindy.xdy")
3483 (require "page-ranges.xdy")
3484 (require "book-order.xdy")
3485 (define-location-class "arabic-page-numbers"
3486     ("arabic-numbers") :min-range-length 1)
3487 (require "makeindex.xdy")
3488 (define-attributes ((hyperindexref)))
3489 (markup-locref :open "\hyperindexref{" :close "}")
3490 (markup-locref :open "\hyperindexref{" :close "}" :attr "hyperpage")
3491 (markup-locref :open "\textbf{\hyperindexref{" :close "}}" :attr "textbf")
3492 (markup-locref :open "\textit{\hyperindexref{" :close "}}" :attr "textit")
3493 (define-location-class-order ("roman-page-numbers"
3494     "arabic-page-numbers"
3495     "alpha-page-numbers"
3496     "Roman-page-numbers"
3497     "Alpha-page-numbers"
3498     "see"
3499     "seealso"))
3500 \end{filecontents*}
3501 % \end{Verbatim} for syntax highlighting
3502 \end{LWRwriteconf}
```

40.10 lwarp_one_limage.cmd

File `lwarp_one_limage.cmd` Used by `lwarp` to help make `lateximages` when using `WINDOWS`.

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

The arguments are each of the three fields from `lateximages.txt`, and also the base name of the source file.

MiKTeX does not allow file `lwarp_one_limage.cmd` to be created directly by `lwarpmk`, so `lwarp_one_limage.txt` is created instead, then copied to `lwarp_one_limage.cmd` by `lwarpmk`. This occurs each time `lwarpmk` used to create `lateximages`.

```
Config file: 3503 \begin{LWRwriteconf}
3504 \begin{filecontents*}{lwarp_one_limage.txt}
3505 @echo off
3506 pdfseparate -f %1 -l %1 %4_html.pdf lateximages\lateximagetemp-%%d.pdf
3507 pdfcrop --hires lateximages\lateximagetemp-%1.pdf lateximages\%3.pdf
3508 pdftocairo -svg -noshrink lateximages\%3.pdf lateximages\%3.svg
3509 del lateximages\%3.pdf
3510 del lateximages\lateximagetemp-%1.pdf
3511 exit
3512 \end{filecontents*}
3513 \end{LWRwriteconf}
```

40.11 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.

```
Config file: 3514 \begin{LWRwriteconf}
3515 \begin{filecontents*}{lwarp_mathjax.txt}
3516 <!-- https://groups.google.com/forum/#!topic/
3517                         mathjax-users/jUtewUcE2bY -->
3518 <script type="text/x-mathjax-config">
3519 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
3520     var seteqsectionDefault = {name: "", num: 0};
3521     var seteqsections = {}, seteqsection = seteqsectionDefault;
3522     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
3523     var AMS = MathJax.Extension["TeX/AMSmath"];
3524     TEX.Definitions.Add({
3525         macros: {
3526             seteqsection: "mySection",
3527             seteqnumber: "mySetEqNumber"
3528         }
3529     });
3530
3531     PARSE.Augment({
3532         mySection: function (name) {
3533             seteqsection.num = AMS.number;
3534             var n = this.GetArgument(name);
3535             if (n === "") {
3536                 seteqsection = seteqsectionDefault;
3537             } else {
3538                 if (!seteqsections["_"+n])
3539                     seteqsections["_"+n] = {name:n, num:0};
3540                 seteqsection = seteqsections["_"+n];
3541             }
3542             AMS.number = seteqsection.num;
3543         }
3544     });
3545
3546     AMS.Definitions.Add({
3547         macros: {
3548             seteqsection: "mySection",
3549             seteqnumber: "mySetEqNumber"
3550         }
3551     });
3552
3553     AMS.Definitions.Add({
3554         macros: {
3555             seteqsection: "mySection",
3556             seteqnumber: "mySetEqNumber"
3557         }
3558     });
3559
3560     AMS.Definitions.Add({
3561         macros: {
3562             seteqsection: "mySection",
3563             seteqnumber: "mySetEqNumber"
3564         }
3565     });
3566
3567     AMS.Definitions.Add({
3568         macros: {
3569             seteqsection: "mySection",
3570             seteqnumber: "mySetEqNumber"
3571         }
3572     });
3573
3574     AMS.Definitions.Add({
3575         macros: {
3576             seteqsection: "mySection",
3577             seteqnumber: "mySetEqNumber"
3578         }
3579     });
3580
3581     AMS.Definitions.Add({
3582         macros: {
3583             seteqsection: "mySection",
3584             seteqnumber: "mySetEqNumber"
3585         }
3586     });
3587
3588     AMS.Definitions.Add({
3589         macros: {
3590             seteqsection: "mySection",
3591             seteqnumber: "mySetEqNumber"
3592         }
3593     });
3594
3595     AMS.Definitions.Add({
3596         macros: {
3597             seteqsection: "mySection",
3598             seteqnumber: "mySetEqNumber"
3599         }
3600     });
3601
3602     AMS.Definitions.Add({
3603         macros: {
3604             seteqsection: "mySection",
3605             seteqnumber: "mySetEqNumber"
3606         }
3607     });
3608
3609     AMS.Definitions.Add({
3610         macros: {
3611             seteqsection: "mySection",
3612             seteqnumber: "mySetEqNumber"
3613         }
3614     });
3615
3616     AMS.Definitions.Add({
3617         macros: {
3618             seteqsection: "mySection",
3619             seteqnumber: "mySetEqNumber"
3620         }
3621     });
3622
3623     AMS.Definitions.Add({
3624         macros: {
3625             seteqsection: "mySection",
3626             seteqnumber: "mySetEqNumber"
3627         }
3628     });
3629
3630     AMS.Definitions.Add({
3631         macros: {
3632             seteqsection: "mySection",
3633             seteqnumber: "mySetEqNumber"
3634         }
3635     });
3636
3637     AMS.Definitions.Add({
3638         macros: {
3639             seteqsection: "mySection",
3640             seteqnumber: "mySetEqNumber"
3641         }
3642     });
3643
3644     AMS.Definitions.Add({
3645         macros: {
3646             seteqsection: "mySection",
3647             seteqnumber: "mySetEqNumber"
3648         }
3649     });
3650
3651     AMS.Definitions.Add({
3652         macros: {
3653             seteqsection: "mySection",
3654             seteqnumber: "mySetEqNumber"
3655         }
3656     });
3657
3658     AMS.Definitions.Add({
3659         macros: {
3660             seteqsection: "mySection",
3661             seteqnumber: "mySetEqNumber"
3662         }
3663     });
3664
3665     AMS.Definitions.Add({
3666         macros: {
3667             seteqsection: "mySection",
3668             seteqnumber: "mySetEqNumber"
3669         }
3670     });
3671
3672     AMS.Definitions.Add({
3673         macros: {
3674             seteqsection: "mySection",
3675             seteqnumber: "mySetEqNumber"
3676         }
3677     });
3678
3679     AMS.Definitions.Add({
3680         macros: {
3681             seteqsection: "mySection",
3682             seteqnumber: "mySetEqNumber"
3683         }
3684     });
3685
3686     AMS.Definitions.Add({
3687         macros: {
3688             seteqsection: "mySection",
3689             seteqnumber: "mySetEqNumber"
3690         }
3691     });
3692
3693     AMS.Definitions.Add({
3694         macros: {
3695             seteqsection: "mySection",
3696             seteqnumber: "mySetEqNumber"
3697         }
3698     });
3699
3700     AMS.Definitions.Add({
3701         macros: {
3702             seteqsection: "mySection",
3703             seteqnumber: "mySetEqNumber"
3704         }
3705     });
3706
3707     AMS.Definitions.Add({
3708         macros: {
3709             seteqsection: "mySection",
3710             seteqnumber: "mySetEqNumber"
3711         }
3712     });
3713
3714     AMS.Definitions.Add({
3715         macros: {
3716             seteqsection: "mySection",
3717             seteqnumber: "mySetEqNumber"
3718         }
3719     });
3720
3721     AMS.Definitions.Add({
3722         macros: {
3723             seteqsection: "mySection",
3724             seteqnumber: "mySetEqNumber"
3725         }
3726     });
3727
3728     AMS.Definitions.Add({
3729         macros: {
3730             seteqsection: "mySection",
3731             seteqnumber: "mySetEqNumber"
3732         }
3733     });
3734
3735     AMS.Definitions.Add({
3736         macros: {
3737             seteqsection: "mySection",
3738             seteqnumber: "mySetEqNumber"
3739         }
3740     });
3741
3742     AMS.Definitions.Add({
3743         macros: {
3744             seteqsection: "mySection",
3745             seteqnumber: "mySetEqNumber"
3746         }
3747     });
3748
3749     AMS.Definitions.Add({
3750         macros: {
3751             seteqsection: "mySection",
3752             seteqnumber: "mySetEqNumber"
3753         }
3754     });
3755
3756     AMS.Definitions.Add({
3757         macros: {
3758             seteqsection: "mySection",
3759             seteqnumber: "mySetEqNumber"
3760         }
3761     });
3762
3763     AMS.Definitions.Add({
3764         macros: {
3765             seteqsection: "mySection",
3766             seteqnumber: "mySetEqNumber"
3767         }
3768     });
3769
3770     AMS.Definitions.Add({
3771         macros: {
3772             seteqsection: "mySection",
3773             seteqnumber: "mySetEqNumber"
3774         }
3775     });
3776
3777     AMS.Definitions.Add({
3778         macros: {
3779             seteqsection: "mySection",
3780             seteqnumber: "mySetEqNumber"
3781         }
3782     });
3783
3784     AMS.Definitions.Add({
3785         macros: {
3786             seteqsection: "mySection",
3787             seteqnumber: "mySetEqNumber"
3788         }
3789     });
3790
3791     AMS.Definitions.Add({
3792         macros: {
3793             seteqsection: "mySection",
3794             seteqnumber: "mySetEqNumber"
3795         }
3796     });
3797
3798     AMS.Definitions.Add({
3799         macros: {
3800             seteqsection: "mySection",
3801             seteqnumber: "mySetEqNumber"
3802         }
3803     });
3804
3805     AMS.Definitions.Add({
3806         macros: {
3807             seteqsection: "mySection",
3808             seteqnumber: "mySetEqNumber"
3809         }
3810     });
3811
3812     AMS.Definitions.Add({
3813         macros: {
3814             seteqsection: "mySection",
3815             seteqnumber: "mySetEqNumber"
3816         }
3817     });
3818
3819     AMS.Definitions.Add({
3820         macros: {
3821             seteqsection: "mySection",
3822             seteqnumber: "mySetEqNumber"
3823         }
3824     });
3825
3826     AMS.Definitions.Add({
3827         macros: {
3828             seteqsection: "mySection",
3829             seteqnumber: "mySetEqNumber"
3830         }
3831     });
3832
3833     AMS.Definitions.Add({
3834         macros: {
3835             seteqsection: "mySection",
3836             seteqnumber: "mySetEqNumber"
3837         }
3838     });
3839
3840     AMS.Definitions.Add({
3841         macros: {
3842             seteqsection: "mySection",
3843             seteqnumber: "mySetEqNumber"
3844         }
3845     });
3846
3847     AMS.Definitions.Add({
3848         macros: {
3849             seteqsection: "mySection",
3850             seteqnumber: "mySetEqNumber"
3851         }
3852     });
3853
3854     AMS.Definitions.Add({
3855         macros: {
3856             seteqsection: "mySection",
3857             seteqnumber: "mySetEqNumber"
3858         }
3859     });
3860
3861     AMS.Definitions.Add({
3862         macros: {
3863             seteqsection: "mySection",
3864             seteqnumber: "mySetEqNumber"
3865         }
3866     });
3867
3868     AMS.Definitions.Add({
3869         macros: {
3870             seteqsection: "mySection",
3871             seteqnumber: "mySetEqNumber"
3872         }
3873     });
3874
3875     AMS.Definitions.Add({
3876         macros: {
3877             seteqsection: "mySection",
3878             seteqnumber: "mySetEqNumber"
3879         }
3880     });
3881
3882     AMS.Definitions.Add({
3883         macros: {
3884             seteqsection: "mySection",
3885             seteqnumber: "mySetEqNumber"
3886         }
3887     });
3888
3889     AMS.Definitions.Add({
3890         macros: {
3891             seteqsection: "mySection",
3892             seteqnumber: "mySetEqNumber"
3893         }
3894     });
3895
3896     AMS.Definitions.Add({
3897         macros: {
3898             seteqsection: "mySection",
3899             seteqnumber: "mySetEqNumber"
3900         }
3901     });
3902
3903     AMS.Definitions.Add({
3904         macros: {
3905             seteqsection: "mySection",
3906             seteqnumber: "mySetEqNumber"
3907         }
3908     });
3909
3910     AMS.Definitions.Add({
3911         macros: {
3912             seteqsection: "mySection",
3913             seteqnumber: "mySetEqNumber"
3914         }
3915     });
3916
3917     AMS.Definitions.Add({
3918         macros: {
3919             seteqsection: "mySection",
3920             seteqnumber: "mySetEqNumber"
3921         }
3922     });
3923
3924     AMS.Definitions.Add({
3925         macros: {
3926             seteqsection: "mySection",
3927             seteqnumber: "mySetEqNumber"
3928         }
3929     });
3930
3931     AMS.Definitions.Add({
3932         macros: {
3933             seteqsection: "mySection",
3934             seteqnumber: "mySetEqNumber"
3935         }
3936     });
3937
3938     AMS.Definitions.Add({
3939         macros: {
3940             seteqsection: "mySection",
3941             seteqnumber: "mySetEqNumber"
3942         }
3943     });
3944
3945     AMS.Definitions.Add({
3946         macros: {
3947             seteqsection: "mySection",
3948             seteqnumber: "mySetEqNumber"
3949         }
3950     });
3951
3952     AMS.Definitions.Add({
3953         macros: {
3954             seteqsection: "mySection",
3955             seteqnumber: "mySetEqNumber"
3956         }
3957     });
3958
3959     AMS.Definitions.Add({
3960         macros: {
3961             seteqsection: "mySection",
3962             seteqnumber: "mySetEqNumber"
3963         }
3964     });
3965
3966     AMS.Definitions.Add({
3967         macros: {
3968             seteqsection: "mySection",
3969             seteqnumber: "mySetEqNumber"
3970         }
3971     });
3972
3973     AMS.Definitions.Add({
3974         macros: {
3975             seteqsection: "mySection",
3976             seteqnumber: "mySetEqNumber"
3977         }
3978     });
3979
3980     AMS.Definitions.Add({
3981         macros: {
3982             seteqsection: "mySection",
3983             seteqnumber: "mySetEqNumber"
3984         }
3985     });
3986
3987     AMS.Definitions.Add({
3988         macros: {
3989             seteqsection: "mySection",
3990             seteqnumber: "mySetEqNumber"
3991         }
3992     });
3993
3994     AMS.Definitions.Add({
3995         macros: {
3996             seteqsection: "mySection",
3997             seteqnumber: "mySetEqNumber"
3998         }
3999     });
3999
4000     AMS.Definitions.Add({
4001         macros: {
4002             seteqsection: "mySection",
4003             seteqnumber: "mySetEqNumber"
4004         }
4005     });
4006
4007     AMS.Definitions.Add({
4008         macros: {
4009             seteqsection: "mySection",
4010             seteqnumber: "mySetEqNumber"
4011         }
4012     });
4013
4014     AMS.Definitions.Add({
4015         macros: {
4016             seteqsection: "mySection",
4017             seteqnumber: "mySetEqNumber"
4018         }
4019     });
4020
4021     AMS.Definitions.Add({
4022         macros: {
4023             seteqsection: "mySection",
4024             seteqnumber: "mySetEqNumber"
4025         }
4026     });
4027
4028     AMS.Definitions.Add({
4029         macros: {
4030             seteqsection: "mySection",
4031             seteqnumber: "mySetEqNumber"
4032         }
4033     });
4034
4035     AMS.Definitions.Add({
4036         macros: {
4037             seteqsection: "mySection",
4038             seteqnumber: "mySetEqNumber"
4039         }
4040     });
4041
4042     AMS.Definitions.Add({
4043         macros: {
4044             seteqsection: "mySection",
4045             seteqnumber: "mySetEqNumber"
4046         }
4047     });
4048
4049     AMS.Definitions.Add({
4050         macros: {
4051             seteqsection: "mySection",
4052             seteqnumber: "mySetEqNumber"
4053         }
4054     });
4055
4056     AMS.Definitions.Add({
4057         macros: {
4058             seteqsection: "mySection",
4059             seteqnumber: "mySetEqNumber"
4060         }
4061     });
4062
4063     AMS.Definitions.Add({
4064         macros: {
4065             seteqsection: "mySection",
4066             seteqnumber: "mySetEqNumber"
4067         }
4068     });
4069
4070     AMS.Definitions.Add({
4071         macros: {
4072             seteqsection: "mySection",
4073             seteqnumber: "mySetEqNumber"
4074         }
4075     });
4076
4077     AMS.Definitions.Add({
4078         macros: {
4079             seteqsection: "mySection",
4080             seteqnumber: "mySetEqNumber"
4081         }
4082     });
4083
4084     AMS.Definitions.Add({
4085         macros: {
4086             seteqsection: "mySection",
4087             seteqnumber: "mySetEqNumber"
4088         }
4089     });
4090
4091     AMS.Definitions.Add({
4092         macros: {
4093             seteqsection: "mySection",
4094             seteqnumber: "mySetEqNumber"
4095         }
4096     });
4097
4098     AMS.Definitions.Add({
4099         macros: {
4100             seteqsection: "mySection",
4101             seteqnumber: "mySetEqNumber"
4102         }
4103     });
4104
4105     AMS.Definitions.Add({
4106         macros: {
4107             seteqsection: "mySection",
4108             seteqnumber: "mySetEqNumber"
4109         }
4110     });
4111
4112     AMS.Definitions.Add({
4113         macros: {
4114             seteqsection: "mySection",
4115             seteqnumber: "mySetEqNumber"
4116         }
4117     });
4118
4119     AMS.Definitions.Add({
4120         macros: {
4121             seteqsection: "mySection",
4122             seteqnumber: "mySetEqNumber"
4123         }
4124     });
4125
4126     AMS.Definitions.Add({
4127         macros: {
4128             seteqsection: "mySection",
4129             seteqnumber: "mySetEqNumber"
4130         }
4131     });
4132
4133     AMS.Definitions.Add({
4134         macros: {
4135             seteqsection: "mySection",
4136             seteqnumber: "mySetEqNumber"
4137         }
4138     });
4139
4140     AMS.Definitions.Add({
4141         macros: {
4142             seteqsection: "mySection",
4143             seteqnumber: "mySetEqNumber"
4144         }
4145     });
4146
4147     AMS.Definitions.Add({
4148         macros: {
4149             seteqsection: "mySection",
4150             seteqnumber: "mySetEqNumber"
4151         }
4152     });
4153
4154     AMS.Definitions.Add({
4155         macros: {
4156             seteqsection: "mySection",
4157             seteqnumber: "mySetEqNumber"
4158         }
4159     });
4160
4161     AMS.Definitions.Add({
4162         macros: {
4163             seteqsection: "mySection",
4164             seteqnumber: "mySetEqNumber"
4165         }
4166     });
4167
4168     AMS.Definitions.Add({
4169         macros: {
4170             seteqsection: "mySection",
4171             seteqnumber: "mySetEqNumber"
4172         }
4173     });
4174
4175     AMS.Definitions.Add({
4176         macros: {
4177             seteqsection: "mySection",
4178             seteqnumber: "mySetEqNumber"
4179         }
4180     });
4181
4182     AMS.Definitions.Add({
4183         macros: {
4184             seteqsection: "mySection",
4185             seteqnumber: "mySetEqNumber"
4186         }
4187     });
4188
4189     AMS.Definitions.Add({
4190         macros: {
4191             seteqsection: "mySection",
4192             seteqnumber: "mySetEqNumber"
4193         }
4194     });
4195
4196     AMS.Definitions.Add({
4197         macros: {
4198             seteqsection: "mySection",
4199             seteqnumber: "mySetEqNumber"
4200         }
4201     });
4202
4203     AMS.Definitions.Add({
4204         macros: {
4205             seteqsection: "mySection",
4206             seteqnumber: "mySetEqNumber"
4207         }
4208     });
4209
4210     AMS.Definitions.Add({
4211         macros: {
4212             seteqsection: "mySection",
4213             seteqnumber: "mySetEqNumber"
4214         }
4215     });
4216
4217     AMS.Definitions.Add({
4218         macros: {
4219             seteqsection: "mySection",
4220             seteqnumber: "mySetEqNumber"
4221         }
4222     });
4223
4224     AMS.Definitions.Add({
4225         macros: {
4226             seteqsection: "mySection",
4227             seteqnumber: "mySetEqNumber"
4228         }
4229     });
4230
4231     AMS.Definitions.Add({
4232         macros: {
4233             seteqsection: "mySection",
4234             seteqnumber: "mySetEqNumber"
4235         }
4236     });
4237
4238     AMS.Definitions.Add({
4239         macros: {
4240             seteqsection: "mySection",
4241             seteqnumber: "mySetEqNumber"
4242         }
4243     });
4244
4245     AMS.Definitions.Add({
4246         macros: {
4247             seteqsection: "mySection",
4248             seteqnumber: "mySetEqNumber"
4249         }
4250     });
4251
4252     AMS.Definitions.Add({
4253         macros: {
4254             seteqsection: "mySection",
4255             seteqnumber: "mySetEqNumber"
4256         }
4257     });
4258
4259     AMS.Definitions.Add({
4260         macros: {
4261             seteqsection: "mySection",
4262             seteqnumber: "mySetEqNumber"
4263         }
4264     });
4265
4266     AMS.Definitions.Add({
4267         macros: {
4268             seteqsection: "mySection",
4269             seteqnumber: "mySetEqNumber"
4270         }
4271     });
4272
4273     AMS.Definitions.Add({
4274         macros: {
4275             seteqsection: "mySection",
4276             seteqnumber: "mySetEqNumber"
4277         }
4278     });
4279
4280     AMS.Definitions.Add({
4281         macros: {
4282             seteqsection: "mySection",
4283             seteqnumber: "mySetEqNumber"
4284         }
4285     });
4286
4287     AMS.Definitions.Add({
4288         macros: {
4289             seteqsection: "mySection",
4290             seteqnumber: "mySetEqNumber"
4291         }
4292     });
4293
4294     AMS.Definitions.Add({
4295         macros: {
4296             seteqsection: "mySection",
4297             seteqnumber: "mySetEqNumber"
4298         }
4299     });
4299
4300     AMS.Definitions.Add({
4301         macros: {
4302             seteqsection: "mySection",
4303             seteqnumber: "mySetEqNumber"
4304         }
4305     });
4306
4307     AMS.Definitions.Add({
4308         macros: {
4309             seteqsection: "mySection",
4310             seteqnumber: "mySetEqNumber"
4311         }
4312     });
4313
4314     AMS.Definitions.Add({
4315         macros: {
4316             seteqsection: "mySection",
4317             seteqnumber: "mySetEqNumber"
4318         }
4319     });
4319
4320     AMS.Definitions.Add({
4321         macros: {
4322             seteqsection: "mySection",
4323             seteqnumber: "mySetEqNumber"
4324         }
4325     });
4325
4326     AMS.Definitions.Add({
4327         macros: {
4328             seteqsection: "mySection",
4329             seteqnumber: "mySetEqNumber"
4330         }
4331     });
4331
4332     AMS.Definitions.Add({
4333         macros: {
4334             seteqsection: "mySection",
4335             seteqnumber: "mySetEqNumber"
4336         }
4337     });
4337
4338     AMS.Definitions.Add({
4339         macros: {
4340             seteqsection: "mySection",
4341             seteqnumber: "mySetEqNumber"
4342         }
4343     });
4343
4344     AMS.Definitions.Add({
4345         macros: {
4346             seteqsection: "mySection",
4347             seteqnumber: "mySetEqNumber"
4348         }
4349     });
4349
4350     AMS.Definitions.Add({
4351         macros: {
4352             seteqsection: "mySection",
4353             seteqnumber: "mySetEqNumber"
4354         }
4355     });
4355
4356     AMS.Definitions.Add({
4357         macros: {
4358             seteqsection: "mySection",
4359             seteqnumber: "mySetEqNumber"
4360         }
4361     });
4361
4362     AMS.Definitions.Add({
4363         macros: {
4364             seteqsection: "mySection",
4365             seteqnumber: "mySetEqNumber"
4366         }
4367     });
4367
4368     AMS.Definitions.Add({
4369         macros: {
4370             seteqsection: "mySection",
4371             seteqnumber: "mySetEqNumber"
4372         }
4373     });
4373
4374     AMS.Definitions.Add({
4375         macros: {
4376             seteqsection: "mySection",
4377             seteqnumber: "mySetEqNumber"
4378         }
4379     });
4379
4380     AMS.Definitions.Add({
4381         macros: {
4382             seteqsection: "mySection",
4383             seteqnumber: "mySetEqNumber"
4384         }
4385     });
4385
4386     AMS.Definitions.Add({
4387         macros: {
4388             seteqsection: "mySection",
4389             seteqnumber: "mySetEqNumber"
4390         }
4391     });
4391
4392     AMS.Definitions.Add({
4393         macros: {
4394             seteqsection: "mySection",
4395             seteqnumber: "mySetEqNumber"
4396         }
4397     });
4397
4398     AMS.Definitions.Add({
4399         macros: {
4400             seteqsection: "mySection",
4401             seteqnumber: "mySetEqNumber"
4402         }
4403     });
4403
4404     AMS.Definitions.Add({
4405         macros: {
4406             seteqsection: "mySection",
4407             seteqnumber: "mySetEqNumber"
4408         }
4409     });
4409
4410     AMS.Definitions.Add({
4411         macros: {
4412             seteqsection: "mySection",
4413             seteqnumber: "mySetEqNumber"
4414         }
4415     });
4415
4416     AMS.Definitions.Add({
4417         macros: {
4418             seteqsection: "mySection",
4419             seteqnumber: "mySetEqNumber"
4420         }
4421     });
4421
4422     AMS.Definitions.Add({
4423         macros: {
4424             seteqsection: "mySection",
4425             seteqnumber: "mySetEqNumber"
4426         }
4427     });
4427
4428     AMS.Definitions.Add({
4429         macros: {
4430             seteqsection: "mySection",
4431             seteqnumber: "mySetEqNumber"
4432         }
4433     });
4433
4434     AMS.Definitions.Add({
4435         macros: {
4436             seteqsection: "mySection",
4437             seteqnumber: "mySetEqNumber"
4438         }
4439     });
4439
4440     AMS.Definitions.Add({
4441         macros: {
4442             seteqsection: "mySection",
4443             seteqnumber: "mySetEqNumber"
4444         }
4445     });
4445
4446     AMS.Definitions.Add({
4447         macros: {
4448             seteqsection: "mySection",
4449             seteqnumber: "mySetEqNumber"
4450         }
4451     });
4451
4452     AMS.Definitions.Add({
4453         macros: {
4454             seteqsection: "mySection",
4455             seteqnumber: "mySetEqNumber"
4456         }
4457     });
4457
4458     AMS.Definitions.Add({
4459         macros: {
4460             seteqsection: "mySection",
4461             seteqnumber: "mySetEqNumber"
4462         }
4463     });
4463
4464     AMS.Definitions.Add({
4465         macros: {
4466             seteqsection: "mySection",
4467             seteqnumber: "mySetEqNumber"
4468         }
4469     });
4469
4470     AMS.Definitions.Add({
4471         macros: {
4472             seteqsection: "mySection",
4473             seteqnumber: "mySetEqNumber"
4474         }
4475     });
4475
4476     AMS.Definitions.Add({
4477         macros: {
4478             seteqsection: "mySection",
4479             seteqnumber: "mySetEqNumber"
4480         }
4481     });
4481
4482     AMS.Definitions.Add({
4483         macros: {
4484             seteqsection: "mySection",
4485             seteqnumber: "mySetEqNumber"
4486         }
4487     });
4487
4488     AMS.Definitions.Add({
4489         macros: {
4490             seteqsection: "mySection",
4491             seteqnumber: "mySetEqNumber"
4492         }
4493     });
4493
4494     AMS.Definitions.Add({
4495         macros: {
4496             seteqsection: "mySection",
4497             seteqnumber: "mySetEqNumber"
4498         }
4499     });
4499
4500     AMS.Definitions.Add({
4501         macros: {
4502             seteqsection: "mySection",
4503             seteqnumber: "mySetEqNumber"
4504         }
4505     });
4505
4506     AMS.Definitions.Add({
4507         macros: {
4508             seteqsection: "mySection",
4509             seteqnumber: "mySetEqNumber"
4510         }
4511     });
4511
4512     AMS.Definitions.Add({
4513         macros: {
4514             seteqsection: "mySection",
4515             seteqnumber: "mySetEqNumber"
4516         }
4517     });
4517
4518     AMS.Definitions.Add({
4519         macros: {
4520             seteqsection: "mySection",
4521             seteqnumber: "mySetEqNumber"
4522         }
4523     });
4523
4524     AMS.Definitions.Add({
4525         macros: {
4526             seteqsection: "mySection",
4527             seteqnumber: "mySetEqNumber"
4528         }
4529     });
4529
4530     AMS.Definitions.Add({
4531         macros: {
4532             seteqsection: "mySection",
4533             seteqnumber: "mySetEqNumber"
4534         }
4535     });
4535
4536     AMS.Definitions.Add({
4537         macros: {
4538             seteqsection: "mySection",
4539             seteqnumber: "mySetEqNumber"
4540         }
4541     });
4541
4542     AMS.Definitions.Add({
4543         macros: {
4544             seteqsection: "mySection",
4545             seteqnumber: "mySetEqNumber"
4546         }
4547     });
4547
4548     AMS.Definitions.Add({
4549         macros: {
4550             seteqsection: "mySection",
4551             seteqnumber: "mySetEqNumber"
4552         }
4553     });
4553
4554     AMS.Definitions.Add({
4555         macros: {
4556             seteqsection: "mySection",
4557             seteqnumber: "mySetEqNumber"
4558         }
4559     });
4559
4560     AMS.Definitions.Add({
4561         macros: {
4562             seteqsection: "mySection",
4563             seteqnumber: "mySetEqNumber"
4564         }
4565     });
4565
4566     AMS.Definitions.Add({
4567         macros: {
4568             seteqsection: "mySection",
4569             seteqnumber: "mySetEqNumber"
4570         }
4571     });
4571
4572     AMS.Definitions.Add({
4573         macros: {
4574             seteqsection: "mySection",
4575             seteqnumber: "mySetEqNumber"
4576         }
4577     });
4577
4578     AMS.Definitions.Add({
4579         macros: {
4580             seteqsection: "mySection",
4581             seteqnumber: "mySetEqNumber"
4582         }
4583     });
4583
4584     AMS.Definitions.Add({
4585         macros: {
4586             seteqsection: "mySection",
4587             seteqnumber: "mySetEqNumber"
4588         }
4589     });
4589
4590     AMS.Definitions.Add({
4591         macros: {
4592             seteqsection: "mySection",
4593             seteqnumber: "mySetEqNumber"
4594         }
4595     });
4595
4596     AMS.Definitions.Add({
4597         macros: {
4598             seteqsection: "mySection",
4599             seteqnumber: "mySetEqNumber"
4600         }
4601     });
4601
4602     AMS.Definitions.Add({
4603         macros: {
4604             seteqsection: "mySection",
4605             seteqnumber: "mySetEqNumber"
4606         }
4607     });
4607
4608     AMS.Definitions.Add({
4609         macros: {
4610             seteqsection: "mySection",
4611             seteqnumber: "mySetEqNumber"
4612         }
4613     });
4613
4614     AMS.Definitions.Add({
4615         macros: {
4616             seteqsection: "mySection",
4617             seteqnumber: "mySetEqNumber"
4618         }
4619     });
4619
4620     AMS.Definitions.Add({
4621         macros: {
4622             seteqsection: "mySection",
4623             seteqnumber: "mySetEqNumber"
4624         }
4625     });
4625
4626     AMS.Definitions.Add({
4627         macros: {
4628             seteqsection: "mySection",
4629             seteqnumber: "mySetEqNumber"
4630         }
4631     });
4631
4632     AMS.Definitions.Add({
4633         macros: {
4634             seteqsection: "mySection",
4635             seteqnumber: "mySetEqNumber"
4
```

```
3543     },
3544     mySetEqNumber: function (name) {
3545         var n = this.GetArgument(name);
3546         if (!n || !n.match(/^ *[0-9]+ *$/))
3547             n = "";
3548         else
3549             n = parseInt(n)-1;
3550         <!-- $ syntax highlighting -->
3551         if (n === "" || n < 1)
3552             TEX.Error
3553             ("Argument to "+name+" should be a positive integer");
3554         AMS.number = n;
3555     }
3556 });
3557 MathJax.Hub.Config({
3558     TeX: {
3559         equationNumbers: {
3560             formatTag: function (n) {
3561                 <!-- if not numeric, don't include the chapter -->
3562                 if (!n.match(/^ *[0-9]+ *$/))
3563                     <!-- $ syntax highlighting -->
3564                     return "("+(n).replace(/^\./,"")+"";
3565                 else
3566                     return "("+(seteqsection.name+"."+n).replace(/^\./,"")+"";
3567             },
3568             formatID: function (n) {
3569                 n = (seteqsection.name+'.'+n).replace
3570                     (/[:] '>&]/g,"").replace(/^\./,"");
3571                 return 'mjax-eqn-' + n;
3572             }
3573         }
3574     }
3575 });
3576 });
3577 </script>
3578
3579 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
3580 <script type="text/x-mathjax-config">
3581     MathJax.Ajax.config.path["Contrib"] =
3582         "https://cdn.mathjax.org/mathjax/contrib";
3583 </script>
3584
3585 <script type="text/x-mathjax-config">
3586 MathJax.Hub.Config({
3587     TeX: {
3588         equationNumbers: {
3589             autoNumber: "AMS"
3590         }
3591     }
3592 });
3593
```

```

3593 </script>
3594
3595 <!-- Alternative CDN provider: -->
3596 <script type="text/javascript" async
3597 src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.4/MathJax.js?config=TeX-AMS_HTML-full">
3598 </script>
3599
3600 <!-- No longer supported after April 30, 2017: -->
3601 <!--
3602 <script
3603 src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
3604 </script>
3605 -->
3606
3607 \end{filecontents*}
3608 % \end{Verbatim} for syntax highlighting
3609 \end{LWRwriteconf}

```

40.12 lwarpmk.lua — lwarpmk option

Opt **lwarpmk** Creates a local copy of *lwarpmk*.

Prog **lwarpmk** Command-line utility to process l warp files and images.

parallel processing lateximages and SVG math images are generated using multiple processes in parallel. For UNIX and LINUX, every 32 images the wait command is issued to wait for the previous batch of images to finish processing before starting a new batch. For WINDOWS, every 32 images one task is dispatched with

```
START /B /WAIT /BELOWNORMAL
```

which causes the operating system to wait until this lesser-priority tasks finishes, hopefully also waiting for the normal priority tasks which were already in progress to also complete. Afterwards, the next batch of images is started.

The following is only generated if the **lwarpmk** option was given to l warp.

```

3610 \begin{LWR@createlwarpmk}

3611 \begin{filecontents*}{lwarpmk.lua}
3612 #!/usr/bin/env texlua
3613
3614 -- Copyright 2016-2018 Brian Dunn
3615
3616
3617 printversion = "v0.65"
3618 requiredconfversion = "1" -- also at *lwarpmk.conf

```

```
3619
3620 function printhelp ()
3621 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
3622 end
3623
3624
3625 function printusage ()
3626 --
3627 -- Print the usage of the lwarpmk command:
3628 --
3629 print ( [[
3630
3631 lwarpmk print [-p project]: Compile the print version if necessary.
3632 lwarpmk print1 [-p project]: Forced single compile of the print version.
3633 lwarpmk printindex [-p project]: Process print indexes.
3634 lwarpmk printglossary [-p project]: Process the glossary for the print version.
3635 lwarpmk html [-p project]: Compile the HTML version if necessary.
3636 lwarpmk html1 [-p project]: Forced single compile of the HTML version.
3637 lwarpmk htmlindex [-p project]: Process HTML indexes.
3638 lwarpmk htmlglossary [-p project]: Process the glossary for the html version.
3639 lwarpmk again [-p project]: Touch the source code to trigger recompiles.
3640 lwarpmk limages [-p project]: Process the "lateximages" created by lwarp.sty.
3641 lwarpmk pdftohtml [-p project]:
3642     For use with latexmk or a Makefile:
3643     Converts project_html.pdf to project_html.html and individual HTML files.
3644     Finishes the HTML conversion even if there was a compile error.
3645 lwarpmk pdftosvg <list of file names>: Converts each PDF file to SVG.
3646 lwarpmk epstopdf <list of file names>: Converts each EPS file to PDF.
3647 lwarpmk clean [-p project]: Remove *.aux, *.toc, *.lof/t,
3648     *.idx, *.ind, *.log, *_html_inc.*, .gl*
3649 lwarpmk cleanall [-p project]: Remove auxiliary files, project.pdf, *.html
3650 lwarpmk cleanlimages: Removes all images from the "lateximages" directory.
3651 lwarpmk -h: Print this help message.
3652 lwarpmk --help: Print this help message.
3653
3654 ]] )
3655 -- printconf ()
3656 end
3657
3658
3659 function splitfile (destfile,sourcefile)
3660 --
3661 -- Split one large sourcefile into a number of files,
3662 -- starting with destfile.
3663 -- The file is split at each occurrence of <!--|Start file|newfilename|*
3664 --
3665 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
3666 local sfile = io.open(sourcefile)
3667 io.output(destfile)
3668 for line in sfile:lines() do
```

```
3669 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
3670 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then
3671     -- split the file
3672     io.output(newfilename) ;
3673 else
3674     -- not a splitpoint
3675     io.write (line .. "\n") ;
3676 end
3677 end -- do
3678 io.close(sfile)
3679 end -- function
3680
3681
3682 function cvalueerror ( line, linenum , cvalue )
3683 --
3684 -- Incorrect value, so print an error and exit.
3685 --
3686     print ("lwarpmk: ===")
3687     print ("lwarpmk: " .. linenum .. " : " .. line ) ;
3688     print (
3689         "lwarpmk: incorrect variable value \""
3690         .. cvalue ..
3691         "\" in lwarpmk.conf.\n"
3692     ) ;
3693     print ("lwarpmk: ===")
3694 --     printconf () ;
3694     os.exit(1) ;
3695 end
3696
3697
3698 function ignoreconf ()
3699 -- Global argument index
3700 argindex = 2
3701 end
3702
3703 function loadconf ()
3704 --
3705 -- Load settings from the project's "lwarpmk.conf" file:
3706 --
3707 -- Default configuration filename:
3708 local conffile = "lwarpmk.conf"
3709 local confroot = "lwarpmk"
3710 -- Global argument index
3711 argindex = 2
3712 -- Optional configuration filename:
3713 if ( arg[argindex] == "-p" ) then
3714     argindex = argindex + 1
3715     confroot = arg[argindex]
3716     conffile = confroot.."lwarpmkconf"
3717     argindex = argindex + 1
3718 end
```

```
3719 -- Additional defaults:
3720 confversion = "0"
3721 opsystem = "Unix"
3722 latexmk = "false"
3723 printlatexcmd = ""
3724 HTMLlatexcmd = ""
3725 printindexcmd = ""
3726 HTMLIndexcmd = ""
3727 latexmkindexcmd = ""
3728 -- to be removed:
3729 -- indexprog = "makeindex"
3730 -- makeindexstyle = "lwarp.ist"
3731 -- xindylanguage = "english"
3732 -- xindycodepage = "utf8"
3733 -- xindystyle = "lwarp.xdy"
3734 -- pdftotextenc = "UTF-8"
3735 glossarycmd = "makeglossaries"
3736 -- Verify the file exists:
3737 if (lfs.attributes(conffile,"mode")==nil) then
3738     -- file not exists
3739     print ("lwarpmk: ===")
3740     print ("lwarpmk: File \" .. conffile .. \" does not exist.")
3741     print ("lwarpmk: Move to the project's source directory,")
3742     print ("lwarpmk: recompile using pdflatex, xelatex, or lualatex,")
3743     print ("lwarpmk: then try using lwarpmk again.")
3744     if (arg[argindex] ~= nil) then
3745         print (
3746             "lwarpmk: (\\" .. confroot ..
3747             "\\" does not appear to be a project name.)"
3748         )
3749     end
3750     print ("lwarpmk: ===")
3751     printhelp () ;
3752     os.exit(1) -- exit the entire lwarpmk script
3753 else -- file exists
3754 -- Read the file:
3755 print ("lwarpmk: Reading " .. conffile .. ".")
3756 local cfile = io.open(conffile)
3757 -- Scan each line, parsing each line as: name = [[string]]
3758 local linenum = 0
3759 for line in cfile:lines() do -- scan lines
3760     linenum = linenum + 1
3761     i,j,cvarname,cvalue = string.find (line, "(%w-_[^%]*%)%s*=%s*%[%(([^%])*)%]%" ) ;
3762 -- Error if incorrect enclosing characters:
3763 if ( i == nil ) then
3764     print ("lwarpmk: ===")
3765     print ("lwarpmk: " .. linenum .. " : " .. line ) ;
3766     print ("lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
3767     print ("lwarpmk: ===")
3768 --     printconf () ;
```

```
3769     os.exit(1) ;
3770 end -- nil
3771 if ( cvarname == "confversion" ) then
3772     confversion = cvalue
3773 elseif ( cvarname == "opsystem" ) then
3774     -- Verify choice of opsystem:
3775     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
3776         opsystem = cvalue
3777     else
3778         cvalueerror ( line, linenum , cvalue )
3779     end
3780 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
3781 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
3782 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
3783 elseif ( cvarname == "latexmk" ) then latexmk = cvalue
3784 elseif ( cvarname == "printlatexcmd" ) then printlatexcmd = cvalue
3785 elseif ( cvarname == "HTMLlatexcmd" ) then HTMLlatexcmd = cvalue
3786 elseif ( cvarname == "printindexcmd" ) then printindexcmd = cvalue
3787 elseif ( cvarname == "HTMLindexcmd" ) then HTMLindexcmd = cvalue
3788 elseif ( cvarname == "latexmkindexcmd" ) then latexmkindexcmd = cvalue
3789 elseif ( cvarname == "glossarycmd" ) then glossarycmd = cvalue
3790 elseif ( cvarname == "pdftotextenc" ) then pdftotextenc = cvalue
3791 else
3792     print ("lwarpmk: ===")
3793     print ("lwarpmk: " .. linenum .. " : " .. line ) ;
3794     print (
3795         "lwarpmk: Incorrect variable name \"\" .. cvarname .. \"\" in " ..
3796         conffile .. ".\n"
3797     ) ;
3798     print ("lwarpmk: ===")
3799 --     printconf () ;
3800 os.exit(1) ;
3801 end -- cvarname
3802 end -- do scan lines
3803 io.close(cfile)
3804 end -- file exists
3805 -- Error if sourcename is "lwarp".
3806 -- This could happen if a local copy of lwarp has recently been recompiled.
3807 if sourcename=="lwarp" then
3808     print ("lwarpmk: ===")
3809     print ("lwarpmk: lwarp.sty has recently been recompiled in this directory,")
3810     print ("lwarpmk: and \"lwarpmk.conf\" is no longer set for your own project.")
3811     print ("lwarpmk: Recompile your own project using pdf/lua/xelatex <projectname>.")
3812     print ("lwarpmk: After a recompile, \"lwarpmk.conf\" will be set for your project,")
3813     print ("lwarpmk: and you may again use lwarpmk.")
3814     print ("lwarpmk: ===")
3815     os.exit(1)
3816 end -- sourcename of "lwarp"
3817 -- Select some operating-system commands:
3818 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
```

```
3819     rmname = "rm"
3820     mvname = "mv"
3821     cpname = "cp"
3822     touchnamepre = "touch"
3823     touchnamepost = ""
3824     newtouchname = "touch"
3825     dirslash = "/"
3826     opquote= "\`"
3827     cmdgroupopenname = " ( "
3828     cmdgroupclosename = " ) "
3829     seqname = " && "
3830     bgname = " &"
3831 elseif opsystem=="Windows" then -- For Windows
3832     rmname = "DEL"
3833     mvname = "MOVE"
3834     cpname = "COPY"
3835     touchnamepre = "COPY /b"
3836     touchnamepost = "+,,,"
3837     newtouchname = "echo empty >"
3838     dirslash = "\\"
3839     opquote= "\\""
3840     cmdgroupopenname = ""
3841     cmdgroupclosename = ""
3842     seqname = " & "
3843     bgname = ""
3844 else
3845     print ("lwarpmk: ===")
3846     print ("lwarpmk: Select Unix or Windows for opsystem." )
3847     print ("lwarpmk: ===")
3848     os.exit(1)
3849 end --- for Windows
3850 -- Warning if the operating system does not appear to be correct,
3851 -- in case files were transferred to another system.
3852 if ( (package.config:sub(1,1)) ~= dirslash ) then
3853     print ("lwarpmk: ===")
3854     print ("lwarpmk: It appears that lwarpmk.conf is for a different operating system." )
3855     print ("lwarpmk: To adjust lwarpmk.conf for the current operating system," )
3856     print ("lwarpmk: recompile the original document using xe/lua/pdflatex." )
3857     print ("lwarpmk: ")
3858     print ("lwarpmk: lwarpmk shall attempt to continue...")
3859     print ("lwarpmk: ===")
3860 end
3861 -- Error if the configuration file's version is not current:
3862 if ( confversion ~= requiredconfversion ) then
3863     print ("lwarpmk: ===")
3864     print ("lwarpmk: The configuration files lwarpmk.conf and "..sourcename.."lwarpmkconf" )
3865     print ("lwarpmk: must be updated. To update the configuration files," )
3866     print ("lwarpmk: recompile "..sourcename.."tex using xe/lua/pdflatex," )
3867     print ("lwarpmk: then use lwarpmk again.")
3868     print ("lwarpmk: ===")
```

```
3869     os.exit(1)
3870 end
3871 end -- loadconf
3872
3873
3874 function executecheckerror ( executecommands , errormessage )
3875 --
3876 -- Execute an operating system call,
3877 -- and maybe exit with an error message.
3878 --
3879 local err
3880 err = os.execute ( executecommands )
3881 if ( err ~= 0 ) then
3882     print ("lwarpmk: ===")
3883     print ("lwarpmk: " .. errormessage )
3884     print ("lwarpmk: ===")
3885     os.exit(1)
3886 end
3887 end -- executecheckerror
3888
3889
3890 function refreshdate ()
3891 os.execute(touchnamepre .. " " .. sourcename .. ".tex" .. touchnamepost)
3892 end
3893
3894
3895
3896 function reruntoget (filesource)
3897 --
3898 -- Scan the LaTeX log file for the phrase "Rerun to get",
3899 -- indicating that the file should be compiled again.
3900 -- Return true if found.
3901 --
3902 local fsource = io.open(filesource)
3903 for line in fsource:lines() do
3904 if ( string.find(line,"Rerun to get") ~= nil ) then
3905     io.close(fsource)
3906     return true
3907 end -- if
3908 end -- do
3909 io.close(fsource)
3910 return false
3911 end
3912
3913
3914
3915 function onetime (latexcmd, fsuffix)
3916 --
3917 -- Compile one time, return true if should compile again.
3918 -- fsuffix is "" for print, "_html" for HTML output.
```

```
3919 --
3920 print("lwarpmk: Compiling with: " .. latexcmd)
3921 executecheckerror (
3922     latexcmd ,
3923     "Compile error."
3924 )
3925 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
3926 end
3927
3928
3929 function manytimes (latexcmd, fsuffix)
3930 --
3931 -- Compile up to five times.
3932 -- fsuffix is "" for print, "_html" for HTML output
3933 --
3934 if onetime(latexcmd, fsuffix) == true then
3935 if onetime(latexcmd, fsuffix) == true then
3936 if onetime(latexcmd, fsuffix) == true then
3937 if onetime(latexcmd, fsuffix) == true then
3938 if onetime(latexcmd, fsuffix) == true then
3939 end end end end end
3940 end
3941
3942
3943 function verifyfileexists (filename)
3944 --
3945 -- Exit if the given file does not exist.
3946 --
3947 if (lfs.attributes ( filename , "modification" ) == nil ) then
3948     print ("lwarpmk: ===")
3949     print ("lwarpmk: " .. filename .. " not found." ) ;
3950     print ("lwarpmk: ===")
3951     os.exit (1) ;
3952 end
3953 end
3954
3955
3956
3957 function pdftohtml ()
3958 --
3959 -- Convert <project>_html.pdf into HTML files:
3960 --
3961 -- Convert to text:
3962 print ("lwarpmk: Converting " .. sourcename
3963     .. "_html.pdf to " .. sourcename .. "_html.html")
3964 os.execute("pdftotext -enc " .. pdftotextenc .. " -nopgbrk -layout "
3965     .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
3966 -- Split the result into individual HTML files:
3967 splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
3968 end
```

```
3969
3970
3971 function removeaux ()
3972 --
3973 -- Remove auxiliary files:
3974 -- All .aux files are removed since there may be many bbl*.aux files.
3975 --
3976 os.execute ( rmname .. " *.aux " ..
3977     sourcename .. ".toc" .. sourcename .. "_html.toc" ..
3978     sourcename .. ".lof" .. sourcename .. "_html.lof" ..
3979     sourcename .. ".lot" .. sourcename .. "_html.lot" ..
3980     " *.idx" ..
3981     " *.ind" ..
3982     sourcename .. ".ps" .. sourcename .. "_html.ps" ..
3983     sourcename .. ".log" .. sourcename .. "_html.log" ..
3984     sourcename .. ".gl*" .. sourcename .. "_html.gl*" ..
3985     " *_html_inc.*"
3986     )
3987 end
3988
3989 function checkhtmldpdfexists ()
3990 --
3991 -- Error if the HTML document does not exist.
3992 -- The lateximages are drawn from the HTML PDF version of the document,
3993 -- so "lwarpmk html" must be done before "lwarpmk images".
3994 --
3995 local htmlpdffile = io.open(sourcename .. "_html.pdf", "r")
3996 if ( htmlpdffile == nil ) then
3997     print ("")
3998     print ("lwarpmk: ===")
3999     print ("lwarpmk: The HTML version of the document does not exist.")
4000     print ("lwarpmk: Enter \"lwarpmk html\" to compile the HTML version.")
4001     print ("lwarpmk: ===")
4002     os.exit(1)
4003 end
4004 io.close (htmlpdffile)
4005 end -- checkhtmldpdfexists
4006
4007
4008 function warnlimages ()
4009 --
4010 -- Warning of a missing lateximages.txt file:
4011 --
4012     print ("lwarpmk: ===")
4013     print ("lwarpmk: \\\"lateximages.txt\\\" does not exist.")
4014     print ("lwarpmk: Your project does not use SVG math or other lateximages,")
4015     print ("lwarpmk: or the file has been deleted somehow.")
4016     print ("lwarpmk: Use \\\"lwarpmk html\\\" to recompile your project")
4017     print ("lwarpmk: and recreate \\\"lateximages.txt\\\"")
4018     print ("lwarpmk: If your project does not use SVG math or other lateximages,")
```

```
4019     print ("lwarpmk: then \\\"lateximages.txt\\\" will never exist, and")
4020     print ("lwarpmk: \\\"lwarpmk limages\\\" will not be necessary.")
4021     print ("lwarpmk: ===")
4022 end -- warnlimages
4023
4024
4025 function checklimages ()
4026 --
4027 -- Check lateximages.txt to see if need to recompile first.
4028 -- If any entry has a page number of zero, then there were incorrect images.
4029 --
4030 print ("lwarpmk: Checking for a valid lateximages.txt file.")
4031 local limagesfile = io.open("lateximages.txt", "r")
4032 if ( limagesfile == nil ) then
4033     warnlimages ()
4034     os.exit(1)
4035 end
4036 -- Track warning to recompile if find a page 0
4037 local pagezerowarning = false
4038 -- Scan lateximages.txt
4039 for line in limagesfile:lines() do
4040     -- lwimgpage is the page number in the PDF which has the image
4041     -- lwimghash is true if this filename is a hash
4042     -- lwimgname is the lateximage filename root to assign for the image
4043     i,j,lwimgpage,lwimghash,lwimgname = string.find (line,"|(.*)|(.*)|(.*)|")
4044     -- For each entry:
4045     if ( (i~=nil) ) then
4046         -- If the page number is 0, image references are incorrect
4047         -- and must recompile the soure document:
4048         if ( lwimgpage == "0" ) then
4049             pagezerowarning = true
4050         end
4051     end -- if i~=nil
4052 end -- do
4053 if ( pagezerowarning ) then
4054     print ("")
4055     print ("lwarpmk: ===")
4056     print ("lwarpmk: The document must be recompiled before creating the lateximages.")
4057     print ("lwarpmk: Enter \\\"lwarpmk html\\\" again, then try \\\"lwarpmk limages\\\" again.")
4058     print ("lwarpmk: ===")
4059     os.exit(1) ;
4060 end -- pagezerowarning
4061 end -- checklimages
4062
4063
4064 function createuniximage ( lwimgfullname )
4065 --
4066 -- Create one lateximage for Unix / Linux / Mac OS.
4067 --
4068 executecheckerror (
```

```
4069 cmdgroupopenname ..
4070 "pdfseparate -f " .. lwimgpage .. " -l " .. lwimgpage .. " " ..
4071     sourcename .. "_html.pdf" ..
4072     "lateximages" .. dirslash .. "lateximagetemp-%d" .. ".pdf" ..
4073     seqname ..
4074 -- Crop the image:
4075 "pdffcrop --hires lateximages" .. dirslash .. "lateximagetemp-" ..
4076     lwimgpage .. ".pdf" ..
4077     "lateximages" .. dirslash .. lwimgname .. ".pdf" ..
4078     seqname ..
4079 -- Convert the image to svg:
4080 "pdftocairo -svg -noshrink lateximages" .. dirslash .. lwimgname .. ".pdf" ..
4081     "lateximages" .. dirslash .. lwimgname .. ".svg" ..
4082     seqname ..
4083 -- Remove the temporary files:
4084 rmname .. " lateximages" .. dirslash .. lwimgname .. ".pdf" .. seqname ..
4085 rmname .. " lateximages" .. dirslash .. "lateximagetemp-" .. lwimgpage .. ".pdf" ..
4086 cmdgroupclosename .. " >/dev/null " .. bgname
4087 ,
4088 "File error trying to convert " .. lwimgfullname
4089 )
4090 -- Every 32 images, wait for completion at below normal priority,
4091 -- allowing other image tasks to catch up.
4092 numimageprocesses = numimageprocesses + 1
4093 if ( numimageprocesses > 32 ) then
4094     numimageprocesses = 0
4095     print ( "lwarpmk: waiting" )
4096     executecheckerror ( "wait" , "File error trying to wait." )
4097 end
4098 end -- createuniximage
4099
4100
4101 function createwindowsimage ( lwimgfullname )
4102 --
4103 -- Create one lateximage for Windows.
4104 --
4105 -- Every 32 images, wait for completion at below normal priority,
4106 -- allowing other image tasks to catch up.
4107 numimageprocesses = numimageprocesses + 1
4108 if ( numimageprocesses > 32 ) then
4109     numimageprocesses = 0
4110     thiswaitcommand = "/WAIT /BELOWNORMAL"
4111     print ( "lwarpmk: waiting" )
4112 else
4113     thiswaitcommand = ""
4114 end
4115 -- Execute the image generation command
4116 executecheckerror (
4117     "start /B " .. thiswaitcommand .. " \"\"\\\" lwarp_one_limage " ..
4118     lwimgpage .. " " ..
```

```
4119     lwimghash .. " " ..
4120     lwimgname .. " " ..
4121     sourcename .. " <nul >nul"
4122     ,
4123     "File error trying to create image."
4124 )
4125 end -- createwindowsimage
4126
4127
4128 function createonelateximage ( line )
4129 --
4130 -- Given the next line of lateximages.txt, convert a single image.
4131 --
4132 -- lwimgpage is the page number in the PDF which has the image
4133 -- lwimghash is true if this filename is a hash
4134 -- lwimgname is the lateximage filename root to assign for the image
4135 i,j,lwimgpage,lwimghash,lwimgname = string.find (line,"|(.*)|(.*)|(.*)|")
4136 -- For each entry:
4137 if ( (i~=nil) ) then
4138     -- Skip if the page number is 0:
4139     if ( lwimgpage == "0" ) then
4140         pagezerowarning = true
4141     else
4142         -- Skip if this image is hashed and already exists:
4143         local lwimgfullname = "lateximages" .. dirslash .. lwimgname .. ".svg"
4144         if (
4145             (lwimghash ~= "true") or
4146             (lfs.attributes(lwimgfullname,"mode")==nil) -- file not exists
4147         )
4148         then -- not hashed or not exists:
4149             -- Print the name of the file being generated:
4150             print ( "lwarpmk: " .. lwimgname )
4151             -- Touch/create the dest so that only once instance tries to build it:
4152             executecheckerror (
4153                 newtouchname .. " " .. lwimgfullname ,
4154                 "File error trying to touch " .. lwimgfullname
4155             )
4156             -- Separate out the image into its own single-page pdf:
4157             if opsystem=="Unix" then
4158                 createuniximage (lwimgfullname)
4159             elseif opsystem=="Windows" then
4160                 createwindowsimage (lwimgfullname)
4161             end
4162             end -- not hashed or not exists
4163         end -- not page 0
4164 end -- not nil
4165 end -- createonelateximage
4166
4167
4168 function createlateximages ()
```

```
4169 --
4170 -- Create lateximages based on lateximages.txt:
4171 --
4172 -- See if the document must be recompiled first:
4173 checklimages ()
4174 -- See if the print version exists:
4175 checkhtmlpdfexists ()
4176 -- Attempt to create the lateximages:
4177 print ("lwarpmk: Creating lateximages.")
4178 local limagesfile = io.open("lateximages.txt", "r")
4179 if ( limagesfile == nil ) then
4180     warnlateximages ()
4181     os.exit(1)
4182 end
4183 -- Create the lateximages directory, ignore error if already exists
4184 err = os.execute("mkdir lateximages")
4185 -- For Windows, create lwarp_one_limage.cmd from lwarp_one_limage.txt:
4186 if opsystem=="Windows" then
4187     executecheckerror (
4188         cpname .. " lwarp_one_limage.txt lwarp_one_limage.cmd" ,
4189         "File error trying to copy lwarp_one_limage.txt to lwarp_one_limage.cmd"
4190     )
4191 end -- create lwarp_one_limage.cmd
4192 -- Track the number of parallel processes
4193 numimageprocesses = 0
4194 -- Track warning to recompile if find a page 0
4195 pagezerowarning = false
4196 -- Scan lateximages.txt
4197 for line in limagesfile:lines() do
4198     createonelateximage ( line )
4199 end -- do
4200 io.close(limagesfile)
4201 print ( "lwarpmk limages: ===" )
4202 print ( "lwarpmk limages: Wait a moment for the images to complete" )
4203 print ( "lwarpmk limages: before reloading the page." )
4204 print ( "lwarpmk limages: ===" )
4205 print ( "lwarpmk limages: Done." )
4206 if ( pagezerowarning == true ) then
4207     print ( "lwarpmk limages: WARNING: Images will be incorrect." )
4208     print ( "lwarpmk limages: Enter \"lwarpmk cleanimages\", then" )
4209     print ( "lwarpmk limages: recompile the document one more time, then" )
4210     print ( "lwarpmk limages: repeat \"lwarpmk images\" again." )
4211 end -- pagezerowarning
4212 end -- function
4213
4214
4215 function convertepstopdf ()
4216 --
4217 -- Converts EPS files to PDF files.
4218 -- The filenames are arg[argindex] and up.
```

```
4219 -- arg[1] is the command "pdftosvg".
4220 --
4221 ignoreconf ()
4222 for i = argindex , #arg do
4223     if (lfs.attributes(arg[i],"mode")==nil) then
4224         print ("lwarpmk: File \" .. arg[i] .. \" does not exist.")
4225     else
4226         print ("lwarpmk: Converting \" .. arg[i] .. \"")
4227         os.execute ( "epstopdf " .. arg[i] )
4228     end -- if
4229 end -- do
4230 end --function
4231
4232
4233 function convertpdftosvg ()
4234 --
4235 -- Converts PDF files to SVG files.
4236 -- The filenames are arg[argindex] and up.
4237 -- arg[1] is the command "pdftosvg".
4238 --
4239 ignoreconf ()
4240 for i = argindex , #arg do
4241     if (lfs.attributes(arg[i],"mode")==nil) then
4242         print ("lwarpmk: File \" .. arg[i] .. \" does not exist.")
4243     else
4244         print ("lwarpmk: Converting \" .. arg[i] .. \"")
4245         os.execute ( "pdftocairo -svg " .. arg[i] )
4246     end -- if
4247 end -- do
4248 end --function
4249
4250
4251 -- Force an update and conclude processing:
4252 function updateanddone ()
4253 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
4254 refreshdate ()
4255 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
4256 print ("lwarpmk: Done.")
4257 end -- function
4258
4259
4260 -- Start of the main code: --
4261
4262
4263 -- lwarpmk --version :
4264
4265 if (arg[1] == "--version") then
4266 print ( "lwarpmk: " .. printversion )
4267
4268 else -- not --version
```

```
4269
4270
4271 -- print intro:
4272
4273 print ("lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package.")
4274
4275
4276 -- lwarpmk print:
4277
4278 if arg[1] == "print" then
4279   loadconf ()
4280   if ( latexmk == "true" ) then
4281     print ("lwarpmk: Compiling with: " .. printlatexcmd)
4282     executecheckerror (
4283       printlatexcmd ,
4284       "Compile error."
4285     )
4286     print ("lwarpmk: Done.")
4287 else -- not latexmk
4288   verifyfileexists (sourcename .. ".tex") ;
4289   -- See if up to date:
4290   if (
4291     ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
4292     (
4293       lfs.attributes ( sourcename .. ".tex" , "modification" ) >
4294       lfs.attributes ( sourcename .. ".pdf" , "modification" )
4295     )
4296   ) then
4297     -- Recompile if not yet up to date:
4298     manytimes(printlatexcmd, "")
4299     print ("lwarpmk: Done.") ;
4300   else
4301     print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
4302   end
4303 end -- not latexmk
4304
4305
4306 -- lwarpmk print1:
4307
4308 elseif arg[1] == "print1" then
4309   loadconf ()
4310   verifyfileexists (sourcename .. ".tex") ;
4311   onetime(printlatexcmd, "")
4312   print ("lwarpmk: Done.") ;
4313
4314
4315 -- lwarpmk printindex:
4316 -- Compile the index then touch the source
4317 -- to trigger a recompile of the document:
4318
```

```
4319 elseif arg[1] == "printindex" then
4320 loadconf ()
4321 os.execute ( printindexcmd )
4322 print ("lwarpmk: -----")
4323 updateanddone ()
4324
4325
4326 -- lwarpmk printglossary:
4327 -- Compile the glossary then touch the source
4328 -- to trigger a recompile of the document:
4329
4330 elseif arg[1] == "printglossary" then
4331 loadconf ()
4332 print ("lwarpmk: Processing the glossary.")
4333
4334 os.execute(glossarycmd .. " " .. sourcename)
4335 updateanddone ()
4336
4337
4338 -- lwarpmk html:
4339
4340 elseif arg[1] == "html" then
4341 loadconf ()
4342 if ( latexmk == "true" ) then
4343     print ("lwarpmk: Compiling with: " .. HTMLlatexcmd)
4344     executecheckerror (
4345         HTMLlatexcmd ,
4346         "Compile error."
4347     )
4348     pdftohtml ()
4349     print ("lwarpmk: Done.")
4350 else -- not latexmk
4351     verifyfileexists ( sourcename .. ".tex" ) ;
4352     -- See if exists and is up to date:
4353     if (
4354         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
4355         (
4356             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
4357             lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
4358         )
4359     ) then
4360         -- Recompile if not yet up to date:
4361         manytimes(HTMLlatexcmd, "_html")
4362         pdftohtml ()
4363         print ("lwarpmk: Done.")
4364     else
4365         print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
4366     end
4367 end -- not latexmk
4368
```

```
4369
4370 -- lwarfpmk html1:
4371
4372 elseif arg[1] == "html1" then
4373     loadconf ()
4374     verifyfileexists ( sourcename .. ".tex" ) ;
4375     onetime(HTMLlateXcmd, "_html")
4376     pdftohtml ()
4377     print ("lwarfpmk: Done.")
4378
4379
4380 -- lwarfpmk pdftohtml:
4381 elseif arg[1] == "pdftohtml" then
4382     loadconf ()
4383     pdftohtml ()
4384
4385
4386 -- lwarfpmk htmlindex:
4387 -- Compile the index then touch the source
4388 -- to trigger a recompile of the document:
4389
4390 elseif arg[1] == "htmlindex" then
4391 loadconf ()
4392 os.execute ( HTMLIndexcmd )
4393 print ("lwarfpmk: -----")
4394 updateanddone ()
4395
4396
4397 -- lwarfpmk htmlglossary:
4398 -- Compile the glossary then touch the source
4399 -- to trigger a recompile of the document.
4400 -- The <sourcename>.xdy file is created by the glossaries package.
4401
4402 elseif arg[1] == "htmlglossary" then
4403 loadconf ()
4404 print ("lwarfpmk: Processing the glossary.")
4405 os.execute(glossarycmd .. " " .. sourcename .. "_html")
4406 updateanddone ()
4407
4408
4409 -- lwarfpmk limages:
4410 -- Scan the lateximages.txt file to create lateximages.
4411
4412 elseif arg[1] == "limages" then
4413 loadconf ()
4414 print ("lwarfpmk: Processing images.")
4415 createlateximages ()
4416 print ("lwarfpmk: Done.")
4417
4418
```

```
4419 -- lwarpmk again:  
4420 -- Touch the source to trigger a recompile.  
4421  
4422 elseif arg[1] == "again" then  
4423 loadconf ()  
4424 updateanddone ()  
4425  
4426  
4427 -- lwarpmk clean:  
4428 -- Remove project.aux, .toc, .lof, .lot, .log, *.idx, *.ind, *_html_inc.*, .gl*  
4429  
4430 elseif arg[1] == "clean" then  
4431 loadconf ()  
4432 removeaux ()  
4433 print ("lwarpmk: Done.")  
4434  
4435  
4436 -- lwarpmk cleanall  
4437 -- Remove project.aux, .toc, .lof, .lot, .log, *.idx, *.ind, *_html_inc.*, .gl*  
4438 -- and also project.pdf, project.dvi, *.html  
4439  
4440 elseif arg[1] == "cleanall" then  
4441 loadconf ()  
4442 removeaux ()  
4443 os.execute ( rmname .. " " ..  
4444     sourcename .. ".pdf" .. sourcename .. "_html.pdf" ..  
4445     sourcename .. ".dvi" .. sourcename .. "_html.dvi" ..  
4446     "*.html"  
4447     )  
4448 print ("lwarpmk: Done.")  
4449  
4450  
4451 -- lwarpmk cleanimages  
4452 -- Remove images from the lateximages directory.  
4453  
4454 elseif arg[1] == "cleanimages" then  
4455 loadconf ()  
4456 os.execute ( rmname .. " lateximages/*" )  
4457 print ("lwarpmk: Done.")  
4458  
4459 -- lwarpmk epstopdf <list of file names>  
4460 -- Convert EPS files to PDF using epstopdf  
4461 elseif arg[1] == "epstopdf" then  
4462 convertepstopdf ()  
4463 print ("lwarpmk: Done.")  
4464  
4465  
4466 -- lwarpmk pdftosvg <list of file names>  
4467 -- Convert PDF files to SVG using pdftocairo  
4468 elseif arg[1] == "pdftosvg" then
```

```

4469 convertpdftosvg ()
4470 print ("lwarfpmk: Done.")
4471
4472
4473 -- lwarfpmk with no argument :
4474
4475 elseif (arg[1] == nil) then
4476 printhelp ()
4477
4478
4479 -- lwarfpmk -h or lwarfpmk --help :
4480
4481 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
4482 printusage ()
4483
4484
4485 -- Unknown command:
4486
4487 else
4488 printhelp ()
4489 print ("\nlwarfpmk: ***** Unknown command \"...arg[1]..\". *****\n")
4490 end
4491
4492 end -- not --version
4493 \end{filecontents*}
4494 % \end{Verbatim} for syntax highlighting

4495 \end{LWR@createlwarfpmk}

```

41 Stacks

for HTML output: 4496 \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.



41.1 Assigning depths

initial depths for empty stack entries:

```
4497 \newcommand*\{LWR@depthnone\}{-5}
```

all sectioning depths are deeper than LWR@depthfinished:

```
4498 \newcommand*\{LWR@depthfinished\}{-4}
4499 \newcommand*\{LWR@depthpart\}{-1}
4500 \newcommand*\{LWR@depthchapter\}{0}
4501 \newcommand*\{LWR@depthsection\}{1}
4502 \newcommand*\{LWR@depthsubsection\}{2}
4503 \newcommand*\{LWR@depthsubsubsection\}{3}
4504 \newcommand*\{LWR@depthparagraph\}{4}
4505 \newcommand*\{LWR@depth subparagraph\}{5}
```

used by \itemize, \enumerate, \description:

```
4506 \newcommand*\{LWR@depthlist\}{6}
```

used by \item:

```
4507 \newcommand*\{LWR@depthlistitem\}{7}
```

41.2 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.

```
4508 \newcommand*\{LWR@closeone\}{}% top of the stack
4509 \newcommand*\{LWR@closetwo\}{}%
4510 \newcommand*\{LWR@closethree\}{}%
4511 \newcommand*\{LWR@closefour\}{}%
4512 \newcommand*\{LWR@closefive\}{}%
4513 \newcommand*\{LWR@closesix\}{}%
4514 \newcommand*\{LWR@closeseven\}{}%
4515 \newcommand*\{LWR@closeeight\}{}%
4516 \newcommand*\{LWR@closenine\}{}%
4517 \newcommand*\{LWR@closeten\}{}%
4518 \newcommand*\{LWR@closeeleven\}{}%
4519 \newcommand*\{LWR@closetwelve\}{}%
```

41.3 Closing depths

A stack to record the depth of each level:

-  Note that nested L^AT_EX structures may push depths which are non-sequential.

Ex:

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

```
4520 \newcommand*{\LWR@closedepthonen}{\LWR@depthnone}%
4521 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}%
4522 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}%
4523 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}%
4524 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}%
4525 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}%
4526 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}%
4527 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}%
4528 \newcommand*{\LWR@closedeptnine}{\LWR@depthnone}%
4529 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}%
4530 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}%
4531 \newcommand*{\LWR@closedepthtwelve}{\LWR@depthnone}
```

41.4 Pushing and popping the stack

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

Pushes one return action and its L^AT_EX depth onto the stacks.

```
4532 \NewDocumentCommand{\pushclose}{m m}
4533 {
4534 \global\let\LWR@closedepthtwelve\LWR@closeeleven
4535 \global\let\LWR@closeeleven\LWR@closeten
4536 \global\let\LWR@closeten\LWR@closenine
4537 \global\let\LWR@closenine\LWR@closeeight
4538 \global\let\LWR@closeeight\LWR@closeseven
4539 \global\let\LWR@closeseven\LWR@closesix
4540 \global\let\LWR@closesix\LWR@closefive
```

```
4541 \global\let\LWR@closefive\LWR@closefour
4542 \global\let\LWR@closefour\LWR@closethree
4543 \global\let\LWR@closethree\LWR@closetwo
4544 \global\let\LWR@closetwo\LWR@closeone
4545 \global\let\LWR@closeone#1
4546 \global\let\LWR@closedepthtwelve\LWR@closedeptheleven
4547 \global\let\LWR@closedeptheleven\LWR@closedepthten
4548 \global\let\LWR@closedepthten\LWR@closedepthnine
4549 \global\let\LWR@closedepthnine\LWR@closedeptheight
4550 \global\let\LWR@closedeptheight\LWR@closedepthseven
4551 \global\let\LWR@closedepthseven\LWR@closedepthsix
4552 \global\let\LWR@closedepthsix\LWR@closedepthfive
4553 \global\let\LWR@closedepthfive\LWR@closedepthfour
4554 \global\let\LWR@closedepthfour\LWR@closedepththree
4555 \global\let\LWR@closedepththree\LWR@closedepthtwo
4556 \global\let\LWR@closedepthtwo\LWR@closedepthonne
4557 \global\let\LWR@closedepthonne#2
4558 }
```

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

```
4559 \newcommand*{\popclose}
4560 {
4561 \global\let\LWR@closeone\LWR@closetwo
4562 \global\let\LWR@closetwo\LWR@closethree
4563 \global\let\LWR@closethree\LWR@closefour
4564 \global\let\LWR@closefour\LWR@closefive
4565 \global\let\LWR@closefive\LWR@closesix
4566 \global\let\LWR@closesix\LWR@closeseven
4567 \global\let\LWR@closeseven\LWR@closeeight
4568 \global\let\LWR@closeeight\LWR@closenine
4569 \global\let\LWR@closenine\LWR@closeten
4570 \global\let\LWR@closeten\LWR@closeeleven
4571 \global\let\LWR@closeeleven\LWR@closetwelve
4572 \global\let\LWR@closedepthonne\LWR@closedepthtwo
4573 \global\let\LWR@closedepthtwo\LWR@closedepththree
4574 \global\let\LWR@closedepththree\LWR@closedepthfour
4575 \global\let\LWR@closedepthfour\LWR@closedepthfive
4576 \global\let\LWR@closedepthfive\LWR@closedepthsix
4577 \global\let\LWR@closedepthsix\LWR@closedepthseven
4578 \global\let\LWR@closedepthseven\LWR@closedeptheight
4579 \global\let\LWR@closedeptheight\LWR@closedepthnine
4580 \global\let\LWR@closedepthnine\LWR@closedepthten
4581 \global\let\LWR@closedepthten\LWR@closedeptheleven
4582 \global\let\LWR@closedeptheleven\LWR@closedepthtwelve
4583 }

4584 \end{warpHTML}
```

42 Data arrays

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

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

To assign an empty value:

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

for HTML output: 4585 `\begin{warpHTML}`

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

4586 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
4587   \xdef\LWR@thisexparrayname{#1#2}%
4588   \ifstrempty{#3}%
4589     {\csgdef{\LWR@thisexparrayname}{}{}}%
4590     {\csxdef{\LWR@thisexparrayname}{#3}}%
4591 }
```



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

4592 \newcommand*{\LWR@getexparray}[2]{%
4593   \nameuse{#1#2}%
4594 }
```



```
4595 \end{warpHTML}
```

43 Localizing catcodes

for HTML & PRINT: 4596 `\begin{warpall}`

- ⚠ **Misplaced alignment tab character &** Place `\StartDefiningTabulars` and `\StopDefiningTabulars` before and after defining macros or environments which include the tabular & character in their definitions.

The catcode of `&` must be changed before the definitions begin, and must be restored afterwards. Doing so avoids the error

`Misplaced alignment tab character &.`

`\StartDefiningTabulars` Place before defining something with & in it.

```
4597 \newcommand{\StartDefiningTabulars}{%
4598 \LWR@traceinfo{StartDefiningTabulars}%
4599 \warpHTMLonly{\catcode`\&=\active}%
4600 }
```

`\StopDefiningTabulars` Place after defining something with & in it.

```
4601 \newcommand{\StopDefiningTabulars}{%
4602 \LWR@traceinfo{StopDefiningTabulars}%
4603 \warpHTMLonly{\catcode`\&=4}%
4604 }
```

`Bool LWR@mathmacro` True if currently defining math macros. Used to disable SVG math hashing and MathJax math contents while defining a macro using inline math. Begin a macro, it is not guaranteed that the contents are static, and so the image must be unique. The contents also almost certainly will not be parsed correctly by MathJax.

```
4605 \newbool{LWR@mathmacro}
4606 \boolfalse{LWR@mathmacro}
```

`\StartDefiningMath` Place before defining something with \$ in it.

```
4607 \newcommand{\StartDefiningMath}{%
4608 \LWR@traceinfo{StartDefiningMath}%
4609 \warpHTMLonly{\catcode`\$=\active}%
4610 }
```

`\StopDefiningMath` Place after defining something with \$ in it.

```
4611 \newcommand{\StopDefiningMath}{%
4612 \LWR@traceinfo{StopDefiningMath}%
4613 \warpHTMLonly{\catcode`\$=3}% math shift
4614 }

4615 \end{warpall}
```

44 Localizing dynamic math

Inline SVG math usually uses a hash of its contents to generate `lateximages` which are reusable for multiple instances with the same contents. If the contents

may change for each use, such as depending on the current value of a counter, then `\inlinemathother` must be used before the inline math expression, and `\inlinemathnormal` must be used after.

For MathJax, the inline math expression is usually printed for MathJax to interpret. When marked as dynamic math, the following inline math expression will be displayed as an unhashed inline SVG image instead.

For existing code and packages, it may be possible to patch macros after they have been defined, using the `xpatch` package, which is pre-loaded by `lwarp`:

```
\xpatchcmd{\macro}
  {$math expression$}
  {\inlinemathother$math expression$\inlinemathnormal}
  {}
  {\typeout{Error patching macro.}}
```

for HTML & PRINT: 4616 `\begin{warpall}`

`Bool LWR@dynamicmath` True to mark inline math which is dynamic in nature, thus should not be hashed for reuse.
Default: `false`

```
4617 \newbool{LWR@dynamicmath}
4618 \boolfalse{LWR@dynamicmath}
```

`\inlinemathother` Place before using `$... $` or `\(... \)` if the contents of the math are not static, depending on counters or dynamic macros.

```
4619 \newcommand{\inlinemathother}{%
4620 \LWR@traceinfo{\inlinemathother}%
4621 \booltrue{LWR@dynamicmath}%
4622 }
```

`\inlinemathnormal` Place after using `$... $` or `\(... \)` with dynamic contents.

```
4623 \newcommand{\inlinemathnormal}{%
4624 \LWR@traceinfo{\inlinemathnormal}%
4625 \boolfalse{LWR@dynamicmath}%
4626 }
```

```
4627 \end{warpall}
```

45 Sanitizing labels and filenames

Special handling for underscores in labels and filenames.

for HTML output: 4628 \begin{warpHTML}

\LWR@sanitized The sanitized version of what was given to \LWR@sanitize. Characters are set to their detokenized versions. Required for underscores in labels and filenames.

4629 \newcommand*\{\LWR@sanitized\}{}

\LWR@sanitize {\text{}}

Sanitizes the text and returns the result in \LWR@sanitized.

```
4630 \newcommand*\{\LWR@sanitize\}[1]{%
4631 \LWR@traceinfo{\LWR@sanitize: !#1!}%
4632 \edef\LWR@sanitized{\#1}%
4633 \LWR@traceinfo{\LWR@sanitize expanded: !\LWR@sanitized!}%
4634 \edef\LWR@sanitized{\detokenize\expandafter{\LWR@sanitized}}%
4635 \LWR@traceinfo{\LWR@sanitize result: !\LWR@sanitized!}%
4636 }
```

4637 \end{warpHTML}

46 HTML entities

for HTML output: 4638 \begin{warpHTML}

HTML Unicode entities:

4639 \let\LWR@origampersand\&

\HTMLentity {\text{}}

```
4640 \newcommand*\{\HTMLentity\}[1]{%
4641 % \LWR@traceinfo{\HTMLentity \detokenize{\#1}}%
4642 \begingroup%
4643 \LWR@FBcancel%
4644 \LWR@origampersand\#1;%
4645 \endgroup%
4646 % \LWR@traceinfo{\HTMLentity done}%
4647 }
```

```
\HTMLunicode  {\langle hex_unicode\rangle}

4648 \newcommand*{\HTMLunicode}[1]{\HTMLentity{\LWR@origpound{}x#1} }

\&

4649 \renewrobustcmd*{\&}{\HTMLentity{amp} }

\textless
\textgreater
4650 \let\LWR@origtextless\textless
4651 \renewcommand*{\textless}{\HTMLentity{lt} }
4652
4653 \let\LWR@origtextgreater\textgreater
4654 \renewcommand*{\textgreater}{\HTMLentity{gt} }

4655 \end{warpHTML}
```

47 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: 4656 `\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.

4657 `\providecommand*{\BaseJobname}{\jobname}`

`\HTMLFilename` The prefix for all generated HTML files other than the home page, defaulting to empty. See section [8.4.1](#).

4658 `\providecommand*{\HTMLFilename}{}`

`\HomeHTMLFilename` The filename of the home page, defaulting to the `\BaseJobname`. See section [8.4.1](#).

4659 `\providecommand*{\HomeHTMLFilename}{\BaseJobname}`

\SetHTMLFileName {*number*}

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

```
4660 \newcommand*{\SetHTMLFileName}[1]{%
4661 \setcounter{LWR@htmlfilename}{#1}%
4662 }
```

Bool FileSectionNames Selects how to create HTML file names.

Defaults to use section names in the filenames.

```
4663 \newbool{FileSectionNames}
4664 \booltrue{FileSectionNames}

4665 \end{warpall}
```

for HTML output: 4666 \begin{warpHTML}

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

```
4667 \newcounter{LWR@htmlfilename}
4668 \setcounter{LWR@htmlfilename}{0}
```

\LWR@htmlsectionfilename {*htmlfilename or name*}

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

```
4669 \newcommand*{\LWR@htmlsectionfilename}[1]{%
4670 \LWR@traceinfo{LWR@htmlsectionfilename A !\detokenize{#1}!}%
```

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

```
4671 % \LWR@traceinfo{about to assign temp}%
4672 \edef\LWR@tempone{\#1}%
4673 \LWR@traceinfo{about to compare with ??}%
4674 \ifthenelse{\equal{\LWR@tempone}{??}}{%
4675 {\LWR@traceinfo{found ??}}{%
4676 {\LWR@traceinfo{not found ??}}{%
4677 \LWR@traceinfo{about to compare with zero or empty}%
4678 \ifthenelse{%
4679     \equal{\LWR@tempone}{0}}{%
```

```

4680      \OR \equal{\LWR@tempone}{}%
4681      \OR \equal{\LWR@tempone}{??}%
4682 }%
4683 {%
4684     \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
4685     \HomeHTMLFilename.html%
4686 }%

```

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:

```

4687 {%
4688     \LWR@traceinfo{LWR@htmlsectionfilename C \LWR@tempone}%
4689     \ifthenelse{%
4690         \equal{\HTMLFilename}{} \AND
4691         \equal{\LWR@tempone}{Index} \OR
4692         \equal{\LWR@tempone}{index}%
4693     }%
4694     {%
4695         \LWR@traceinfo{Prefixing the index name with an underscore.}%
4696         \_#1.html%
4697     }%

```

Otherwise, create a filename with the chosen prefix:

```

4698     {\HTMLFilename\LWR@isolate{\#1}.html}%
4699 }%
4700 \LWR@traceinfo{LWR@htmlsectionfilename Z}%
4701 }

```

\LWR@htmlrefsectionfilename {⟨label⟩}

Prints the filename for the given label

```

4702 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
4703 \LWR@traceinfo{LWR@htmlrefsectionfilename: !\detokenize{\#1}!}%

```

\LWR@nullfonts to allow math in a section name.

```

4704 \begingroup%
4705 \LWR@nullfonts%
4706 \LWR@htmlsectionfilename{\LWR@htmlfileref{\#1}}%
4707 \endgroup%
4708 \LWR@traceinfo{LWR@htmlrefsectionfilename: done}%
4709 }

4710 \end{warpHTML}

```

48 Homepage link

for HTML & PRINT: 4711 \begin{warpall}

\linkhomename Holds the default name for the home link.

4712 \newcommand{\linkhomename}{Home}

4713 \end{warpall}

for HTML output: 4714 \begin{warpHTML}

\LinkHome May be used wherever you wish to place a link back to the homepage. The filename must be detokenized for underscores.

4715 \newcommand*\LinkHome{%
 4716 \LWR@subhyperrefclass{\HomeHTMLfilename.html}{\linkhomename}{linkhome}%
 4717 }

4718 \end{warpHTML}

for PRINT output: 4719 \begin{warpprint}

\LinkHome May be used wherever you wish to place a link back to the homepage. For print output, if hyperref is available a hyperlink to the first page is used, named by \linkhomename. If hyperref is not available, a pageref is used instead.

4720 \AtBeginDocument{%
 4721 \@ifundefined{hyperref}{%
 4722 \newcommand*\LinkHome{\linkhomename\ --- page \pageref{page-LWRfirstpage}}%
 4723 }{%
 4724 \newcommand*\LinkHome{\hyperref[page-LWRfirstpage]{\linkhomename}}%
 4725 }%
 4726 }%
 4727 \\
 4728 \AfterEndPreamble{\label{page-LWRfirstpage}}

4729 \end{warpprint}

for HTML output: 4730 \begin{warpHTML}

\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.

```
4731 \newcommand*\{LWR@topnavigation}{  
4732 \LWR@htmlelementclassline{nav}{topnavigation}{\LinkHome}  
4733 }
```

`\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.

```
4734 \newcommand*\{LWR@botnavigation}{  
4735 \LWR@htmlelementclassline{nav}{botnavigation}{\LinkHome}  
4736 }  
  
4737 \end{warpHTML}
```

49 \LWRPrintStack diagnostic tool

 Diagnostics tool: Prints the L^AT_EX nesting depth values for the stack levels. `\LWR@startpars` is used before printing the stack, so that `\LWRPrintStack` may be called from anywhere in the normal text flow.

for HTML output: 4738 `\begin{warpHTML}`

`\LWRPrintStack` Prints the closeddepth stack.

```
4739 \newcommand*\{LWR@subprintstack}{  
4740 \LWR@closedepthone\ \LWR@closedepthtwo\ \LWR@closedepththree\   
4741 \LWR@closedepthfour\ \LWR@closedepthfive\ \LWR@closedepthsix\   
4742 \LWR@closedepthseven\ \LWR@closedeptheight\ \LWR@closedepthnine\   
4743 \LWR@closedepthonen\ \LWR@closedeptheleven\ \LWR@closedepthtwelve\   
4744 }  
4745  
4746 \newcommand*\{LWRPrintStack}{  
4747 \LWR@startpars  
4748 \LWR@subprintstack  
4749 }  
  
4750 \end{warpHTML}
```

for PRINT output: 4751 `\begin{warpprint}`

```
4752 \newcommand*\{LWRPrintStack}{}  
  
4753 \end{warpprint}
```

50 Closing stack levels

for HTML output: 4754 \begin{warpHTML}

Close one nested level:

```
4754 \newcommand*{\LWR@closeoneprevious}{%
4756 %
4757 \LWR@closeone
4758 %
4759 \popclose
4760 }
```

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

```
4761 \newcommand*{\LWR@closeprevious}[1]{%
4762 \LWR@traceinfo{\LWR@closeprevious to depth #1, depths are \LWR@subprintstack}%
```

Close any pending paragraph:

```
4763 \LWR@stoppars%
```

Close anything nested deeper than the desired depth. First close anything deeper, then at most one of the same level.

```
4764 \whileboolexpr{test{\ifnumcomp{\LWR@closedepthone}{>}{#1}}}{%
4765 {%
4766     \LWR@traceinfo{\LWR@closeprevious: closing out depth \LWR@closedepthone}%
4767     \LWR@closeoneprevious%
4768 }%
4769 \ifboolexpr{test{\ifnumcomp{\LWR@closedepthone}{=}{#1}}}{%
4770 {%
4771     \LWR@traceinfo{\LWR@closeprevious: closing out depth \LWR@closedepthone}%
4772     \LWR@closeoneprevious%
4773 }{%
4774 \LWR@traceinfo{\LWR@closeprevious: done, depths are \LWR@subprintstack}%
4775 }%
4776 \end{warpHTML}
```

51 PDF pages and styles

for HTML output: 4777 \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.

```
4778 \newcommand{\LWR@forcenewpage}{%
4779 \LWR@traceinfo{\LWR@forcenewpage}%
4780 \ifinner\else%
4781 \LWR@stopars\LWR@orignewpage\LWR@startpars%
4782 \fi%
4783 }
```

`\pagestyle`, etc. are nullified for HTML output.

```
\pagestyle {\langle style\rangle}
4784 \renewcommand*{\pagestyle}[1]{}
```

```
\thispagestyle {\langle style\rangle}
4785 \renewcommand*{\thispagestyle}[1]{}
```

```
\markboth {\langle left\rangle} {\langle right\rangle}
4786 \renewcommand*{\markboth}[2]{}
```

```
\markright {\langle right\rangle}
4787 \renewcommand*{\markright}[1]{}
```

```
\raggedbottom
4788 \renewcommand*{\raggedbottom}{}{}
```

```
\flushbottom
4789 \renewcommand*{\flushbottom}{}{}
```

```
\sloppy
4790 \renewcommand*{\sloppy}{}{}
```

```
\fussy
4791 \renewcommand*{\fussy}{}{}
```

```
\pagenumbering * {<commands>}
4792 \RenewDocumentCommand{\pagenumbering}{s m} {}
4793 \end{warpHTML}
```

52 HTML tags, spans, divs, elements

for HTML output: 4794 \begin{warpHTML}

52.1 Mapping L^AT_EX sections to HTML sections

```
4795 \newcommand*{\LWR@tagtitle}{h1}
4796 \newcommand*{\LWR@tagtitleend}{/h1}
4797 \newcommand*{\LWR@tagpart}{h2}
4798 \newcommand*{\LWR@tagpartend}{/h2}
4799 \newcommand*{\LWR@tagchapter}{h3}
4800 \newcommand*{\LWR@tagchapterend}{/h3}
4801 \newcommand*{\LWR@tagsection}{h4}
4802 \newcommand*{\LWR@tagsectionend}{/h4}
4803 \newcommand*{\LWR@tagsubsection}{h5}
4804 \newcommand*{\LWR@tagsubsectionend}{/h5}
4805 \newcommand*{\LWR@tagsubsubsection}{h6}
4806 \newcommand*{\LWR@tagsubsubsubsection}{/h6}
4807 \newcommand*{\LWR@tagparagraph}{span class="paragraph"}
4808 \newcommand*{\LWR@tagparagraphend}{/span}
4809 \newcommand*{\LWR@tag subparagraph}{span class="subparagraph"}
4810 \newcommand*{\LWR@tag subparagraphend}{/span}
4811
4812 \newcommand*{\LWR@tagregularparagraph}{p}
```

52.2 Babel-French tag modifications

Adjust babel-french for HTML spaces. So far, this only works for *pdflatex* and *xelatex*.

(Emulates or patches code by DANIEL FLIPO.)

```
4813 \providetcommand*{\LWR@FBcancel}{}
4814
4815 \AtBeginDocument{%
4816 \@ifundefined{frenchbsetup}{%
4817 }{%
4818 }}
```

```

4819     \frenchbsetup{FrenchFootnotes=false}%
4820 %
4821     \LetLtxMacro{\LWR@FBcancel}{\NoAutoSpacing}%
4822     \renewrobustcmd*{\FBcolonspace}{%
4823         \begingroup%
4824         \LWR@FBcancel%
4825         \LWR@origampersand{}nbsp;%
4826         \endgroup%
4827     }%
4828     \renewrobustcmd*{\FBthinspace}{%
4829         \begingroup%
4830         \LWR@FBcancel%
4831         \LWR@origampersand{}\LWR@origpound{}x202f;% \
4832         \endgroup%
4833     }%
4834     \renewrobustcmd*{\FBguillspace}{%
4835         \begingroup%
4836         \LWR@FBcancel%
4837         \LWR@origampersand{}nbsp;% ~, for \og xyz \fg{}%
4838         \endgroup%
4839     }%
4840     \DeclareDocumentCommand{\FBmedkern}{.}{%
4841         \begingroup%
4842         \LWR@FBcancel%
4843         \LWR@origampersand{}\LWR@origpound{}x202f;% \
4844         \endgroup%
4845     }%
4846     \DeclareDocumentCommand{\FBthickkern}{.}{%
4847         \begingroup%
4848         \LWR@FBcancel%
4849         \LWR@origampersand{}nbsp;% ~
4850         \endgroup%
4851     }%
4852     \renewrobustcmd*{\~}{\HTMLentity{nbsp}}% was overwritten by babel-french
4853     \ifFBunicode%
4854     \else%
4855         \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}%
4856         \DeclareTextCommandDefault{\FBtextellipsis}{\textellipsis\xspace}%
4857     \fi%
4858 }%
4859 }

```

52.3 HTML tags

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

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

```

4860 \newcommand*{\LWR@htmllagc}[1]{%
4861 \LWR@traceinfo{\LWR@htmllagc !\detokenize{#1}!}%
4862 \begingroup%
4863 \LWR@FBcancel%
4864 \ifmmode\else\protect\LWR@origttfamily\fi%
4865 \protect\LWR@origtextless%
4866 \LWR@isolate{#1}%
4867 \protect\LWR@origtextgreater%
4868 \endgroup%
4869 % \LWR@traceinfo{\LWR@htmllagc: done}%
4870 }

```

Env `LWR@nestspan` Disable `minipage`, `parbox`, and HTML `<div>`s inside a ``.

- ⚠ `\begin{LWR@nestspan}` must follow the opening `` tag to allow a paragraph to start if the span is at the beginning of a new paragraph.
- ⚠ `\end{LWR@nestspan}` must follow the `` or a `<p>` may appear inside the span.

```

4871 \newcommand*{\LWR@nestspanitem}{%
4872 \if@newlist\else{\LWR@htmllagc{br /}}\fi%
4873 \LWR@origitem%
4874 }%
4875
4876 \newenvironment*{\LWR@nestspan}
4877 {%
4878 \LWR@traceinfo{\LWR@nestspan starting}%
4879 \ifnumcomp{\value{\LWR@lateximagedepth}}{>}{0}%
4880 {%
4881     \LWR@traceinfo{\LWR@nestspan: inside a lateximage}%
4882 }%
4883 {%
4884     \LWR@traceinfo{\LWR@nestspan: NOT inside a lateximage}%
4885     \addtocounter{\LWR@spandepth}{1}%
4886     \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{}{%
4887         \RenewDocumentEnvironment{BlockClass}{o m}{}{}%
4888         \ renewcommand{\BlockClassSingle}[2]{##2}%
4889         \ renewcommand{\LWR@forcenewpage}{}%
4890         \ renewcommand{\LWR@liststart}{}{%
4891             \let\item\LWR@nestspanitem%
4892         }%
4893         \ renewcommand{\LWR@listend}{\LWR@htmllagc{br /}\LWR@htmllagc{br /}}%
4894 }%
4895 {%
4896     \LWR@traceinfo{\LWR@nestspan starting: done}%
4897 }%
4898 \LWR@traceinfo{\LWR@nestspan ending}%

```

```

4899 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
4900 {}%
4901 {\addtocounter{LWR@spandepth}{-1}}%
4902 \LWR@traceinfo{LWR@nestspan ending: done}%
4903 }%
4904
4905 \AfterEndEnvironment{LWR@nestspan}{\global\let\par\LWR@closeparagraph}

```

\LWR@htmlspan {*tag*} {*text*}

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

```

4906 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
4907 \LWR@ensuredoingapar%
4908 \LWR@htmlltagc{#1}%
4909 \begin{LWR@nestspan}%
4910 #2%
4911 \LWR@htmlltagc{/#1}%
4912 \end{LWR@nestspan}%
4913 }

```

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

```

4914 \NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{%
4915 \LWR@traceinfo{LWR@htmlspanclass |#1|#2|}%
4916 \LWR@ensuredoingapar%
4917 \LWR@subhtmlelementclass{span}[#1]{#2}%
4918 \begin{LWR@nestspan}%
4919 #3%
4920 \LWR@htmlltagc{/span}%
4921 \LWR@traceinfo{LWR@htmlspanclass done}%
4922 \end{LWR@nestspan}%
4923 }

```

\LWR@htmlltag {*tag*}

Print an HTML tag: <*tag*>

```

4924 \newcommand*{\LWR@htmlltag}[1]{%
4925 % \LWR@traceinfo{LWR@htmlltagb !\detokenize{#1}!}%
4926 \LWR@htmlltagc{#1}%
4927 % \LWR@traceinfo{LWR@htmlltagb: done}%
4928 }

```

52.4 Block tags and comments

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

```
\LWR@htmlopencomment
\LWR@htmclosecomment
4929 \newcommand*{\LWR@htmlopencomment}{%
4930 {%
4931 % \LWR@traceinfo{\LWR@htmlopencomment}%
4932 \begingroup%
4933 \LWR@FBcancel%
4934 \ifmmode\else\protect\LWR@origttfamily\fi%
4935 \LWR@print@mbox{\LWR@origtextless{}!-\/-}%
4936 \endgroup%
4937 }%
4938 }
4939
4940 \newcommand*{\LWR@htmclosecomment}{%
4941 {%
4942 % \LWR@traceinfo{\LWR@htmclosecomment}%
4943 \begingroup%
4944 \LWR@FBcancel%
4945 \ifmmode\else\protect\LWR@origttfamily\fi%
4946 \LWR@print@mbox{~-}\LWR@origtextgreater}%
4947 \endgroup%
4948 }%
4949 }

\LWR@htmlcomment {<comment>}
4950 \newcommand{\LWR@htmlcomment}[1]{%
4951 \LWR@htmlopencomment{}%
4952 {%
4953 \LWR@origttfamily% break ligatures
4954 #1%
4955 }%
4956 \LWR@htmclosecomment{}}

\LWR@htmlblockcomment {<comment>}
4957 \newcommand{\LWR@htmlblockcomment}[1]
4958 {\LWR@stopars \LWR@htmlcomment{\#1}\LWR@startpars}

\LWR@htmlblocktag {<tag>} print a stand-alone HTML tag
```

```

4959 \newcommand*{\LWR@htmlblocktag}[1]{%
4960 \LWR@stoppars%
4961 \LWR@htmntag{#1}%
4962 \LWR@startpars%
4963 }

```

52.5 Div class and element class

\LWR@subhtmlelementclass {*element*} [*style*] {*class*}

Factored and reused in several places.

The trailing spaces allow more places for a line break.

The use of \textquotedbl instead of " provides improved compatibility with xeCJK.

```

4964 \NewDocumentCommand{\LWR@subhtmlelementclass}{m O{} m}{%
4965 \LWR@traceinfo{\LWR@subhtmlelementclass !#1#!#2#!#3!}%
4966 \ifblank{#2}%
4967 {%
4968     \LWR@htmntag{#1 class=\textquotedbl#3\textquotedbl}%
4969 }%
4970 {%
4971     \LWR@htmntag{%
4972         #1 % space
4973         class=\textquotedbl#3\textquotedbl\ % space
4974         style=\textquotedbl#2\textquotedbl%
4975     }%
4976 }%
4977 \LWR@traceinfo{\LWR@subhtmlelementclass done}%
4978 }

```

\LWR@htmlelementclass {*element*} {*class*} [*style*]

```

4979 \NewDocumentCommand{\LWR@htmlelementclass}{m o m}{%
4980 \LWR@stoppars%
4981 \LWR@subhtmlelementclass{#1}[#2]{#3}%
4982 \LWR@startpars%
4983 }

```

\LWR@htmlelementclassend {*element*} {*class*}

```

4984 \newcommand*{\LWR@htmlelementclassend}[2]{%
4985 \LWR@stoppars%
4986 \LWR@htmntag{/#1}%

```

```

4987 \ifbool{HTMLDebugComments}{%
4988     \LWR@htmlcomment{End of #1 ``#2''}%
4989 }{}%
4990 \LWR@startpars%
4991 }

\LWR@htmldivclass  [{style}]{class}%
4992 \NewDocumentCommand{\LWR@htmldivclass}{o m}{%
4993 \LWR@htmlelementclass{div}[#1]{#2}%
4994 }

\LWR@htmldivclassend {class}%
4995 \newcommand*{\LWR@htmldivclassend}[1]{%
4996 \LWR@htmlelementclassend{div}{#1}%
4997 }

```

52.6 Single-line elements

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

```

\LWR@htmlelementclassline {element} [{style}]{class}]{text}%
4998 \NewDocumentCommand{\LWR@htmlelementclassline}{m o m +m}{%
4999 \LWR@stoppars
5000 \LWR@subhtmlelementclass{#1}[#2]{#3}%
5001 #4%
5002 \LWR@htmlltag{/#1}
5003 \LWR@startpars
5004 }

```

52.7 HTML5 semantic elements

```

\LWR@htmlelement {element}%
5005 \newcommand*{\LWR@htmlelement}[1]{%
5006 \LWR@htmlblocktag{#1}%
5007 }

```

```
\LWR@htmlelementend {<element>}

5008 \newcommand*\LWR@htmlelementend[1]{%
5009 \LWR@stoppars
5010 \LWR@htmlltag{/#1}
5011 \LWR@startpars
5012 }
5013
5014 \end{warpHTML}
```

52.8 High-level block and inline classes

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

Nullified versions are provided for print mode.

For other direct-formatting commands, see section 89.

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

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

for PRINT output: 5015 \begin{warpprint}
5016 \NewDocumentEnvironment{BlockClass}{o m}{}{%
5017 \end{warpprint}

for HTML output: 5018 \begin{warpHTML}
5019 \NewDocumentEnvironment{\LWR@print@BlockClass}{o m}{}{%
5020 \NewDocumentEnvironment{\LWR@HTML@BlockClass}{o m}%
5021 {}%
5022 \LWR@origpar%
5023 \LWR@htmldivclass[#1]{#2}%
5024 }%
5025 {\LWR@htmldivclassend{#2}}
5026
5027 \LWR@formattedenv{BlockClass}
5028 \end{warpHTML}

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

for HTML & PRINT: 5029 \begin{warpall}
5030 \newcommand{\BlockClassSingle}[2]{#2}
5031 \end{warpall}

```
for HTML output: 5032 \begin{warpHTML}
5033 \newcommand{\LWR@HTML@BlockClassSingle}[2]{%
5034 \LWR@origpar%
5035 \LWR@htmlelementclassline{div}{#1}{#2}%
5036 }
5037
5038 \LWR@formatted{BlockClassSingle}
5039 \end{warpHTML}
```

\InlineClass ((WP style)) [(style)] {class} {text}

High-level interface for inline span classes.

((WP style)) is css styling to add when formatting for a word processor import.

[(style)] is the css styling to add when not formatting for a word processor.

```
for PRINT output: 5040 \begin{warpprint}
5041 \NewDocumentCommand{\InlineClass}{D{(){}{}} o m +m}{#4}%
5042 \end{warpprint}
```

```
for HTML output: 5043 \begin{warpHTML}
5044 \NewDocumentCommand{\LWR@print@InlineClass}{D{(){}{}} o m +m}{#4}%
5045
5046 \NewDocumentCommand{\LWR@HTML@InlineClass}{D{(){}{}} o m +m}{%
5047 \ifbool{FormatWP}{%
5048   \LWR@htmlspanclass[#1]{#3}{#4}%
5049 }{%
5050   \LWR@htmlspanclass[#2]{#3}{#4}%
5051 }%
5052 }
5053
5054 \LWR@formatted{InlineClass}
5055 \end{warpHTML}
```

Env LWR@BlockClassWP {{WPstyle}} {{HTMLstyle}} {class} Low-level interface for <div> classes with an automatic float ID. These are often used when \ifbool{FormatWP}.

The use of \textquotedbl instead of " provides improved compatibility with xeCJK.

```
for PRINT output: 5056 \begin{warpprint}
5057 \NewDocumentEnvironment{LWR@BlockClassWP}{m m m}{ }{ }%
5058 \end{warpprint}
```

```
for HTML output: 5059 \begin{warpHTML}
5060 \NewDocumentEnvironment{LWR@print@LWR@BlockClassWP}{m m m}{ }{ }%
5061 \NewDocumentEnvironment{LWR@HTML@LWR@BlockClassWP}{m m m}{ }{ }%
5062 {%
```

```

5063 \LWR@stoppars%
5064 \ifbool{FormatWP}%
5065 {%
5066     \addtocounter{LWR@thisautoidWP}{1}%

5067     \LWR@htmlltag{%
5068         div class=\textquotedbl#3\textquotedbl\ % space
5069         id=\textquotedbl%
5070             \LWR@print@mbox{autoidWP-\arabic{LWR@thisautoidWP}}%
5071             \textquotedbl%
5072             \ifblank{}{}{ style=\textquotedbl#1\textquotedbl}%
5073     }%
5074 }% FormatWP
5075 {%
5076     \LWR@htmlltag{%
5077         div class=\textquotedbl#3\textquotedbl%
5078         \ifblank{}{}{ style=\textquotedbl#2\textquotedbl}%
5079     }%
5080 }% not FormatWP
5081 \LWR@startpars%
5082 }
5083 {\LWR@htmldivclassend{#3}}
5084
5085 \LWR@formattedenv{LWR@BlockClassWP}
5086 \end{warpHTML}

```

52.9 Closing HTML tags

for HTML output: 5087 \begin{warpHTML}

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

```

5088 \newcommand*\{\LWR@printclosepart\}
5089     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
5090 \newcommand*\{\LWR@printclosechapter\}
5091     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
5092 \newcommand*\{\LWR@printclosesection\}
5093     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
5094 \newcommand*\{\LWR@printclosesubsection\}
5095     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
5096 \newcommand*\{\LWR@printclosesubsubsection\}
5097     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
5098 \newcommand*\{\LWR@printcloseparagraph\}
5099     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
5100 \newcommand*\{\LWR@printclosesubparagraph\}
5101     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}

```

Lists require closing HTML tags:

```

5102 \newcommand*{\LWR@printcloselistitem}
5103   {\LWR@htmltag{/li}}
5104 \newcommand*{\LWR@printclosedescitem}
5105   {\LWR@htmltag{/dd}}
5106 \newcommand*{\LWR@printcloseitemize}
5107   {\LWR@htmltag{/ul}}
5108 \newcommand*{\LWR@printcloseenumerate}
5109   {\LWR@htmltag{/ol}}
5110 \newcommand*{\LWR@printclosedescription}
5111   {\LWR@htmltag{/dl}}
5112 \end{warpHTML}

```

53 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 54 has high-level commands which allow paragraph-tag generation to start/stop. Even when allowed (`\LWR@doingstartpars`), tags are not generated until a L^AT_EX paragraph is being used (`\LWR@doingapar`). `\LWR@lateximagedepth` is used to prevent nesting tags inside a `\teximage`. `\LWR@spandepth` is used to prevent nesting paragraph tags inside a paragraph, which became important inside `\fbox` commands and other spans.

for HTML output: 5113 `\begin{warpHTML}`

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

```

5114 \newcounter{\LWR@spandepth}
5115 \setcounter{\LWR@spandepth}{0}

```

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

```

5116 \newbool{\LWR@doingstartpars}
5117 \boolfalse{\LWR@doingstartpars}

```

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

```

5118 \newbool{\LWR@doingapar}
5119 \global\boolfalse{\LWR@doingapar}

```

\LWR@ensuredoingapar If are about to print something visible, and if allowed to start a new paragraph, ensure that are \LWR@doingapar, so that paragraph tags are placed:

```
5120 \newcommand*\LWR@ensuredoingapar{%
5121 \ifbool{\LWR@doingstartpars}{%
5122 {\global\booltrue{\LWR@doingapar}}}{%
5123 {}}%
5124 }
```

\PN@parnotes@auto Redefined by parnotes to print paragraph notes at the end of each paragraph.

```
5125 \def\PN@parnotes@auto{}%
```

\LWR@openparagraph

```
5126 \newcommand*\LWR@openparagraph{%
5127 {}%
```

See if paragraph handling is enabled:

```
5128 \ifbool{\LWR@doingstartpars}{%
5129 {}% handling pars}
```

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

```
5130 \ifboolexpr{%
5131     test {\ifnumcomp{\value{\LWR@lateximagedepth}}{>}{0}} or%
5132     test {\ifnumcomp{\value{\LWR@spandepth}}{>}{0}}%
5133 }% 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 ``.

```
5134 {}% no nested par tags
```

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

```
5135 {}% yes nest par tags
```

If `parnotes` is used, paragraph notes are inserted before starting the next paragraph:

```
5136 \PN@parnotes@auto%
```

The opening paragraph tag:

```
5137 \LWR@htmlltagc{\LWR@tagregularparagraph}%
```

Now have started a paragraph.

```
5138 \global\booltrue{LWR@doingapar}%
```

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

```
5139 \let\par\LWR@closeparagraph%
5140 }% end of yes nest par tags
5141 }% end of handling pars
5142 {}% not handling pars
5143 }
```

\LWR@closeparagraph

```
5144 \newcommand*{\LWR@closeparagraph}
5145 {%
5146 % \LWR@traceinfo{LWR@closeparagraph}%
```

See if paragraph handling is enabled:

```
5147 \ifbool{LWR@doingapar}{}%
```

If currently in paragraph mode:

```
5148 {}% handling pars
```

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

```
5149 \ifboolexpr{
5150     test {\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}} or
5151     test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}}
5152 }%
```

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

```
5153 {}% no nested par tags
5154 \ifboolexpr{
5155     test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}} and
5156     test {\ifnumcomp{\value{LWR@lateximagedepth}}{=}{0}}
5157 }%
```

```

5158     {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}}%
5159     {}%
5160 }% no nested par tags

```

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

```
5161   {\% yes nest par tags}
```

Print a closing tag and some extra vertical space.

```

5162   \unskip%
5163   \LWR@htmltagc{/}\LWR@tagregularparagraph}%
5164   \LWR@orignewline%

```

No longer doing a paragraph:

```
5165   \global\boolfalse{LWR@doingapar}%
```

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

```
5166   \global\boolfalse{LWR@minipagethispar}%
```

If `parnotes` is used, paragraph notes are inserted after ending the previous paragraph:

```

5167   \PN@parnotes@auto%
5168 }% end of yes nest par tags
5169 }% end of handling pars

```

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

```

5170 {\% not handling pars
5171   \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
5172   {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}}%
5173   {}%
5174 }% not handling pars

```

In most cases, finish with a L^AT_EX `\par`, but in the case of paragraphs between lines in a tabular fetch the next token instead:

```

5175 \ifboolexpr{%
5176   not bool {LWR@doingapar} and
5177   test {\ifnumcomp{\value{LWR@tabulardepth}}{>}{0}} and
5178   test {
5179     \ifnumcomp{\value{LWR@tabulardepth}}{=}{\value{LWR@tabularpardepth}}}
5180   } and
5181   bool {LWR@intabularmetadata} and
5182   not bool {LWR@tableparcell} and

```

```

5183     test {\ifnumcomp{\value{LWR@lateximagedepth}}{=}{0}}
5184 }%
5185 {%
5186     \LWR@getmynexttoken%
5187 }{%
5188     \LWR@origpar%
5189 }%
5190 }

5191 \end{warpHTML}

```

54 Paragraph start/stop handling

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

Section 53 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: 5192 `\begin{warpHTML}`

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

```

5193 \newcommand*{\LWR@startpars}{%
5194 {%
5195 % \LWR@traceinfo{\LWR@startpars}%

```

Ignore if inside a span:

```

5196 \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
5197 {}%
5198 {}

```

See if currently handling HTML paragraphs:

```

5199 \ifbool{\LWR@doingstartpars}{%

```

If already in paragraph mode, do nothing.

```

5200 {}%

```

If not currently in paragraph mode:

```
5201      {%
```

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

```
5202      \PushPreHook{par}{\LWR@openparagraph}%
```

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

```
5203      \let\par\LWR@closeparagraph
5204
5205      }% an intentionally blank line
```

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

```
5206      \global\setbool{\LWR@doingstartpars}{true}%
```

No <par> tag yet to undo:

```
5207      \global\boolfalse{\LWR@doingapar}%
5208 }% nestspan
5209 % \LWR@traceinfo{\LWR@startpars: done}%
5210 }
```

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

```
5211 \newcommand*{\LWR@stoppars}%
5212 {%
```

Ignore if inside a span:

```
5213 \ifnumcomp{\value{\LWR@spandepth}}{>}{0}%
5214 {}%
5215 {%
```

See if currently handling HTML paragraphs:

```
5216 \ifbool{\LWR@doingapar}{}
```

if currently in an HTML paragraph:

```
5217 {%
```

Print a closing tag:

```
5218 \unskip%
```

```
5219      \LWR@htmntagc{/\LWR@tagregularparagraph}%
5220      \LWR@orignewline%
```

No longer have an open HTML paragraph:

```
5221      \global\boolfalse{LWR@doingapar}%
```

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

```
5222      \global\boolfalse{LWR@minipagethispar}
5223
5224  }% an intentionally blank line
```

If was not in an HTML paragraph:

```
5225  {}%
```

See if currently allowing HTML paragraphs:

```
5226  \ifbool{LWR@doingstartpars}{}
```

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

```
5227  {\ClearPreHook{par}}%
```

Else: Do nothing:

```
5228  {}%
```

No longer in paragraph mode:

```
5229  \global\setbool{LWR@doingstartpars}{false}%
```

No `<p>` tag to undo:

```
5230  \global\boolfalse{LWR@doingapar}%
5231 }% nestspan
5232 }
```

```
5233 \end{warpHTML}
```

55 Page headers and footers

for HTML & PRINT: 5234 `\begin{warpall}`

In the following, catcode is manually changed back and forth without groups, since new macros are being defined which must not be contained within the groups.

```
5235 \newcommand{\LWR@firstpagetop}{} % for the home page alone
5236 \newcommand{\LWR@pagetop}{} % for all other pages
5237 \newcommand{\LWR@pagebottom}{}%
```

\HTMLFirstPageTop {*<text and logos>*}

```
5238 \newcommand{\HTMLFirstPageTop}[1]{%
5239     \renewcommand{\LWR@firstpagetop}{#1}%
5240 }
```

\HTMLPageTop {*<text and logos>*}

```
5241 \newcommand{\HTMLPageTop}[1]{%
5242     \renewcommand{\LWR@pagetop}{#1}%
5243 }
```

\HTMLPageBottom {*<text and logos>*}

```
5244 \newcommand{\HTMLPageBottom}[1]{%
5245     \renewcommand{\LWR@pagebottom}{#1}%
5246 }
```

```
5247 \end{warpall}
```

56 CSS

for HTML output: 5248 \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.

```
5249 \newcommand*{\LWR@currentcss}{lwarp.css}
```

\CSSFilename {*<new-css-filename.css>*} Assigns the css file to be used by the following HTML pages.

```
5250 \newcommand*{\CSSFilename}[1]{%
5251 \renewcommand*{\LWR@currentcss}{#1}%
5252 }
```

```

5252 \@onelvel@sanitize\LWR@currentcss%
5253 }
5254
5255 \end{warpHTML}

for PRINT output: 5256 \begin{warpprint}
5257 \newcommand*{\CSSfilename}[1]{}
5258 \end{warpprint}

```

57 Title, HTML meta author, HTML meta description

for HTML output: 5259 \begin{warpHTML}

\title {\<title>} Modified to remember \thetitle, which is used to set the HTML page titles.

```

5260 \let\LWR@origtitle\title
5261
5262 \renewcommand*{\title}[1]{%
5263   \LWR@origtitle{#1}%
5264   \begingroup%
5265     \renewcommand{\thanks}[1]{%
5266       \protected@xdef\thetitle{#1}%
5267     \endgroup%
5268 }
5269 \end{warpHTML}

```

for HTML & PRINT: 5270 \begin{warpall}

\HTMLTitle {\<Titlename>} The Title to place into an HTML meta tag. The default is to use the document \title's setting.

```

5271 \providecommand{\thetitle}{}
5272
5273 \newcommand{\theHTMLTitle}{\thetitle}
5274
5275 \newcommand{\HTMLTitle}[1]{\renewcommand{\theHTMLTitle}{#1}}

```

\HTMLAuthor {\<authorname>} The author to place into an HTML meta tag. If none given, the default is \theauthor, which is empty unless the titling package is used.

```

5276 \providecommand{\theauthor}{}
5277
5278 \newcommand{\theHTMLAuthor}{\theauthor}
5279
5280 \newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}

```

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

- ⚠ **HTML author** Do not use double quotes, and do not exceed 150 characters.

\HTMLDescription {*New HTML meta description.*} Assigns the HTML file's description meta tag.

```

5281 \newcommand{\LWR@currentHTMLDescription}{}
5282
5283 \newcommand{\HTMLDescription}[1]{%
5284 \renewcommand{\LWR@currentHTMLDescription}{#1}%
5285 }
5286
5287 \end{warpall}

```

58 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 mostly work as-is.

The footmisc stable option is emulated by lwarp.

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

```

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

```

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

```

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

```

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

Several kinds of footnotes are used: in a regular page, in a minipage, or as thanks in the titlepage. Each of these is handled differently.

58.1 Regular page footnotes

In HTML documents, footnotes are placed at the bottom of the web page or the section, depending on `FootnoteDepth`, using the 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 58.4 for the implementation.

58.2 Minipage footnotes

See section 58.5 for how minipage footnotes are gathered. See section 88.3 for how minipage footnotes are placed into the document.

58.3 Titlepage thanks

See section 65.7 for titlepage footnotes.

58.4 Regular page footnote implementation

for HTML & PRINT: 5288 `\begin{warpall}`

Ctr `FootnoteDepth` Determines how deeply to place footnotes in the HTML files, similar to `tocdepth`.
Default: 3 The default of 3 places footnotes before each `\subsubsection` or higher. See table 9 for a table of L^AT_EX section headings.

5289 `\newcounter{FootnoteDepth}`
5290 `\setcounter{FootnoteDepth}{3}`

5291 `\end{warpall}`

for HTML output: 5292 `\begin{warpHTML}`

Patch L^AT_EX footnotes to use a new \box instead of an insert for l warp footnotes. This avoids having the original \footins appear at the bottom of a lateximage, which is on its own new page.

```
5293 \newbox\LWR@footnotes
```

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

```
\@makefntext {<text>}
```

```
5294 \long\def\@makefntext#1{\textsuperscript{\@thefnmark}~#1}
```

```
\@makefnmark
```

```
5295 \def\@makefnmark{%
5296   \textsuperscript{\@thefnmark}%
5297 }
```

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

In a lateximage during HTML output, the lateximage is placed inside a print-mode minipage, but the footnotes are broken out by:

```
\def\@mpfn{footnote}
\def\thempfn{\thefootnote}
\let\@footnotetext\LWR@footnotetext
```

```
\LWR@footnotetext {<text>}
```

```
5298 \long\def\LWR@footnotetext#1{%
5299 \LWR@traceinfo{\LWR@footnotetext}%
5300 \global\setbox\LWR@footnotes=\vbox{%
```

Add to any current footnotes:

```
5301 \unvbox\LWR@footnotes%
```

Remember the footnote number for \ref:

```
5302 \protected@edef\@currentlabel{%
5303   \csname p@footnote\endcsname\@thefnmark%
5304 }% @currentlabel
```

Open a group:

```
5305 \color@begingroup%
```

Use HTML superscripts in the footnote even inside a `lateximage`:

```
5306 \renewrobustcmd{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

Use paragraph tags if in a tabular data cell or a `lateximage`:

```
5307 \ifthenelse{%
5308   \boolean{LWR@doingstartpars} \AND%
5309   \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
5310 }%
5311 {}%
5312 {\LWR@htmlltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
5313 \@makefntext{#1}%
```

Closing paragraph tag:

```
5314 \ifthenelse{%
5315   \boolean{LWR@doingstartpars} \AND%
5316   \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
5317 }%
5318 {\par}%
5319 {}%
5320 \LWR@htmlltagc{/}\LWR@tagregularparagraph\%
5321 \LWR@orignewline\%
5322 }%
```

Close the group:

```
5323 \color@endgroup%
5324 }% vbox
```

Paragraph handling:

```
5325 \LWR@ensuredoingapar%
5326 }%
```

```
\@footnotetext {<text>}
```

```
5327 \LetLtxMacro{\footnotetext}{\LWR@footnotetext}
```

58.5 Minipage footnote implementation

Patch L^AT_EX minipage footnotes to use a new \box instead of an insert for lwarp minipage footnotes. This avoids having the original \Cmpfootins appear at the bottom of a lateximage, which is on its own new page.

```

5328 \newbox\LWR@mpfootnotes

\Cmpfootnotetext {⟨text⟩}

5329 \long\def\Cmpfootnotetext#1{%
5330 \LWR@traceinfo{@mpfootnotetext}%
5331 \global\setbox\LWR@mpfootnotes\vbox{%
5332   \unvbox\LWR@mpfootnotes%
5333   \reset@font\footnotesize%
5334   \hsize\columnwidth%
5335   \parboxrestore%
5336   \protected@edef\@currentlabel{%
5337     \csname p@mpfootnote\endcsname\@thefnmark}%
5338   \color@begingroup%

```

Use paragraph tags if in a tabular data cell or a lateximage:

```

5339 \ifthenelse{%
5340   \boolean{\LWR@doingstartpars} \AND%
5341   \cnttest{\value{\LWR@lateximagedepth}}{=}{0}%
5342 }%
5343 {}%
5344 {\LWR@htmllagc{\LWR@tagregularparagraph}}%

5345 \makefntext{%
5346   \ignorespaces#1%
5347 }%

```

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

```

5348 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}{>}{0} }{%
5349 }%
5350 {}%
5351 \LWR@htmllagc{/LWR@tagregularparagraph}%
5352 \LWR@orignewline%
5353 }%
5354 \color@endgroup%
5355 }% vbox

```

Paragraph handling:

```
5356 \LWR@ensuredoingapar%
5357 \LWR@traceinfo{@mpfootnotetext: done}%
5358 }
```

\thempfootnote Redefined to remove the \itshape, which caused an obscure compiling error in some situations.

```
5359 \AtBeginDocument{%
5360 \def\thempfootnote{\alph{c}\mpfootnote}%
5361 }
```

58.6 Printing pending footnotes

\LWR@printpendingfootnotes Enclose the footnotes in a class, print, then clear.

```
5362 \newcommand*{\LWR@printpendingfootnotes}{%
5363 \ifvoid\LWR@footnotes\else
5364     \LWR@forcenewpage
5365     \begin{BlockClass}{footnotes}
5366     \LWR@origmedskip
5367     \unvbox\LWR@footnotes
5368     \setbox\LWR@footnotes=\vbox{}
5369     \end{BlockClass}
5370 \fi
5371 }
```

\LWR@maybeprintpendingfootnotes {<depth>} Used to print footnotes before sections only if formatting for an EPUB or word processor:

```
5372 \newcommand*{\LWR@maybeprintpendingfootnotes}[1]{%
5373 \ifboolexpr{
5374     not test{\ifnumcomp{#1}{>}{\value{FootnoteDepth}}} or
5375     bool{FormatEPUB} or
5376     bool{FormatWP}
5377 }%
5378 {\LWR@printpendingfootnotes}%
5379 {}%
5380 }
```

\LWR@printpendingmpfootnotes Enclose the minipage footnotes in a class, print, then clear.

```
5381 \newcommand*{\LWR@printpendingmpfootnotes}{%
5382 \ifvoid\LWR@mpfootnotes\else
5383     \LWR@forcenewpage
```

```

5384     \begin{BlockClass}{footnotes}
5385     \LWR@print@vspace*\{\baselineskip}
5386     \unvbox\LWR@mpfootnotes
5387     \setbox\LWR@mpfootnotes=\vbox{}
5388     \end{BlockClass}
5389 \fi
5390 }

5391 \end{warpHTML}

```

59 Marginpars

\marginpar [*<left>*] [*<right>*] \marginpar may contains paragraphs, but in order to remain inline with the surrounding text lwarp nullifies block-related macros inside the \marginpar. Paragraph breaks are converted to
 tags.

\marginparBlock [*<left>*] [*<right>*] To include block-related macros, use \marginparBlock, which takes the same arguments but creates a <div> instead of a . A line break will occur in the text where the \marginBlock occurs.

for HTML output: 5392 \begin{warpHTML}

\marginpar [*<left>*] [*<right>*]

```

5393 \renewcommand{\marginpar}[2] [] {%
5394 \ifbool{FormatWP}{%
5395 {%
5396 \begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{}{marginblock}%
5397 #2
5398 \end{LWR@BlockClassWP}%
5399 }%
5400 {%
5401     \LWR@htmlspanclass{marginpar}{#2}%
5402 }%
5403 }

```

\marginparBlock [*<left>*] [*<right>*]

For use when the marginpar will be more than one paragraph, and/or contains more than simple text.

HTML version.

```
5404 \newcommand{\marginparBlock}[2] [] {%
```

```
5405 \ifbool{FormatWP}{%
5406 {%
5407 \begin{LWR@BlockClassWP}[width:2in; float:right; margin:10pt]{}
5408 #2
5409 \end{LWR@BlockClassWP}
5410 }%
5411 {%
5412 \begin{BlockClass}[width:2in; float:right; margin:10pt]{marginparblock}
5413 #2
5414 \end{BlockClass}
5415 }%
5416 }

\reversemarginpar
5417 \renewcommand*{\reversemarginpar}{}}

\normalmarginpar
5418 \renewcommand*{\normalmarginpar}{}}

5419 \end{warpHTML}

for PRINT output: 5420 \begin{warpprint}

\marginparBlock  [\langle left\rangle] \{\langle right\rangle\}

For use when the marginpar will be more than one paragraph, and/or contains more
than simple text.

Print version.

5421 \LetLtxMacro{\marginparBlock}{\marginpar}

5422 \end{warpprint}
```

60 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: 5423 \begin{warpall}

Ctr FileDepth {*section depth*} determines how deeply to break into new HTML files, similar to tocdepth. The default of -5 produces one large HTML file.

```
5424 \newcounter{FileDepth}
5425 \setcounter{FileDepth}{-5}
```

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

```
5426 \newbool{CombineHigherDepths}
5427 \booltrue{CombineHigherDepths}
```

```
5428 \end{warpall}
```

for HTML output: 5429 \begin{warpHTML}

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

```
5430 \newcommand*{\LWR@thisfilename}{}%
```

\LWR@thisnewfilename The filename being sanitized.

```
5431 \newcommand*{\LWR@thisnewfilename}{}%
```

\LWR@filenamenoblocks {*filename*}

Convert blanks into dashes, removes short words, store result in \LWR@thisfilename.

⚠ duplicate filename Be sure that this does not result in duplicate filenames! Use the optional toc caption entry parameter for formatting. Remember to \protect L^AT_EX commands which appear in section names and toc captions.

```
5432 \newcommand*{\LWR@filenamenoblocks}[1]{%
5433 \begingroup
```

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

```
5434 \LWR@nullfonts%
5435 \renewcommand*{\LWR@htmlltagc}[1]{}%
```

Replaces common symbols and short words with hyphens:

```
5436 \edef\LWR@thisnewfilename{\#1}%
5437 \LWR@traceinfo{\LWR@filenamenoblocks \edef: !\LWR@thisnewfilename!}%
5438 \fullexpandarg%
```

Convert spaces into hyphens:

```
5439 \StrSubstitute{\LWR@thisnewfilename}{\ }{-}[\LWR@thisnewfilename]
5440 \StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctuation into hyphens:

```
5441 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
5442 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
5443 \StrSubstitute{\LWR@thisnewfilename}%
5444     {\LWR@origampersand}{-}[\LWR@thisnewfilename]
5445 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
5446 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
5447 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
5448 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
5449 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
5450 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
5451 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
5452 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
5453 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
5454 \StrSubstitute{\LWR@thisnewfilename}%
5455     {\textless}{-}[\LWR@thisnewfilename]
5456 \StrSubstitute{\LWR@thisnewfilename}%
5457     {\textgreater}{-}[\LWR@thisnewfilename]
5458 \StrSubstitute{\LWR@thisnewfilename}{\LWR@origpound}{-}[\LWR@thisnewfilename]

5459 \StrSubstitute{\LWR@thisnewfilename}{\_}{-}[\LWR@thisnewfilename]
5460 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
5461 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
5462 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
5463 \StrSubstitute{\LWR@thisnewfilename}{\|}{-}[\LWR@thisnewfilename]
5464 \StrSubstitute{\LWR@thisnewfilename}%
5465     {\textbackslash}{-}[\LWR@thisnewfilename]
5466 \StrSubstitute{\LWR@thisnewfilename}{^}{-}[\LWR@thisnewfilename]
5467 \StrSubstitute{\LWR@thisnewfilename}{\~}{-}[\LWR@thisnewfilename]
5468 \StrSubstitute{\LWR@thisnewfilename}{\~\{}}{-}[\LWR@thisnewfilename]
5469 %      "\~{}" for babel
5470 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
5471 \StrSubstitute{\LWR@thisnewfilename}{\}]{-}[\LWR@thisnewfilename]
5472 \StrSubstitute{\LWR@thisnewfilename}{`}{-}[\LWR@thisnewfilename]
```

Convert short words:

```
5473 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
5474 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
5475 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
5476 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
5477 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
```

```

5478 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
5479 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
5480 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
5481 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
5482 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
5483 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
5484 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
5485 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
5486 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
5487 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
5488 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
5489 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
5490 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

5491 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
5492 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
5493 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
5494 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]

```

If pdf \LaTeX and not utf8 encoding, don't try to convert emdash, endash:

```

5495 \ifPDFTeX% pdflatex or dvi latex
5496 \ifdefstring{\inputencodingname}{utf8}{%
5497 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
5498 %      emdash
5499 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
5500 %      endash
5501 }{}%
5502 \else% not PDFTeX
5503 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
5504 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
5505 \fi%

```

Return the global result:

```

5506 \global\let\LWR@thisfilename\LWR@thisnewfilename%
5507 \endgroup%
5508 \LWR@traceinfo{LWR@filenamenoblanks: result is \LWR@thisfilename}%
5509 }

```

`LWR@previousautopagelabel` Ctr Remembers which autopage label was most recently generated. Used to avoid duplicates.

```

5510 \newcounter{LWR@previousautopagelabel}
5511 \setcounter{LWR@previousautopagelabel}{-1}

```

```
\LWR@newautopagelabel {<page number counter>}

5512 \newcommand*{\LWR@newautopagelabel}[1]{%
5513 \ifnumequal{\value{\LWR@previousautopagelabel}}{\value{page}}{%
5514 {}% no action if this autopage label has already been defined
5515 {}%
5516 \label{autopage-\arabic{#1}}%
5517 \setcounter{\LWR@previousautopagelabel}{\value{page}}%
5518 }%
5519 }
```

60.1 Customizing MATHJAX

\LWR@customizedMathJax Additional MATHJAX definitions to be added to the start of each HTML page.

```
5520 \newcommand*{\LWR@customizedMathJax}{}%
```

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

Examples:

```
\CustomizeMathJax{
    \newcommand{\expval}[1]{\langle #1 \rangle}
    \newcommand{\abs}[1]{\lvert #1 \rvert}
}

\CustomizeMathJax{\newcommand{\arsinh}{\text{arsinh}}}
\CustomizeMathJax{\newcommand{\arcosh}{\text{arcosh}}}
\CustomizeMathJax{\newcommand{\NN}{\mathbb{N}}}

5521 \newcommand*{\CustomizeMathJax}[1]{%
5522     \appto{\LWR@customizedMathJax}{%
5523         \(#1\)\par
5524     }%
5525 }
```

\LWR@customizeMathJax

```
5526 \newcommand{\LWR@customizeMathJax}{}%
5527 \ifbool{mathjax}{%
    5528 \LWR@stopars
    5529 \LWR@htmlcomment{Nullify \textbackslash ensuremath, footnotes for MathJax:}
    5530 }
```

```
5531 \\newcommand\\ensuremath[1]{##1}\\)
5532
5533 \\newcommand\\footnote[2][]{\\text{( Footnote ##1 )}}\\)
5534
5535 \\newcommand\\footnotemark[1][]{\\text{( Footnote ##1 )}}\\)
5536
5537 \\LWR@htmlcomment{Additional customizations for MathJax:\\}
5538
5539 \\LWR@customizedMathJax
5540
5541 \\LWR@startpars
5542 }\\}
5543 }

5544 \\end{warpHTML}
```

for PRINT output: 5545 \\begin{warpprint}

\\CustomizeMathJax The print-mode version:

```
5546 \\newcommand*{\\CustomizeMathJax}[1]{}
5547 \\end{warpprint}
```

for HTML output: 5548 \\begin{warpHTML}

\\LWR@newhtmlfile {*section name*}

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

```
5549 \\newcommand*{\\LWR@newhtmlfile}[1]{
5550 \\LWR@traceinfo{LWR@newhtmlfile}}
```

At the bottom of the ending file:

```
5551 \\LWR@htmlelementclassend{section}{textbody}
5552 \\LWR@htmlelementclassend{div}{bodycontainer}
5553 \\LWR@htmlelementclassend{div}{bodyandsidetoc}
5554
5555 \\LWR@printpendingfootnotes
5556
```

No footer between files if EPUB:

```
5557 \ifbool{FormatEPUB}{  
5558 {}  
5559 {  
5560     \LWR@htmlelement{footer}  
5561     \LWR@pagebottom  
5563     \LWR@htmlelementend{footer}  
5565 }}
```

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

```
5566 \ifthenelse{\boolean{FormatEPUB} \OR \boolean{FormatWP}}{  
5567 {}  
5568 {\ifnumcomp{\value{LWR@htmlfilename}}{>}{0}{\LWR@botnavigation}{}}}
```

End of this HTML file:

```
5569 \LWR@stopars  
5570 \LWR@htmlelement{/body}\LWR@orignewline  
5571 \LWR@htmlelement{/html}\LWR@orignewline  
5572 \LWR@traceinfo{LWR@newhtmlfile: about to LWR@orignewpage}  
5573 \LWR@orignewpage  
5574  
5575 \addtocounter{LWR@htmlfilename}{1}%
```

If using a filename based on section name, create a version without blanks. The filename without blanks will be placed into \LWR@thisfilename. Duplicates will be detected using MD5 hashes.

If not using a filename, the file number will be used instead.

```
5576 \ifbool{FileSectionNames}{%  
5577 {%
```

Convert the section name to a filename with blanks and common words removed. The resulting filename is in \LWR@thisfilename.

```
5578     \LWR@filenamenoblanks{#1}%
```

Create a macro name from the MD5 hash of the file name.

```
5579     \edef\LWR@hashedname{\LWR@mdfive{\LWR@thisfilename}}%
```

If the macro name is not yet defined, this filename is unique.

```
5580     \ifcsundef{\LWR@filename\LWR@hashedname}{%
```

If the filename is unique, create a macro using the hashed name, to be used to test for additional duplicates in the future.

```
5581 \csdef{\LWR@filename\LWR@hashedname}{}%
5582 }%
```

If the filename is not unique, create an error.

```
5583 \PackageError{lwarp}
5584 {
5585     Section name '#1', at the line number listed below,
5586     generates the filename '\LWR@thisfilename',
5587     which appears to be a duplicate%
5588 }
5589 {
5590     Look for another section of the same type with a similar
5591     or identical name.
5592     (Lwarp sanitizes most symbols and common short words
5593     when generating file names, and this may cause a conflict.)
5594 }
5595 }
5596 }
```

If using file numbers instead of names, the name is set to the next file number.

```
5597 {\renewcommand*{\LWR@thisfilename}{\arabic{\LWR@htmldocument}}}
```

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.

```
5598 \LWR@traceinfo{\LWR@newhtmldocument: about to print start file}%
```

`\LWR@nullfonts` to allow math in a section name.

```
5599 \begingroup%
5600 \LWR@nullfonts%
5601 \LWR@htmlblockcomment{%
5602 |Start file|%
5603 \LWR@htmldocumentfilename{\LWR@thisfilename}|%
5604 }%
5605 \endgroup%
```

At the top of the starting file:

```
5606 \LWR@stopars
5607
```

Start a new file with the given section name:

```
5608 \LWR@filestart[#1]  
5609
```

Track the page numbers:

```
5610 \setcounter{LWR@latestautopage}{\value{page}}%  
5611 \LWR@newautopagelabel{LWR@latestautopage}%
```

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

```
5612 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}{}  
5613 {}  
5614 {\LWR@topnavigation}  
5615
```

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

```
5616 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}{}  
5617 {}  
5618 {  
5619     \LWR@htmlelement{header}  
5620  
5621     \LWR@pagetop  
5622  
5623     \LWR@htmlelementend{header}  
5624 }  
5625
```

The container for the sidetoc and text body:

```
5626 \LWR@htmlelementclass{div}{bodyandsidetoc}
```

No sidetoc if formatting for an EPUB or word processor:

```
5627 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}{}  
5628 {}  
5629 {\LWR@sidetoc}  
5630
```

Start of the <textbody>:

```
5631 \LWR@htmlelementclass{div}{bodycontainer}  
5632 \LWR@htmlelementclass{section}{textbody}  
5633
```

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

```
5634 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}%
5635     {}%
5636     {}%
5637     \ifcsvoid{\thetitle}{}{%
5638         \LWR@printthetitle%
5639     }%
5640 }
```

Keep paragraph tags disabled for now:

```
5641 \LWR@stopars
5642
```

If using MATHJAX, disable `\ensuremath` by printing a nullified definition at the start of each file, and add further customizations:

```
5643 \LWR@customizeMathJax
5644 \LWR@traceinfo{\LWR@newhtmlfile: done}
5645 }

5646 \end{warpHTML}
```

61 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^EL^AT_EX and LuaL^AT_EX directly support accented section and file names.

for HTML output: 5647 \begin{warpHTML}

61.1 User-level starred section commands

\ForceHTMLPage For HTML output, forces the next section to be on its own HTML page, if FileDepth allows, even if starred. For use with \printindex and others which generate a starred section which should be on its own HTML page. Also see \ForceHTMLTOC.

For print output, no effect.

```
5648 \newbool{LWR@forcinghtmlpage}
5649 \boolfalse{LWR@forcinghtmlpage}
5650
5651 \newcommand*\{\ForceHTMLPage}{%
5652 \global\booltrue{LWR@forcinghtmlpage}%
5653 }
```

\ForceHTMLTOC For HTML output, forces the next section to have a TOC entry, even if starred. For use with \printindex and others which generate a starred section which should be in the TOC so that it may be accessed via HTML. Not necessary if used with tocbibind. Also see \ForceHTMLPage.

For print output, no effect.

```
5654 \newbool{LWR@forcinghtmltoc}
5655 \boolfalse{LWR@forcinghtmltoc}
5656
5657 \newcommand*\{\ForceHTMLTOC}{%
5658 \global\booltrue{LWR@forcinghtmltoc}%
5659 }
```

```
5660 \end{warpHTML}
```

for PRINT output: 5661 \begin{warpprint}
5662 \newcommand*\{\ForceHTMLPage}{}
5663 \newcommand*\{\ForceHTMLTOC}{}
5664 \end{warpprint}

for HTML output: 5665 \begin{warpHTML}

61.2 Book class commands

\mainmatter Declare the main matter section of the document. Does not reset the page number, which must be consecutive arabic numbers for the HTML conversion.

```
5666 \newbool{LWR@mainmatter}
```

```
5667 \DeclareDocumentCommand{\mainmatter}{}{%
5668 \booltrue{LWR@mainmatter}%
5669 }
```

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

```
5670 \DeclareDocumentCommand{\frontmatter}{}{%
5671 \boolfalse{LWR@mainmatter}%
5672 }
```

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

```
5673 \DeclareDocumentCommand{\backmatter}{}{%
5674 \boolfalse{LWR@mainmatter}%
5675 }
```

61.3 Sectioning support macros

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

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

```
5676 \newcommand*{\LWR@sectionnumber}[1]{%
5677 \InlineClass{sectionnumber}{#1}%
5678 }
```

autosec A tag used by the toc and index.

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

Create an autosection tag.

```
5679 \newcommand*{\LWR@createautosec}[1]{%
5680 \LWR@htmntag{%
5681     #1 % space
5682     id=\textquotedbl\textcolor{LWR@color}\LWR@print@mbox{autosec-\arabic{page}}\textcolor{LWR@color}\textquotedbl%
5683 }%
5684 }
```

\LWR@pushoneclose {*<depth>*} {*<printclose>*} Stacks the new sectioning level's closing tag, to be used when this section is closed some time later.

 \LWR@stoppars must be executed first.

```
5685 \NewDocumentCommand{\LWR@pushoneclose}{m m}{%
5686 \LWR@traceinfo{LWR@pushoneclose #1}%
5687     \pushclose{#2}{#1}%
5688 }
```

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

Closes currently stacked tags of a lesser level, then opens the new nesting level by saving this new sectioning level's closing tag for later use.

 \LWR@stoppars must be executed first.

```
5689 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
5690 \LWR@closeprevious{#1}%

Push a new section depth:
```

```
5691 \LWR@pushoneclose{#1}{#2}%
5692 }
```

Ctr \LWR@prevFileDepth Remembers the previous LWR@FileDepth.

Initialized to a deep level so that any section will trigger a new HTML page after the home page.

```
5693 \newcounter{LWR@prevFileDepth}
5694 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

\@secntformat {<sectiontype>}

```
5695 \def \@secntformat#1{\csname the#1\endcsname\quad}
```

\simplechapterdelim Used by tocbibind and anonchap.

```
5696 \newcommand*\simplechapterdelim{}
```

\@chapcntformat {<sectiontype>}

\let to \@secntformat by default, but may be redefined by \simplechapter and \restorechapter from tocbibind or anonchap.

```
5697 \let\@chapcntformat\@seccntformat
\@partcntformat {\⟨sectiontype⟩}
\let to \@seccntformat by default, but may be redefined by ctex.
```

```
5698 \let\@partcntformat\@seccntformat
```

\@partnameformat Prints “Part” for part sections.
Nullified by ctex.

```
5699 \newcommand*{\@partnameformat}{\LWR@isolate{\partname}~}%
```

Ctr LWR@currentautosec Records the page number when the section was created. If a math expression is included in the section name, and SVG math is used, the corresponding `lateximage` will cause the page number to change by the time the following `autosec` label is created.

```
5700 \newcounter{LWR@currentautosec}
```

```
\LWR@section * [⟨TOC name⟩] {\⟨name⟩} {\⟨sectiontype⟩}
```

The common actions for the high-level sectioning commands.

```
5701 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
5702 \LWR@traceinfo{\LWR@section: starting}%
5703 \LWR@stopars}%
```

Cancel special `minipage` horizontal space interaction:

```
5704 \global\boolfalse{LWR@minipagethispar}%
```

Start a new HTML file unless starred, and if is a shallow sectioning depth.

Exception: Also start a new HTML file for `\part*`, for appendix.

Generate a new L^AT_EX page so that TOC and index page number points to the section:

```
5705 \LWR@traceinfo{\LWR@section: testing whether to start a new HTML file}%
5706 \IfBooleanT{#1}{\LWR@traceinfo{\LWR@section: starred}}%
5707 \ifbool{\LWR@forcinghtmlpage}{\LWR@traceinfo{\LWR@section: forcinghtmlpage}}{}%
5708 \ifthenelse{%
5709   \(%
5710     \(\NOT\equal{#1}{\BooleanTrue}\)\OR%
5711     \(\cnttest{\@nameuse{\LWR@depth#4}}{=}{\LWR@depthpart}\)\OR%
```

```

5712      \(\boolean{LWR@forcinghtmlpage}\)%
5713      \)%
5714      \AND%
5715      \cnttest{\@nameuse{LWR@depth#4}}{<=}{\value{FileDepth}}%
5716      \AND%
5717      \(%
5718          \NOT\boolean{CombineHigherDepths}\OR%
5719          \cnttest{\@nameuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}%
5720      \)%
5721      \AND%

5722      \(% phantomsection
5723          \NOT\isempty{#3}%
5724          \OR%
5725          \(\NOT\equal{#1}{\BooleanTrue}\)%
5726      \)%
5727 }%

```

If so: start a new HTML file:

```

5728 {%
5729     \LWR@traceinfo{LWR@section: new HTML file}%

```

See if there was an optional toc name entry:

```

5730     \IfNoValueTF{#2}%

```

If no optional entry

```

5731     {\LWR@newhtmlfile{#3}}%

```

If yes an optional entry

```

5732     {\LWR@newhtmlfile{#2}}%
5733 }% new file

```

Else: No new HTML file:

```

5734 {%

```

Generate a new L^TE_X page so that toc and index page number points to the section:

```

5735     \LWR@traceinfo{LWR@section: not a new HTML file, about to LWR@orignewpage}%
5736     \LWR@orignewpage%
5737 }% not new file
5738

```

Remember this section's name for \nameref:

```
5739 \IfValueT{#3}{%
5740     \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
5741     \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
5742 }%
```

Print an opening comment with the level and the name; ex: “section” “Introduction” Footnotes may be used in section names, which would also appear in the HTML section opening comments, so the short TOC entry is used if possible, and a limited opening comment is made if the sectional unit is starred.

```
5743
5744 \ifbool{HTMLDebugComments}{%
5745     \begingroup%
5746     \LWR@nullfonts%
5747     \IfBooleanTF{#1}{ starred
5748     {\LWR@htmlcomment{Opening #4*}}}
5749     {%
5750         \IfNoValueTF{#2}{ short TOC
5751             {\LWR@htmlcomment{Opening #4 ‘‘#3’’}}%
5752             {\LWR@htmlcomment{Opening #4 ‘‘#2’’}}%
5753         }
5754         \endgroup%
5755     }{%
5756 }
```

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

```
5757 \ifthenelse{%
5758     \cnttest{\@nameuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
5759 }%
5760 {\LWR@startpars}%
5761 {}%
```

Create the opening tag with an autosec:

```
5762 \LWR@traceinfo{LWR@section: about to LWR@createautosec}%
5763 \LWR@createautosec{\@nameuse{LWR@tag#4}}%
5764 \setcounter{LWR@currentautosec}{\value{page}}%
```

Check if starred:

```
5765 \IfBooleanTF{#1}{%
5766 {%
5767     \LWR@traceinfo{LWR@section: starred}%
5768 }
```

Starred, but also forcing a TOC entry, so add unnumbered TOC name or regular name:

```

5768     \ifbool{LWR@forcinghtmtoc}%
5769     {%
5770         \addcontentsline{toc}{#4}{%
5771             \IfValueTF{#2}{\LWR@isolate{#2}}{\LWR@isolate{#3}}%
5772         }%
5773     }%
5774     {}%
5775 }% starred

```

Not starred, so step counter and add to TOC:

```
5776 {%
```

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

```

5777     \ifthenelse{%
5778         \cnttest{\@nameuse{LWR@depth#4}}{<=}{\value{secnumdepth}}}%
5779     {%
5780         {%
5781             \if secnumdepth

```

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

```

5781     \LWR@traceinfo{LWR@section: about to test main matter}%
5782     \ifbool{LWR@mainmatter}%
5783     {%
5784         \LWR@traceinfo{LWR@section: yes mainmatter}%
5785         \refstepcounter{#4}%

```

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

```

5786     \LWR@traceinfo{LWR@section: about to addcontentsline}%
5787     \addcontentsline{toc}{#4}{%
5788     {%
5789         \protect\numberline{%
5790             \@nameuse{pre#4name}%
5791             \@nameuse{the#4}%
5792             \@nameuse{post#4name}%
5793         }%
5794     }%
5795         \ignorespaces%
5796         \IfValueTF{#2}{\LWR@isolate{#2}}{\LWR@isolate{#3}}\protect\relax%
5797     }%
5798 }%
5799     \LWR@traceinfo{LWR@section: finished addcontentsline}%
5800 }% end of if main matter

```

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

```

5801      {%
5802          \LWR@traceinfo{LWR@section: no main matter}%
5803          \addcontentsline{toc}{#4}{%
5804              \IfValueTF{#2}{\LWR@isolate{#2}}{\LWR@isolate{#3}}%
5805          }%
5806      }% end of not main matter
5807  }% end of secnumdepth

```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```

5808      {%
5809          \addcontentsline{toc}{#4}{%
5810              \IfValueTF{#2}{\LWR@isolate{#2}}{\LWR@isolate{#3}}%
5811          }%
5812      }%

```

For part, print “Part”:

```

5813  \ifbool{LWR@mainmatter}{%
5814      {%
5815          \ifthenelse{%
5816              \cnttest{@nameuse{LWR@depth#4}}{<=}%
5817              {\value{secnumdepth}}\And%
5818              \cnttest{@nameuse{LWR@depth#4}}{=}{\LWR@depthpart}\And%
5819          }%
5820          {@partnameformat}%
5821      }%

```

Print the section number:

```

5822      \LWR@traceinfo{LWR@section: about to print section number}%
5823      \ifthenelse{%
5824          \cnttest{@nameuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
5825      }%
5826      {%
5827          \ifstreq{#4}{part}{%
5828              \protect\LWR@sectionnumber{@partcntformat{#4}}%
5829          }%
5830          \ifstreq{#4}{chapter}{%
5831              \protect\LWR@sectionnumber{@chapcntformat{#4}}%
5832              \protect\LWR@sectionnumber{@secntformat{#4}}%
5833          }%
5834      }%
5835      {}%
5836      \LWR@traceinfo{LWR@section: finished print section number}%
5837  }{}%
5838 }% end of not starred

```

Print the section name:

```
5839 \LWR@traceinfo{LWR@section: about to print the section name}%
5840 \LWR@isolate{#3}%
```

Close the heading tag, such as /H2:

```
5841 \LWR@traceinfo{LWR@section: about to close the heading tag}%
5842 \LWR@htmntag{\nameuse{LWR@tag#4end}}%
```

Generate a L^AT_EX label:

```
5843 \LWR@traceinfo{LWR@section: about to create the LaTeX label}%
5844 \LWR@newautopagelabel{LWR@currentautosec}%
```

Start paragraph handing unless is an inline paragraph or subparagraph:

```
5845 \ifthenelse{%
5846     \cnttest{\nameuse{LWR@depth#4}}{<}{\LWR@depthparagraph}%
5847 }%
5848 {\LWR@startpars}%
5849 {}%
```

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

HOWEVER, allow a \part* to start a new HTML page. This is used by appendix.

A starred section does not trigger a new HTML page at the beginning of this macro, so it should not affect it here at the end either. This became an issue when a \listoftables was tested in the middle of the document. The \chapter* for the list was not allowing a new HTML page for the section following it while CombineHigherDepths was true.

```
5850 \ifthenelse{%
5851     \NOT\equal{#1}{\BooleanTrue}\OR%
5852     \cnttest{\nameuse{LWR@depth#4}}{=}{\LWR@depthpart}%
5853 }%
5854 {% not starred
5855     \setcounter{LWR@prevFileDepth}{\nameuse{LWR@depth#4}}%
5856 }% not starred
5857 {}%
```

Reset to defaults if not a phantomsection:

```
5858 \ifstrempty{#3}%
5859 {}%
5860 {}%
5861 \global\boolfalse{LWR@forcinghtmlpage}%

```

```

5862 \global\boolfalse{LWR@forcinghtmltoc}%
5863 }%
5864 %
5865 \LWR@traceinfo{LWR@section: done}%
5866 }

```

61.4 Pre- and post- sectioning names

\prepartname Usually null, but is used by `uj*` and `ut*` Japanese classes.

\postpartname

```

5867 \providecommand*\prepartname{}%
5868 \providecommand*\postpartname{}%

```

\prechaptername Usually null, but is used by `uj*` and `ut*` Japanese classes.

\postchaptername

```

5869 \providecommand*\prechaptername{}%
5870 \providecommand*\postchaptername{}%

```

\presectionname Always null, but provided here for algorithmic simplicity in `\LWR@section`.

\postsectionname

```

5871 \providecommand*\presectionname{}%
5872 \let\postsectionname\presectionname
5873
5874 \let\presubsectionname\presectionname
5875 \let\postsubsectionname\postsectionname
5876
5877 \let\presubsubsectionname\presectionname
5878 \let\postsubsubsectionname\postsectionname
5879
5880 \let\preparagraphname\presectionname
5881 \let\postparagraphname\postsectionname
5882
5883 \let\presubparagraphname\presectionname
5884 \let\postsubparagraphname\postsectionname

```

61.5 \section and friends

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

```

5885 \newcommand{\part@preamble}{}% for koma-script
5886
5887 \DeclareDocumentCommand{\part}{s o m}{%

```

```
5888 \LWR@maybeprintpendingfootnotes{\LWR@depthpart}%
5889 \LWR@stoppars%
5890
5891 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
5892
5893 \LWR@section{#1}{#2}{#3}{part}%
5894
5895 \part@preamble% for koma-script
5896 \renewcommand{\part@preamble}{}%
5897 }

\chapter * [<TOC name>] [<heading name>] {<name>}

5898 \let\@printcites\relax% for quotchap package
5899
5900 \newcommand{\chapter@preamble}{}% for koma-script
5901
5902 \@ifundefined{chapter}
5903 {}
5904 {%
5905 \DeclareDocumentCommand{\chapter}{s o o m}{%
5906 \IfValueTF{#2}{%
5907 \LWR@traceinfo{chapter #2}%
5908 }{%
5909 \LWR@traceinfo{chapter #4}%
5910 }%
5911 \LWR@maybeprintpendingfootnotes{\LWR@depthchapter}%
5912 \LWR@stoppars%
5913
5914 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
5915
5916 \LWR@section{#1}{#2}{#4}{chapter}%
5917
5918 \@printcites% for quotchap package
5919
5920 \chapter@preamble% for koma-script
5921 \renewcommand{\chapter@preamble}{}%
5922 }
5923 }

\section * [<TOC name>] [<heading name>] {<name>}

5924 \DeclareDocumentCommand{\section}{s o o m}{%
5925 \LWR@traceinfo{section: starting}%
5926 \LWR@maybeprintpendingfootnotes{\LWR@depthsection}%
5927 \LWR@stoppars%
5928
5929 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%

```

```
5930
5931 \LWR@section{#1}{#2}{#4}{section}%
5932 }

\subsection * [<TOC name>] {<name>}

5933 \DeclareDocumentCommand{\subsection}{s o m}{%
5934 \LWR@maybeprintpendingfootnotes{\LWR@depthsubsection}%
5935 \LWR@stoppars%
5936
5937 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
5938
5939 \LWR@section{#1}{#2}{#3}{subsection}%
5940 }

\subsubsection * [<TOC name>] {<name>}

5941 \DeclareDocumentCommand{\subsubsection}{s o m}{%
5942 \LWR@maybeprintpendingfootnotes{\LWR@depthsubsubsection}%
5943 \LWR@stoppars%
5944
5945 \LWR@startnewdepth{\LWR@depthsubsubsection}{%
5946 {\LWR@printclosesubsubsection}%
5947
5948 \LWR@section{#1}{#2}{#3}{subsubsection}%
5949 }

\paragraph * [<TOC name>] {<name>}

5950 \DeclareDocumentCommand{\paragraph}{s o m}{%
5951 \LWR@maybeprintpendingfootnotes{\LWR@depthparagraph}%
5952 \LWR@stoppars%
5953
5954 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
5955
5956 \LWR@section{#1}{#2}{#3}{paragraph}%
5957 }

\ subparagraph * [<TOC name>] {<name>}

5958 \DeclareDocumentCommand{\ subparagraph}{s o m}{%
5959 \LWR@maybeprintpendingfootnotes{\LWR@depthsubparagraph}%
5960 \LWR@stoppars%
5961
5962 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
5963
5964 \LWR@section{#1}{#2}{#3}{subparagraph}%
```

```

5965 }

5966 \end{warpHTML}

```

62 Starting a new file

for HTML & PRINT: 5967 \begin{warpall}

\HTMLLanguage Default language for the HTML lang tag.

```

5968 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
5969
5970 \newcommand*{\HTMLLanguage}[1]{%
5971 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
5972 }

```

\theHTMLTitleSeparator May be used inside \theHTMLTitleSection to separate the website's overall HTML title and the particular page's section name.

```

5973 \ifPDFTeX% pdflatex or dvi latex
5974   \ifdefstring{\inputencodingname}{utf8}{%
5975     \newcommand*{\theHTMLTitleSeparator}{-- }% EMdash
5976   }{%
5977     \newcommand*{\theHTMLTitleSeparator}{ - }% hyphen
5978   }%
5979 \else%
5980   \ifpTeX%
5981     \newcommand*{\theHTMLTitleSeparator}{ - }% hyphen
5982   \else%
5983     \newcommand*{\theHTMLTitleSeparator}{ -- }% EMdash
5984   \fi%
5985 \fi%

```

\HTMLTitleBeforeSection Sets the HTML page's meta title tag to show the website title before the section name.

```

5986 \newcommand*{\HTMLTitleBeforeSection}{%
5987   \def\theHTMLTitleSection{%
5988     \theHTMLTitle\theHTMLTitleSeparator\theHTMLSection%
5989   }%
5990 }

```

\HTMLTitleAfterSection Sets the HTML page's meta title tag to show the section name before the website title.

```
5991 \newcommand*\HTMLTitleAfterSection{%
5992     \def\theHTMLTitleSection{%
5993         \theHTMLSection\theHTMLTitleSeparator\theHTMLTitle%
5994     }%
5995 }
```

\theHTMLTitleSection Forms the HTML page's meta title tag. The default is to show the website title before the section name.

```
5996 \HTMLTitleBeforeSection
```

\theHTMLSection The section name is passed to \LWR@filestart, which then sets \theHTMLSection for use inside \theHTMLTitleSection to create an HTML meta title tag.

```
5997 \newcommand*\theHTMLSection{}%
5998 \end{warpall}
```

for HTML output: 5999 \begin{warpHTML}

\LWR@filestart [<sectionname>] Creates the opening HTML tags.

```
6000 \newcommand*\LWR@filestart}[1][]{%
6001 \LWR@traceinfo{\LWR@filestart !#1!}%
```

Locally temporarily disable direct-formatting commands:

```
6002 \begingroup%
6003 \LWR@nullfonts%
```

Save the section name for use while creating the HTML meta title tag:

```
6004 \edef\theHTMLSection{#1}%
```

Create the page's HTML header:

```
6005 \LWR@htmntag{!DOCTYPE html}\LWR@orignewline
```

The language is user-adjustable:

```
6006 \LWR@htmntag{%
6007     html lang=\textquotedbl\currentHTMLLanguage\textquotedbl%
6008 }\LWR@orignewline
```

Start of the meta data:

```
6009 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
6010 \LWR@htmltag{%
6011     meta charset=\textquotedbl{}UTF-8\textquotedbl\ /%
6012 }\LWR@orignewline
```

Author:

```
6013 \ifthenelse{\equal{\theHTMLAuthor}{}}{%
6014 {}%
6015 {%
6016     \LWR@htmltag{%
6017         meta name=\textquotedbl{}author\textquotedbl\ % space
6018         content=\textquotedbl{}\theHTMLAuthor\textquotedbl\ /%
6019     }\LWR@orignewline%
6020 }%
```

lwarp is the generator:

```
6021 \LWR@htmltag{%
6022     meta % space
6023     name=\textquotedbl{}generator\textquotedbl\ % space
6024     content=\textquotedbl{}LaTeX lwarp package\textquotedbl\ /%
6025 }\LWR@orignewline%
```

If there is a description, add it now:

```
6026 \ifdefempty{\LWR@currentHTMLDescription}{}{%
6027     \LWR@htmltag{%
6028         meta name=\textquotedbl{}description\textquotedbl\ % space
6029         content=\textquotedbl{}\LWR@currentHTMLDescription\textquotedbl\ /%
6030     }\LWR@orignewline
6031 }%
```

Mobile-friendly viewport:

```
6032 \LWR@htmltag{%
6033     meta % space
6034     name=\textquotedbl{}viewport\textquotedbl\ % space
6035     content=\textquotedbl{}width=device-width, initial-scale=1.0\textquotedbl\ /%
6036 }\LWR@orignewline
```

IE patch:

```

6037 \LWR@htmntag{!-/-[if lt IE 9]}\LWR@orignewline
6038 \LWR@htmntag{%
6039     script % space
6040     src=\textquotedbl{}%
6041         http://html5shiv.googlecode.com/svn/trunk/html5.js%
6042     \textquotedbl%
6043 }%
6044 \LWR@htmntag{/script}\LWR@orignewline
6045 \LWR@htmntag{![endif]-/-}\LWR@orignewline

```

The page's title, if there is one. A section name is also added if given.

```

6046 \ifthenelse{\equal{\theHTMLTitle}{}{}}%
6047 {}%
6048 {}%
6049     \LWR@htmntag{title}%
6050     \ifdefempty{\theHTMLSection}{%
6051         {\theHTMLTitle}%
6052         {\theHTMLTitleSection}%
6053     \LWR@htmntag{/title}\LWR@orignewline%
6054 }%

```

The page's stylesheet:

```

6055 \LWR@htmntag{%
6056     link % space
6057     rel=\textquotedbl{}stylesheet\textquotedbl\ % space
6058     type=\textquotedbl{}text/css\textquotedbl\ % space
6059     href=\textquotedbl{}\LWR@currentcss\textquotedbl\ /%
6060 }%
6061 \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.

```

6062 \ifbool{mathjax}{%
6063 {}%
6064     \begingroup%
6065     \LWR@restoreoriglists%
6066     \boolfalse{\LWR@verbtags}%
6067         \verbatiminput{lwarp_mathjax.txt}%
6068     \booltrue{\LWR@verbtags}%
6069     \endgroup%
6070     \LWR@stoppars%
6071 }% end of mathjax
6072 {}%

```

End of the header:

```
6073 \LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
6074 \LWR@htmltag{body}\LWR@orignewline
6075 \endgroup
6076 \LWR@traceinfo{LWR@filestart: done}
6077 }
```

```
6078 \end{warpHTML}
```

63 Starting HTML output

for HTML output: 6079 \begin{warpHTML}

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

The use of \textquotedbl instead of " improves compatibility with xeCJK.

```
6080 \catcode`$=\active
6081 \newcommand*{\LWR@LwarpStart}{%
6082   \LWR@traceinfo{LWR@lwarpStart}}
```

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

```
6084 \ifbool{FormatWP}{%
6085   \setcounter{FileDepth}{-5}%
6086   \boolfalse{HTMLDebugComments}}%
6087 }{}}
```

Expand and detokenize \HomeHTMLFilename and \HTMLFilename:

```
6088 \edef\LWR@strresult{\HomeHTMLFilename}
6089 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
6090 \edef\LWR@strresult{\HTMLFilename}
6091 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn and empty page style:

```
6092 \LWR@origonecolumn%
6093 \LWR@origpagestyle{empty}%
```

No black box for overfull lines:

```
6094 \overfullrule=0pt
```

Reduce chance of line overflow when HTML tags are added:

```
6095 \LWR@print@footnotesize%
```

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

```
6096 \LWR@print@raggedright%
6097 \LetLtxMacro{\\"}{\LWR@endofline}%
```

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

```
6098 \linespread{1.3}%
```

For *pdftotext* to reliably identify paragraph splits:

```
6099 \setlength{\parindent}{0pt}
6100 \setlength{\parskip}{2ex}
```

For the *lateximages* record file:

```
6101 \immediate\openout\LWR@lateximagesfile=lateximages.txt
```

Removes space around the caption in the HTML:

```
6102 \setlength{\belowcaptionskip}{0ex}
6103 \setlength{\abovecaptionskip}{0ex}
```

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

```
6104 \renewcommand{\ps@plain}{}%
```

Plug in some new actions. This is done just before the document start so that they won't be over-written by some other package.

Float captions:

```
6105 \let\LWR@origcaption\caption
```

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

```
6106 \let\LWR@origltx@label\ltx@label
6107 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
6108 \LetLtxMacro\TeX\LWR@TeX
6109 \LetLtxMacro\LaTeX\LWR@LaTeX
6110 \LetLtxMacro\LuaTeX\LWR@LuaTeX
6111 \LetLtxMacro\LuaLaTeX\LWR@LuaLaTeX
6112 \LetLtxMacro\XeTeX\LWR@XeTeX
6113 \LetLtxMacro\XeLaTeX\LWR@XeLaTeX
6114 \LetLtxMacro\ConTeXt\LWR@ConTeXt
```

Not yet started any paragraph handling:

```
6115 \global\boolfalse{LWR@doingapar}
6116 \global\boolfalse{LWR@doingstartpars}
```

Document and page settings:

```
6117 \mainmatter
6118 \LWR@origpagenumbering{arabic}
```

Start a new HTML file and a header:

```
6119 \LWR@traceinfo{LWR@lwarpStart: Starting new file.}
6120 \LWR@filestart
6121 \LWR@traceinfo{LWR@lwarpStart: Generating first header.}
6122 \LWR@htmlltag{header}\LWR@orignewline
6123 \LWR@startpars
6124 \LWR@firstpagetop
6125 \LWR@stoppars
6126 \LWR@htmlltag{/header}\LWR@orignewline

6127 \LWR@htmlelementclass{div}{bodywithoutsidetoc}
6128 \LWR@htmlelementclass{div}{bodycontainer}
6129 \LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
6130 \LWR@htmlelementclass{section}{textbody}
```

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

```
6131 \LWR@patchlists
```

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

```
6132 \catcode`\$=\active
```

Required for \nameref to work with SVG math:

```
6133 \immediate\write\@mainaux{\catcode`\$=3%}
6134 \LetLtxMacro{\LWR@syntaxhighlightone$}{\balance} for editor syntax highlighting
```

Allow HTML paragraphs to begin:

```
6135 \LWR@startpars
```

If using MATHJAX, disable \ensuremath by printing a nullified definition at the start of each file, and add further customizations:

```
6136 \LWR@customizeMathJax

6137 \LWR@traceinfo{\LWR@lwarfStart: done}
6138 }
6139 \catcode`\$=3% math shift until lwarf starts

6140 \end{warpHTML}
```

64 Ending HTML output

for HTML output: 6141 \begin{warpHTML}

\LWR@requesttoc {<boolean>} {<suffix>} Requests that a TOC, LOF, or LOTbe generated.

```
6142 \newcommand*{\LWR@requesttoc}[2]{%
6143 \ifbool{#1}{%
6144 {%
6145 \expandafter\newwrite\@nameuse{tf@#2}%
6146 \immediate\openout \@nameuse{tf@#2} \jobname.#2\relax%
6147 }{}%
6148 }}
```

\LWR@LwarfEnd Final stop of all HTML output:

```
6149 \newcommand*{\LWR@LwarfEnd}{%
6150 {%
6151 \LWR@stoppars%
6152 \LWR@closeprevious{\LWR@depthfinished}}
```

At the bottom of the ending file:

Close the textbody:

```
6153 \LWR@htmlelementclassend{section}{textbody}
6154 \LWR@htmlelementclassend{div}{bodycontainer}
6155 \LWR@htmlelementclassend{div}{bodyandsidetoc}
```

Print any pending footnotes:

```
6156 \LWR@printpendingfootnotes
```

Create the footer:

```
6157 \LWR@htmlelement{footer}
6158
6159 \LWR@pagebottom
6160
6161 \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.

```
6162 \ifthenelse{\boolean{FormatEPUB}}{\OR\boolean{FormatWP}}
6163 {}
6164 {
6165   \ifnumcomp{\value{LWR@htmlfilename}}{>}{0}{\LWR@botnavigation}{}
6166 }

6167 \LWR@stopars% final stop of all paragraphs
```

Finish the HTML file:

```
6168 \LWR@htmlelement{/body}\LWR@orignewline
6169 \LWR@htmlelement{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
6170 \LWR@orignewpage
```

For `\textrm{image}` commands:

```
6171 \immediate\closeout\LWR@lateximagesfile
6172 }

6173 \end{warpHTML}
```

65 Title page

package support lwarf supports the native L^AT_EX titling commands, and also supports the packages authblk and titling. If both are used, authblk should be loaded before titling.

⚠ load order

\published and \subtitle If using the titling package, additional titlepage fields for \published and \subtitle may be added by using \AddSubtitlePublished in the preamble. See section 65.8.

affiliation lwarf provides for the \author macro an additional \affiliation macro to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using titlingpage's \theauthor in the main text.

reusing titlepage information The titling package maintains the definitions of \thetitle, \theauthor, etc., after the title has been typeset. These commands are to be used to refer to the document's title and author, etc., in the main text. These definitions have the \thanks and \affiliation removed, and for \author the \and is replaced to generate a simple inline list of authors separated by commas. Note: \theauthor does not work well with authblk unless the traditional L^AT_EX syntax is used.

⚠ \theauthor, authblk

custom titlepages \printtitle, \printauthor, etc., are provided for use inside a custom titlepage or titlingpage environment, and these retain the \thanks and \affiliation.

\printthanks \printthanks has been added to force the printing of thanks inside a titlingpage environment when \maketitle is not used.

⚠ \thanks

Inside a \titlepage or \titlingpage environment, use \thanks instead of \footnote for acknowledgements, etc.

65.1 Setting the title, etc.

The following provide setting commands for both HTML and print outputs.

\author {{<author>}} While using \maketitle and print mode, the author is treated as a single-column tabular and the \and feature finishes the current tabular then starts a new one for the next author. Each author thus is placed into its own tabular, and an affiliation may be placed on its own line such as

```
\author{Name \\ Affiliation \and Second Name \\ Second Affiliation}
```

For HTML, the entire author block is placed inside a <div> of class author, and each individual author is inside a <div> of class oneauthor.

\@title \@title, \@author, etc. store the values as originally assigned, including any \thanks, \and, or \affiliation. These are low-level macros intended to be used by \@date

other macros only inside a `titlepage` or `titlingpage`, and are used by `\maketitle`. The author is printed inside a single-column `tabular`, which becomes multiple single-column `tabulars` if multiples authors are included. For `HTML`, these `tabulars` become side-by-side `<div>`s of class `oneauthor`, all of which are combined into one `<div>` of class `author`.

```
\printtitle \printtitle, etc. are user-level macros intended to be used in custom titlepage or titlingpage environments in cases where \maketitle is not desired. These commands preserve the \thanks, etc., and should not be used in the main text.

\thetitle \thetitle, \theauthor, and \thedate are available if titling has been loaded, and are sanitized user-level versions from which have been removed the \thanks and \affiliation, and \and is changed for inline text usage. The author is printed inline without \affiliation or \thanks, with \and placing commas between multiple authors. Thus, these commands are to be used in the main text whenever the user wishes to refer to the document's title and such. One practical use for this is to place the authors at the bottom of each HTML page, such as:
```

```
\HTMLPageBottom {\text{<div>}}
```

```
\HTMLPageBottom{
\begin{center}\text{copyright~2016 } \theauthor\end{center}
}
```

- ⚠ **\theauthor, authblk** `\theauthor` does not work well if `authblk` is used. If `\theauthor` is important, it is recommended to use the standard `LATEX` syntax for `\author`, optionally with `lwarp`'s `\affiliation` macro as well.
- ⚠ **affiliations** After `\maketitle` has completed, `\theauthor` retains the definition of the author, but `\and` is changed to become a comma and a space, intending to print the authors names separated by spaces. This fails when affiliations are included on their own table rows.
- \affiliation** A solution, provide here, is to define a macro `\affiliation` which, during `\maketitle`, starts a new row and adds the affiliation, but after `\maketitle` is finished `\affiliation` is re-defined to discard its argument, thus printing only the author names when `\author` is later used inline.

65.2 \if@titlepage

for **HTML & PRINT**: 6174 `\begin{warpall}`

\if@titlepage Some classes do not provide \if@titlepage. In this case, provide it and force it false.

```

6175 \ifcsvoid{@titlepagefalse}{
6176     \newif\if@titlepage
6177     \@titlepagefalse
6178 }{}}

6179 \end{warpall}

```

65.3 Changes for \affiliation

\affiliation {\text}

Adds the affiliation to the author for use in \maketitle.

Inside titlepage, this macro prints its argument. Outside, it is null.

for HTML & PRINT: 6180 \begin{warpall}
6181 \providerobustcmd{\affiliation}[1]{}
6182 \end{warpall}

for PRINT output: 6183 \begin{warpprint}

```

6184 \AtBeginEnvironment{titlepage}{%
6185 \renewrobustcmd{\affiliation}[1]{\textsc{\small#1}}
6186 }
6187
6188 \AtBeginDocument{
6189 \@ifpackageloaded{titling}{%
6190 \AtBeginEnvironment{titlingpage}{%
6191 \renewrobustcmd{\affiliation}[1]{\textsc{\small#1}}
6192 }
6193 }% titling loaded
6194 }% AtBeginDocument

```

6195 \end{warpprint}

for HTML output: 6196 \begin{warpHTML}

Env titlepage Sets up a <div> of class titlepage. Provided even for memoir class, since it is used by \maketitle.

```

6197 \DeclareDocumentEnvironment{titlepage}{}{%
6198 {

```

```

6199 \renewrobustcmd{\affiliation}[1]{\\ \InlineClass{affiliation}{##1}}
6200 \LWR@printpendingfootnotes
6201 \LWR@forcenewpage
6202 \BlockClass{titlepage}
6203 }
6204 {
6205 \endBlockClass
6206 \LWR@printpendingfootnotes
6207 }

6208 \end{warpHTML}

```

65.4 Printing the thanks

for HTML & PRINT: 6209 \begin{warpall}

\printthanks Forces the \thanks to be printed.

This is necessary in a titlingpage environment when \maketitle was not used.

```

6210 \newcommand*{\printthanks}{\@thanks}

6211 \end{warpall}

```

65.5 Printing the title, etc. in HTML

The following are for printing the title, etc. in a titlepage or a titlingpage in HTML:

for HTML output: 6212 \begin{warpHTML}

```

\printtitle

6213 \newcommand*{\printtitle}
6214 {%
6215   \LWR@stoppars%
6216   \LWR@htmltag{\LWR@tagtitle}%
6217   \@title%
6218   \LWR@htmltag{\LWR@tagtitleend}%
6219   \LWR@startpars%
6220 }

```

\LWR@printthetitle A private version which prints the title without footnotes, used to title each HTML page.

```
6221 \newcommand*\LWR@printthetitle{%
6222 {%
6223     \LWR@stoppars%
6224     \LWR@htmlltag{\LWR@tagtitle}%
6225     \thetitle%
6226     \LWR@htmlltag{\LWR@tagtitleend}%
6227     \LWR@startpars%
6228 }
```

\printauthor HTML version.

```
6229 \newcommand*\printauthor{
```

The entire author block is contained in a <div> named author:

```
6230 \begin{BlockClass}{author}
```

\and finishes one author and starts the next:

```
6231 \renewcommand{\and}{%
6232 \end{BlockClass}
6233 \begin{BlockClass}{oneauthor}
6234 }
```

Individual authors are contained in a <div> named oneauthor:

```
6235 \begin{BlockClass}{oneauthor}
6236 @author
6237 \end{BlockClass}
6238 \end{BlockClass}
6239 }
```

\printdate

```
6240 \newcommand*\printdate{%
6241 \begin{BlockClass}{titledate}
6242 @date
6243 \end{BlockClass}
6244 }
```

```
6245 \end{warpHTML}
```

65.6 Printing the title, etc. in print form

The following are for printing the title, etc. in a `titlepage` or a `titlingpage` in print form:

```
for PRINT output: 6246 \begin{warpprint}

\printtitle

6247 \newcommand*{\printtitle}{{\Huge\titlename}{}}

\printauthor Print mode.

6248 \newcommand*{\printauthor}{%
6249   {{\large\begin{tabular}[t]{c}\@author\end{tabular}}}}
\printdate

6250 \newcommand*{\printdate}{{\small\textit{\@date}}}

6251 \end{warpprint}
```

65.7 \maketitle for HTML output

An HTML `<div>` of class `titlepage` is used.

`\thanks` are a form of footnotes used in the title page. See section 58 for other kinds of footnotes.

See `\thanksmarkseries{series}`, below, to set the style of the footnote marks.

```
for HTML output: 6252 \begin{warpHTML}

6253 \@ifclassloaded{memoir}
6254 {
6255 \newcommand{\LWR@setfootnoteseries}{%
6256   \renewcommand\thefootnote{\@arabic\c@footnote}%
6257 }
6258 }{%
6259 \if@titlepage
6260 \newcommand{\LWR@setfootnoteseries}{%
6261   \renewcommand\thefootnote{\@arabic\c@footnote}%
6262 }
6263 \else
```

```

6264 \newcommand{\LWR@setfootnoteseries}{%
6265     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
6266 }
6267 \fi
6268 }% not memoir

```

\LWR@maketitlesetup Patches \thanks macros.

```
6269 \newcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```

6270 \LWR@setfootnoteseries%
6271 \def\@makefnmark{\textsuperscript{\thefootnote}}
    \thefootnote \Rightarrow \nameuse{arabic}{footnote}, or
    \thefootnote \Rightarrow \nameuse{fnsymbol}{footnote}

```

Redefine the footnote text:

```
6272 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
6273 \textsuperscript{\@thefnmark}~%
```

```

\makethanksmark \Rightarrow \thanksfootmark \Rightarrow \tamark \Rightarrow
\@thefnmark \Rightarrow \itshape a (or similar)

```

Print the text:

```

6274 ##1%
6275 }%
6276 }

```

```
\@fnsymbol {\langle counter\rangle}
```

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used \| which was not being seen by *pdftotext*.

```

6277 \def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
6278     \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLunicode{2016}}\or
6279     **\or \HTMLentity{dagger}\HTMLentity{dagger}\or
6280     \HTMLentity{Dagger}\HTMLentity{Dagger}\else\@ctrerr\fi}

```

\maketitle HTML mode. Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

6281 \renewcommand*\maketitle{%

An HTML titlepage <div> is used for all classes.

6282 \begin{titlepage}

Set up special patches:

6283 \LWR@maketitlesetup

Typeset the title, etc:

6284 \@maketitle

Immediately generate any \thanks footnotes:

6285 \@thanks

Close the HTML titlepage div and cleanup:

6286 \end{titlepage}
6287 \setcounter{footnote}{0} %
6288 \global\let\thanks\relax
6289 \global\let\maketitle\relax
6290 \global\let\@maketitle\relax
6291 \global\let\@thanks\@empty
6292 \global\let\@author\@empty
6293 \global\let\@date\@empty
6294 \global\let\@title\@empty
6295 \global\let\title\relax
6296 \global\let\author\relax
6297 \global\let\date\relax
6298 \global\let\and\relax
6299 }

\@maketitle HTML mode. Typesets the title, etc.:

6300 \DeclareDocumentCommand{\@maketitle}{}{
6301 \LWR@stoppars\LWR@htmltag{\LWR@tagtitle} %
6302 \@title%
6303 \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars%
6304 \begin{BlockClass}{author}}%

For IEEEtran class:

```

6305      \renewcommand*\{\cr\}{\%}
6306      \renewcommand*\{\crcr\}{\%}
6307      \renewcommand*\{\noalign\}{\%}

6308      \renewcommand{\and}{%
6309          \end{BlockClass}%
6310          \begin{BlockClass}{oneauthor}%
6311      }%
6312          \begin{BlockClass}{oneauthor}%
6313              \author{%
6314                  \end{BlockClass}%
6315          \end{BlockClass}%
6316          \begin{BlockClass}{titledate}%
6317              \date{%
6318          \end{BlockClass}%
6319      }

```

\LWR@titlingmaketitle \maketitle for use inside an HTML titlingpage environment.

```
6320 \newcommand*\{\LWR@titlingmaketitle\}{%
```

Keep pending footnotes out of the title block:

```
6321 \thanks
```

Set up special patches:

```
6322 \LWR@maketitlesetup
```

Typeset the title, etc:

```
6323 \maketitle
```

Immediately generate any \thanks footnotes:

```

6324 \thanks
6325 }
```

```
6326 \end{warpHTML}
```

65.8 \published and \subtitle

\subtitle and \published To add \subtitle and \published to the titlepage, load the titling package and

use `\AddSubtitlePublished` in the preamble.

The default `lwarp.css` has definitions for the `published` and `subtitle` classes.

If `titling` is loaded, `\AddSubtitlePublished` creates a number of additional macros, and also assigns some of the `titling` hooks. If `titling` is not loaded, `\AddSubtitlePublished` creates null macros.

- ⚠ **titling hooks** Do not use `\AddSubtitlePublished` if the user has patched the `titling` hooks for some other reason. Portions are marked `\warpprintonly` to reduce extra tags in HTML. Similarly, `BlockClass` has no effect in print mode. Thus, the following may be marked `warpall`.

for HTML & PRINT: 6327 `\begin{warpall}`

`\AddSubtitlePublished` Adds `\published` and `\subtitle`, and related.

```

6328 \newcommand*{\AddSubtitlePublished}{%
6329 @ifpackageloaded{titling}{% yes titling package
6330     \newcommand{\@published}{}%
6331     \newcommand{\published}[1]{\gdef\@published{\#1}}%
6332     \renewcommand*{\maketitlehooka}{\printpublished}%
6333     \newcommand*{\printpublished}{%
6334         \warpprintonly{\begin{center}\unskip}%
6335         \begin{BlockClass}{published}%
6336         \warpprintonly{\large\itshape}%
6337         \@published%
6338         \end{BlockClass}%
6339         \warpprintonly{\end{center}}%
6340     }%
6341     \newcommand{\@subtitle}{}%
6342     \newcommand{\subtitle}[1]{\gdef\@subtitle{\#1}}%
6343     \renewcommand*{\maketitlehookb}{\printsubtitle}%
6344     \newcommand*{\printsubtitle}{%
6345         \warpprintonly{\begin{center}\unskip}%
6346         \begin{BlockClass}{subtitle}%
6347         \warpprintonly{\Large\itshape}%
6348         \@subtitle%
6349         \end{BlockClass}%
6350         \warpprintonly{\end{center}}%
6351     }%
6352 }% yes titling package
6353 {% no titling package
6354     \newcommand{\published}[1]{}%
6355     \newcommand*{\printpublished}{}%
6356     \newcommand{\subtitle}{}%
6357     \newcommand*{\printsubtitle}{}%
6358 }% no titling package
6359 }% \AddSubtitlePublished

```

```
6360 \end{warpall}
```

66 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: 6361 \begin{warpHTML}

\abstractname User-redefinable title for the abstract.

Also over-written by the babel package.

```
6362 \providecommand*\{\abstractname\}{Abstract}
```

Some classes allow an optional name, so it is allowed here.

Env abstract

```
6363 \DeclareDocumentEnvironment{abstract}{O{\abstractname}}
6364 {
6365 \LWR@forcenewpage
6366 \BlockClass{abstract}
6367 \BlockClassSingle{\abstracttitle}{#1}
6368 }
6369 {
6370 \endBlockClass
6371 }
```

```
6372 \end{warpHTML}
```

67 Quote and verse

67.1 Attributions

\attribution For use with quote, quotation, verse:

Ex: "A quotation." \attribution{\textsc{Author Name}}\\ \textsl{Book Title}}

for HTML output: 6373 \begin{warpHTML}

```

6374 \newcommand{\attribution}[1]{%
6375   \begin{BlockClass}{attribution}
6376   #1
6377   \end{BlockClass}
6378 }
6379 \end{warpHTML}

for PRINT output: 6380 \begin{warpprint}
6381 \newcommand{\attribution}[1]{%
6382   \begin{flushright}
6383   \unskip
6384   #1
6385   \end{flushright}%
6386 }
6387 \end{warpprint}

```

67.2 Quotes, quotations

for HTML output: 6388 \begin{warpHTML}

```

Env  quote

6389 \renewenvironment*{quote}
6390 {
6391 \LWR@forcenewpage
6392 \LWR@htmlblocktag{blockquote}
6393 }
6394 {\LWR@htmlblocktag{/blockquote}}
6395
6396 \renewenvironment*{quotation}
6397 {
6398 \LWR@forcenewpage
6399 \LWR@htmlblocktag{blockquotation}
6400 }
6401 {\LWR@htmlblocktag{/blockquotation}}

6402 \end{warpHTML}

```

67.3 Verse

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

\attrib The documentation for the verse and memoir packages suggest defining an \attrib command, which may already exist in current documents, but it will only work for

print output. l warp provides \attribution, which works for both print and HTML output. To combine the two so that \attrib is used for print and \attribution is used for HTML:

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

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

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

67.3.1 L^AT_EX core verse environment

for HTML output: 6403 \begin{warpHTML}

Env **verse**

```
6404 \renewenvironment{verse}
6405     {\let\\newline% l warp
6406      \list{}{\itemsep      \z@%
6407              \itemindent   -1.5em%
6408              \listparindent\itemindent
6409              \rightmargin \leftmargin
6410              \advance\leftmargin 1.5em}%
6411      \item\relax}
6412      {\endlist}
6413 \end{warpHTML}
```

for HTML & PRINT: 6414 \begin{warpall}

67.3.2 verse and memoir

The following lengths are used by `verse` and `memoir`. They may be set in either print or HTML output, but are only used in HTML. This allows the user to set `\vleftskip` and `\leftmargini` for print output, and optionally select different values for HTML.

Len `\HTMLvleftskip` Sets `\vleftskip` inside a `verse` environment in HTML.

```
6415 \newlength{\HTMLvleftskip}
6416 \setlength{\HTMLvleftskip}{1em}
```

Len `\HTMLleftmargini` Sets `\leftmargini` inside a `verse` environment in HTML.

```
6417 \newlength{\HTMLleftmargini}
6418 \setlength{\HTMLleftmargini}{4.5em}
6419 \end{warpall}
```

68 Verbatim and tabbing

for HTML & PRINT: 6420 `\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.

```
6421 \newlength{\VerbatimHTMLWidth}
6422 \setlength{\VerbatimHTMLWidth}{4in}
6423 \end{warpall}
```

for HTML output: 6424 `\begin{warpHTML}`

Bool `LWR@verbtags` Used to temporarily turn off verbatim tags while doing `\verbatiminput` in the HTML head.

```
6425 \newbool{LWR@verbtags}
6426 \booltrue{LWR@verbtags}
```

`\LWR@atbeginverbatim` [`<1: style>`] [`<2: negative \baselineskip \vspace>`] [`<3: class>`]

Encloses a verbatim environment with the given css class.

The use of `\textquotedbl` instead of " improves compatibility with xeCJK.

```
6427 \newcommand*{\LWR@atbeginverbatim}[3] []
6428 {%
```

Avoid excessive space between lines:

```
6429 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
6430 \LWR@stoppars%
```

Create a new `pre` of the given class. The tags may temporarily be turned off for internal use, such as loading the MATHJAX script.

```
6431 \ifbool{\LWR@verbtags}{%
6432   \LWR@htmltag{pre class=\textquotedbl#3\textquotedbl%
6433   \ifthenelse{\equal{\#1}{}}{}{\ style=\textquotedbl#1\textquotedbl}%
6434 }%
6435   \LWR@orignewline% pre
6436   \leavevmode\unskip\LWR@print@vspace*{-\#2\baselineskip}%
6437 }{}}
```

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.

```
6438 \begingroup%
```

```
6439 % \LWR@print@normalsize%
6440 \LWR@origttfamily%
6441 \LWR@print@small%
```

Since inside a `<pre>`, restore the original list processing:

```
6442 \LWR@restoreoriglists%
```

Turn off babel-french extra space before punctuation:

```
6443 \LWR@FBcancel%
```

Do not produce HTML tags for `\hspace` inside a `verse` par. Restore plain L^AT_EX `\hspace` functionality:

```
6444 \LWR@select@print@hspace%
6445 }
```

```
\LWR@afterendverbatim {\langle negative \baselineskip \vspace \rangle}
```

Finishes enclosing a verbatim environment.

```
6446 \newcommand*\LWR@afterendverbatim}[1]{%
6447 \endgroup%
6448 \par%
```

At the end of the environment, close the pre:

```
6449 \ifbool{LWR@verbtags}{%
6450     \LWR@print@vspace*{-\baselineskip}%
6451     \noindent\LWR@htmltag{/pre}\LWR@orignewline% pre
6452 }{}}
```

Resume regular paragraph handling:

```
6453 \LWR@startpars%
6454 }
```

\verbatiminput {<filename>}

Patch \verbatiminput to add HTML tags:

```
6455 \let\LWRV@origverbatim@input\verbatim@input
6456
6457 \renewcommand{\verbatim@input}[2]{%
6458 \ifbool{LWR@verbtags}{\LWR@forcenewpage}{%
6459 \LWR@atbeginverbatim{2.5}{Verbatim}%
6460 \LWRV@origverbatim@input{\#1}{\#2}%
6461 \LWR@afterendverbatim{1.5}%
6462 }}
```

Env **verbatim**

```
6463 \AfterEndPreamble{
6464 \LWR@traceinfo{Patching verbatim.}
6465 \AtBeginEnvironment{verbatim}{%
6466 \LWR@forcenewpage%
6467 \LWR@atbeginverbatim{2.5}{verbatim}%
6468 }
6469 \AfterEndEnvironment{verbatim}{%
6470     \LWR@afterendverbatim{1}%
6471 }
6472 }
```

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

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

```

6473 \newcommand*{\LWR@HTML@tabbing}{%
6474 \LWR@forcenewpage%
6475 \LWR@atbeginverbatim{3}{tabbing}%
6476 \LWR@print@tabbing%
6477 }
6478
6479 \newcommand*{\LWR@HTML@endtabbing}{%
6480     \LWR@print@endtabbing%
6481     \LWR@afterendverbatim{1}%
6482 }
6483
6484 \LWR@formatted{tabbing}
6485 \LWR@formatted{endtabbing}

6486 \end{warpHTML}

```

69 Theorems

`\newtheorem` {⟨text⟩} [⟨counter⟩] -or- [⟨oldname⟩] {⟨text⟩}

A few minor changes are made to supply HTML tags.

- The entire theorem is placed into a `<div>` of class `theoremcontents`.
- The label for each theorem is placed inside a `` of class `theoremlabel`.
- The contents are placed inside a `<div>` of class `theoremcontents`.

for HTML output: 6487 `\begin{warpHTML}`

`\@begintheorem` {⟨name⟩} {⟨number⟩}

```

6488 \renewcommand{\@begintheorem}[2]{%
6489 \LWR@forcenewpage
6490 \BlockClass{theoremcontents}
6491 \trivlist
6492 \item[\InlineClass{theoremlabel}{\#1 \ #2 \ }]\itshape
6493 }

```

`\@opargbegintheorem` {⟨name⟩} {⟨number⟩} {⟨oparg⟩}

```

6494 \renewcommand{\@opargbegintheorem}{\begin{trivlist}\item[\hskip-\labelwidth\hskip-\labelsep#1 #2 (#3)]\itshape}
6495 \LWR@forcenewpage
6496 \BlockClass{theoremcontents}
6497 \trivlist
6498 \item[\InlineClass{theoremlabel}{\#1 \#2 (#3)}]\itshape
6499 }

\end{theorem}

6500 \renewcommand*{\@endtheorem}{\end{trivlist}}
6501 \end{BlockClass}%
6502 \end{theoremcontents}
6503 }

6504 \end{warpHTML}

```

70 Lists

The environments `itemize`, `enumerate`, and `description` are patched when `lwarp` is started. These patches support the standard L^AT_EX environments, as well as those of `enumerate`, `enumitem`, and `paralist`, and at least the French version of `babel`. Additional patches are done on a package-specific basis.

The L^AT_EX source for `itemize` and `enumerate` are found in `source2e`, but the source for `description` is found in `article.cls`, etc.

empty item To have an empty item, use `\mbox{}` or a trailing backslash. This forces a new line in print output, matching the new line which will appear in HTML output. Ex:

```

\begin{itemize}
\item \mbox{}%
\begin{itemize}
...
\end{itemize}
\item %
\begin{itemize}
...
\end{itemize}

```

\makelabel While inside a list environment, `lwarp` nullifies a number of T_EX horizontal skip and fill commands, allowing the user to define `\makelabel` for print mode while HTML mode ignores those commands.

 **label font** When defining `\makelabel` in a list environment, use `\textbf` etc. instead of `\bfseries`.

70.1 List environment

for HTML output: 6505 `\begin{warpHTML}`

`\LWR@printcloselist` May be locally redefined by `enumerate` or `description`.

6506 `\newcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}`

`\LWR@printopenlist` May be locally redefined by `enumerate` or `description`.

6507 `\newcommand*{\LWR@printopenlist}{\ul style="\LWR@print@mbox{list-style-type:none}"}`

`\@mklab` Removes PDF spacing.

```
6508 \AtBeginDocument{
6509 \def\@mklab#1{%
6510 %      \hfil %
6511 #1}
6512 \let\makelabel\@mklab
6513 }
```

`\@donoparitem` Modified for `HTML` output by replacing `TEX` boxes with plain text. Also removes PDF spacing.

```
6514 \def\@donoparitem{%
6515   \@noparitemfalse
6516%   \global\setbox\@labels\hbox{\hskip -\leftmargin
6517%                           \unhbox\@labels
6518%                           \hskip \leftmargin}%
6519%   \if@minipage\else
6520%     \tempskipa\lastskip
6521%     \vskip -\lastskip
6522%     \advance\tempskipa\outerparskip
6523%     \advance\tempskipa -\parskip
6524%     \vskip\tempskipa
6525%   \fi
6526 }
```

`\@item` Modified for `HTML` output by replacing `TEX` boxes with plain text. Also removes PDF spacing.

```
6527 \def\LWR@HTML@item[#1]{%
6528 \LWR@traceinfo{@item}
6529   \if@noperitem
6530     \odonoperitem
6531   \else
6532     \if@inlabel
6533       \indent
6534       \par
6535     \fi
6536   \ifhmode
6537     \unskip\unskip
6538     \par
6539   \fi
6540   \if@newlist
6541     \if@nobreak
6542       \onbitem
6543     \else
6544       \addpenalty\begin{parpenalty}
6545       \addvspace\topsep
6546       \addvspace{-\parskip}%
6547     \fi
6548   \else
6549     \addpenalty\itempenalty
6550     \addvspace\itemsep
6551   \fi
6552   \global\inlabeltrue
6553 \fi
6554 \everypar{%
6555   \minipagefalse
6556   \global\newlistfalse
6557 \if@inlabel
6558   \global\inlabelfalse
6559 \setbox\z@\lastbox
6560 \ifvoid\z@
6561   \kern-\itemindent
6562 \fi}%
6563 \box@labels
6564 \penalty\z@
6565 \fi
6566 \if@nobreak
6567   \nobreakfalse
6568   \clubpenalty \OM
6569 \else
6570   \clubpenalty \clubpenalty
6571 \everypar{}%
```

```

6572 %      \fi}%

6573   \if@noitemarg
6574     \o@noitemargfalse
6575   \if@nmbrlist

6576     \refstepcounter\@listctr
6577   \fi
6578 \fi

6579   \makelabel{\#1} % extra space
6580 % \sbox\@tempboxa{\makelabel{\#1}%
6581 % \global\setbox\@labels\hbox{%
6582 %   \unhbox\@labels
6583 %   \hskip \itemindent
6584 %   \hskip -\labelwidth
6585 %   \hskip -\labelsep
6586 %   \ifdim \wd\@tempboxa >\labelwidth
6587 %     \box\@tempboxa

6588 %   \else
6589 %     \hbox to\labelwidth {\unhbox\@tempboxa}%
6590 %   \fi
6591 %   \hskip \labelsep}%
6592 \ignorespaces%
6593 }

```

\@nbitem

```

6594 \def\@nbitem{%
6595 %   \tempskipa\@outerparskip
6596 %   \advance\@tempskipa -\parskip
6597 %   \addvspace\@tempskipa
6598 }

```

\LWR@listitem [⟨label⟩]

Handles \item inside a list, itemize, or enumerate.

See \LWR@openparagraph where extra \hskip is used to leave room for the label while inside a list during paragraph construction.

```

6599 \newcommand*\LWR@listitem{%
6600 \LWR@stopars%
6601 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem}%
6602 \LWR@htmltag{li}%
6603 \LWR@startpars%

```

```
6604 \LWR@origitem%
6605 }
```

\LWR@nulllistfills Nullifies various *TEX* fill commands, in case they are used inside *\makelabel*. Problems are caused when these are nullified all the time.

```
6606 \newcommand*\{\LWR@nulllistfills\}%
6607 \renewcommand*\{\hss\}{}%
6608 \renewcommand*\{\llap\}[1]{##1}%
6609 \renewcommand*\{\rlap\}[1]{##1}%
6610 \renewcommand*\{\hfil\}{}%
6611 \renewcommand*\{\hfilneg\}{}%
6612 \renewcommand*\{\hfill\}{}%
6613 }
```

Env list {*<label>*} {*commands*}

```
6614 \newcommand*\{\LWR@liststart\}%
6615 \LWR@traceinfo{\LWR@liststart}%
6616 \LWR@stoppars%
6617 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloselist}%
6618 \LWR@htmltag{\LWR@printopenlist}\LWR@orignewline%
6619 \LWR@startpars%
6620 \setlength{\topsep}{0pt}%
6621 \setlength{\partopsep}{0pt}%
6622 \setlength{\itemsep}{0pt}%
6623 \setlength{\parsep}{0pt}%
6624 \setlength{\leftmargin}{0pt}%
6625 \setlength{\rightmargin}{0pt}%
6626 \setlength{\listparindent}{0pt}%
6627 \setlength{\itemindent}{0pt}%
6628 \setlength{\labelsep}{1em}%
6629 \LWR@nulllistfills%
6630 }

6631 \newcommand*\{\LWR@listend\}%
6632 \LWR@traceinfo{\LWR@listend}%
6633 \LWR@stoppars%
6634 \LWR@closeprevious{\LWR@depthlist}%
6635 \LWR@startpars%
6636 }
```

70.2 Itemize

\LWR@itemizeitem [*<label>*]

Handles `\item` inside an itemize or enumerate.

See `\LWR@openparagraph` where extra `\hspace` is used to leave room for the label while inside a list during paragraph construction.

```
6637 \newcommand*\{LWR@itemizeitem}\{%
6638 \LWR@stopars%
6639 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem}%
6640 \LWR@htmntag{li}%
6641 \LWR@startpars%
6642 \LWR@origitem%
6643 }
```

Env `itemize` [*<options>*]

```
6644 \newcommand*\{LWR@itemizestart}\{%
6645 \renewcommand*\{LWR@printcloselist}{\LWR@printcloseitemize}%
6646 \renewcommand*\{LWR@printopenlist}{ul style="\LWR@print@mbox{list-style-type:none}"}
6647 \let\item\LWR@itemizeitem%
6648 \LWR@nulllistfills%
6649 }
```

70.3 Enumerate

An HTML unordered list is used with customized L^AT_EX-generated labels.

Env `enumerate` [*<options>*]

```
6650 \newcommand*\{LWR@enumeratestart}\{%
6651 \renewcommand*\{LWR@printcloselist}{\LWR@printcloseitemize}%
6652 \renewcommand*\{LWR@printopenlist}{ul style="\LWR@print@mbox{list-style-type:none}"}
6653 \let\item\LWR@itemizeitem%
6654 \LWR@nulllistfills%
6655 }
```

70.4 Description

`\LWR@descitem` [*<label>*] Handles an `\item` inside a description.

```
6656 \newcommand*\{LWR@descitem}[1] []%
6657 {%
6658 \LWR@stopars%
```

```
6659 \LWR@setlatestname{#1}%
6660 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem}%
```

Temporarily disable `\hspace`, which `article.cls`, etc. use per `\item` for descriptions only. This causes `lwarp` to mistakenly place an empty span between HTML list tags.

```
6661 \LWR@select@html@nohspace%
```

Process the original `\item` code:

```
6662 \LWR@origitem[]%
```

Restore `\hspace` for use in the item text:

```
6663 \LWR@select@html@hspace%
```

```
6664 \LWR@htmntag{dt}#1\LWR@htmntag{/dt}%
6665 \LWR@orignewline%
6666 \LWR@htmntag{dd}%
6667 \LWR@startpars%
6668 }
```

`Env description [⟨options⟩]`

```
6669 \newcommand*{\LWR@descriptionstart}{%
6670 \renewcommand*{\LWR@printcloselist}{\LWR@printclosedescription}%
6671 \renewcommand*{\LWR@printopenlist}{\LWR@descitem}%
6672 \let\item\LWR@descitem%
6673 \LWR@nulllistfills%
6674 }
```

70.5 Patching the lists

`\LWR@patchlists` Patches list environments.

`\LWR@patchlists` remembers `\item` as defined by whatever packages have been loaded, then patches the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native L^AT_EX environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```
6675 \newcommand*{\LWR@patchlists}{%
6676     \LetLtxMacro{\item}{\LWR@listitem}%
6677     \LetLtxMacro{\@item}{\LWR@HTML@item}%
6678     \renewcommand*{\@trivlist}{%
```

```
6679      \LWR@traceinfo{@trivlist start}%
6680      \LWR@liststart%
6681      \LWR@orig@trivlist%
6682      \LWR@traceinfo{@trivlist done}%
6683  }%
6684  \renewcommand*{\trivlist}{%
6685      \LWR@traceinfo{trivlist}%
6686      \LWR@origtrivlist%
6687  }%
6688  \renewcommand*{\endtrivlist}{%
6689      \LWR@traceinfo{endtrivlist start}%
6690      \LWR@origendtrivlist\LWR@listend%
6691      \LWR@traceinfo{endtrivlist done}%
6692  }%
6693  \renewcommand*{\itemize}{%
6694      \LWR@itemizestart\LWR@origitemize%
6695  }%
6696  \renewcommand*{\enumerate}{%
6697      \LWR@enumeratestart\LWR@origenumerate%
6698  }%
6699  \renewcommand*{\description}{%
6700      \LWR@descriptionstart\LWR@origdescription%
6701  }%
6702 }
```

\LWR@restoreoriglists Restores the original trivlist environment.

```
6703 \newcommand*{\LWR@restoreoriglists}{%
6704 \LWR@traceinfo{\LWR@restoreoriglists}%
6705 \LetLtxMacro{\item}{\LWR@origitem}%
6706 \LetLtxMacro{\@item}{\LWR@orig@item}%
6707 \let{@trivlist}{\LWR@orig@trivlist}%
6708 \let{\trivlist}{\LWR@origtrivlist}%
6709 \let{\endtrivlist}{\LWR@origendtrivlist}%
6710 \LetLtxMacro{\itemize}{\LWR@origitemize}%
6711 \LetLtxMacro{\enditemize}{\LWR@endorigitemize}%
6712 \LetLtxMacro{\enumerate}{\LWR@origenumerate}%
6713 \LetLtxMacro{\endenumerate}{\LWR@endorigenumerate}%
6714 \LetLtxMacro{\description}{\LWR@origdescription}%
6715 \LetLtxMacro{\enddescription}{\LWR@endorigdescription}%
6716 \let{@mklab}{\LWR@orig@mklab}%
6717 \let{\makelabel}{\LWR@origmakelabel}%
6718 \let{@donoparitem}{\LWR@orig@donoparitem}%
6719 \let{@nbitem}{\LWR@orig@nbitem}%
6720 }

6721 \end{warpHTML}
```

71 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax of the L^AT_EX core and the various tabular-related packages.

71.1 Limitations

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

Defining environments:

 Misplaced alignment
tab character &

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

`\StartDefiningTabulars`

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

`\StopDefiningTabulars`

This includes before and after defining any macro which used `\ttabbox` from `floatrow`.

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

`\StartDefiningTabulars % because & is used in a`

`definition`

`\newenvironment{outerenvironment}`

`{`

`\tabular{cc}`

`left & right \\`

`}`

`{`

`\TabularMacro\ResumeTabular`

`left & right \\`

`\endtabular`

`}`

`\StopDefiningTabulars`

 floatrow

 tabular inside another
environment

Cell contents:**⚠ \multirow**

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

```
... & \multirow{2}{.5in}{text} & ...
... & \mrowcell & ...
```

vposn

Note that recent versions of multirow include a new optional vposn argument.

- The multirow documentation regarding colored cells recommends using a negative number of rows. This will not work with l warp, so \warpprintonly and \warpHTMLonly must be used to make versions for print and HTML.

- See section 299.2 for \multicolumnrow.

l warp does not support directly combining \multicolumn and \multirow. Use \multicolumnrow instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}{c}{3}{0}{1in}[Opt]{Text}
```

The two arguments for \multicolumn come first, followed by the five arguments for \multirow, many of which are optional, followed by the contents.

As per \multirow, skipped cells to the right of the \multicolumnrow statement are not included in the source code on the same line. On the following lines, \mcolrowcell must be used for each cell of each column and each row to be skipped:

```
... & \multicolumnrow{2}{c}{c}{3}{0}{1in}[Opt]{Text} & ...
... & \mcolrowcell & \mcolrowcell & ...
... & \mcolrowcell & \mcolrowcell & ...
```

vposn

Note that recent versions of multirow include a new optional vposn argument.

- Using a custom macro inside a tabular data cell may result in an extra HTML data cell tag, corrupting the HTML table. To avoid this, use \TabularMacro just before the macro. This is ignored in print mode.

```
\TabularMacro\somemacro & more row contents \\
```

Column specifiers:**⚠ * column specification**

- * in a column specification is not used (so far). Repeat the column type the correct number of times.

@ and !

- Only one each of @ and ! is used at each column, and they are used in that order.

\multirow

- In \multirow cells, the print version may have extra instances of <, >, @, and ! cells on the second and later rows in the \multirow which do not appear in the HTML version.

⚠ `\newcolumntype`

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

Rules:

`vertical rules`

- Doubled `\hlines`, `\midrules`, and vertical rules are supported.
- Vertical rules next to either side of an @ or ! column are displayed on both sides of the column.

`width and trim`

- Width options are honored. Trim options are converted to rounded top corners. Trim corners are not rounded with @ or ! columns, and full-width rules ignore trim.

`full-width rules`

- `\toprule`, `\midrule`, `\bottomrule`, and `\hline` ignore trim. When given an optional width, each cell is styled to create the custom border. Without an optional width, the entire row is given a class to assign the standard border.

`combined rules`

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

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

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

- 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.

colortbl:

⚠ `row/cell color`

Only use `\rowcolor` and `\cellcolor` at the start of a row, in that order. colortbl ignores the overhang arguments.

Other:

`longtable headings`

- `tabularx` ignores the width, but X columns do produce paragraph columns or multicolumns.

⚠ `S columns`

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

- For S columns (from the siunitx package), while producing print output, anything non-numeric must be placed inside {} braces, including commands such as `\multirow`. While producing HTML output, though, anything placed inside braces is not seen by lwarf’s tabular handling algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version

for HTML output, without the extra braces, such as:

```
\warpprintonly{1 & 2 & {\multicolumn{2}{2cm}{Text}} & 3 \\}
\warpHTMLonly{1 & 2 & \multicolumn{2}{2cm}{Text} & 3 \\}
```

for HTML output: 6722 \begin{warpHTML}

71.2 Temporary package-related macros

These macros are temporary placeholders for macros defined by various packages. If the relevant package is not loaded, these placeholders are used instead.

71.2.1 arydshln

Emulated by the original L^AT_EX non-dashed versions.

```
6723 \LetLtxMacro{\hdashline}{\hline}
6724 \LetLtxMacro{\cdashline}{\cline}
6725 \LetLtxMacro{\firstdashline}{\hline}
6726 \LetLtxMacro{\lastdashline}{\hline}
```

71.3 Token lookahead

Used by \LWR@futurenonspacelet to look at the next token.

\LWR@mynexttoken

```
6727 \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

```
6728 \def{\LWR@futurenonspacelet#1{\def{\LWR@cs{#1}}%}
6729 \afterassignment{\LWR@fnalone\let\nexttoken= }%
6730 \def{\LWR@fnalone{\expandafter{\futurelet{\LWR@cs{\LWR@fnsltwo}}%}
6731 \def{\LWR@fnsltwo{}}
```

```

6732 \expandafter\ifx\LWR@cs\@sp token\let\next=\LWR@fnslthree%
6733 \else\let\next=\nexttoken\fi\next}
6734 \def\LWR@fnslthree{\afterassignment\LWR@fnslone\let\next= }

```

\LWR@getmynexttoken Looks ahead and copies the next token into \LWR@mynexttoken.

```

6735 \newcommand*\{\LWR@getmynexttoken}{%
6736 \LWR@traceinfo{\LWR@getmynexttoken}%
6737 % nothing must follow this next line
6738 \LWR@futureonospacelet\LWR@mynexttoken\LWR@tabledatacolumntag
6739 }

```

71.4 Tabular variables

Bool LWR@startedrow True if should print a row tag before this column.

```

6740 \newbool{\LWR@startedrow}
6741 \boolfalse{\LWR@startedrow}

```

Bool LWR@tabularcelladded True if have added a data cell for this position.

```

6742 \newbool{\LWR@tabularcelladded}
6743 \boolfalse{\LWR@tabularcelladded}

```

Ctr LWR@hlines Number of \hlines or \midrules above the next row.

```

6744 \newcounter{\LWR@hlines}

```

Ctr LWR@hdashedlines Number of arydshln dashed lines above the next row.

```

6745 \newcounter{\LWR@hdashedlines}

```

Bool LWR@doingtbrule True if the next row will have a top/bottom rule above it.

```

6746 \newbool{\LWR@doingtbrule}
6747 \boolfalse{\LWR@doingtbrule}

```

Bool LWR@doingcmidrule True if the next row will have a cmidrule above it.

This is used by \LWR@tabularfinishrow to force a final empty row to create the border for the \cmidrule.

```

6748 \newbool{\LWR@doingcmidrule}
6749 \boolfalse{\LWR@doingcmidrule}

```

Bool LWR@tableparcell True if are handling a paragraph inside a table cell, so must close the paragraph tag before moving on.

6750 \newbool{LWR@tableparcell}

Bool LWR@skippingmrowcell True if are doing an empty \multirow cell, and thus there is no data tag to close.

6751 \newbool{LWR@skippingmrowcell}

Bool LWR@skippingmcolrowcell True if are doing an empty \multicolumn cell, and thus there is no data tag to close, and do not print @ and ! columns.

6752 \newbool{LWR@skippingmcolrowcell}

Bool LWR@skipatbang True if just finished a \multicolumn so should not create the trailing @ or ! columns table data cells.

6753 \newbool{LWR@skipatbang}

Bool LWR@emptyatbang True if finishing a row and should print empty @ or ! column table data cells.

6754 \newbool{LWR@emptyatbang}

Bool LWR@intabularmetadata True if are in a tabular but not in a data cell. Used to prevent extra HTML breaks if not inside table data.

6755 \newbool{LWR@intabularmetadata}

6756 \boolfalse{LWR@intabularmetadata}

Ctr LWR@tabularDepth Tracks whether & is being used inside a tabular.

6757 \newcounter{LWR@tabulardepth}

6758 \setcounter{LWR@tabulardepth}{0}

Ctr LWR@tabularpardepth Tracks whether should look ahead at the next token when encountering a \par while processing tabular contents.

When LWR@tabularpardepth is deeper than LWR@tabulardepth then lwarp has started looking at the contents of the tabular, and thus any \pars encountered must be followed by another token lookahead.

6759 \newcounter{LWR@tabularpardepth}

6760 \setcounter{LWR@tabularpardepth}{0}

6761 \newcommand*{\LWR@colsresult}{}%temp storage for column format results

6762 \newcommand*{\LWR@pposition}{}%

6763 \newcommand*{\LWR@pleft}{}%

6764 \newcommand*{\LWR@pright}{}%

`LWR@tablecolspec` Holds the parsed column specification, of total width `LWR@tabletotalLaTeXcols`, not counting @ and ! columns.

Will contain a string such as `l1rrccpc`, exactly one letter per L^AT_EX table column, without @, !, <, >, or the vertical bar.

`\LWR@strresult` Holds the result of Str functions.

```
6765 \providecommand*\{\LWR@strresult}\{}  
6766 \providecommand*\{\LWR@strresulttwo}\{}
```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

```
6767 \newcommand*\{\LWR@origcolspec}\{}
```

`Ctr LWR@tablecolspecwidth` Holds the number of tokens in the table columns specification.

This includes one for each @, !, <, > column, and also one for each of the parameters of p, @, !, <, > columns, and three for each D column.

(This is not the total # of L^AT_EX columns in the table.)

```
6768 \newcounter{LWR@tablecolspecwidth}
```

`Ctr LWR@tablecolspecindex` While parsing the L^AT_EX table column specification, starts at 1 and is incremented per token of the specification.

```
6769 \newcounter{LWR@tablecolspecindex}
```

`Ctr LWR@tableLaTeXcolindex` While producing the table, resets to 1 at the start of the table and also at each end of line, and is incremented by 1 by each ampersand.

```
6770 \newcounter{LWR@tableLaTeXcolindex}
```

`Ctr LWR@tabletotalLaTeXcols` While parsing a table column specification, begins at 0 and increments by 1 per L^AT_EX table column. Eventually holds the final number of L^AT_EX table columns in each row, not counting @ and ! columns. (In HTML, @ and ! cells become their own columns, but are not included in `LWR@tabletotalLaTeXcols`.)

```
6771 \newcounter{LWR@tabletotalLaTeXcols}
```

`Ctr LWR@tabletotalLaTeXcolsnext` Holds the next L^AT_EX table column index while parsing, equal to one more than `LWR@tabletotalLaTeXcols`.

```
6772 \newcounter{LWR@tabletotalLaTeXcolsnext}
```

LWR@colatspec	A data array of specifications for @ columns. The leftmost's index is <code>leftedge</code> , the others are counter values. See section 42.
LWR@colbangspec	A data array of specifications for ! columns. The leftmost's index is <code>leftedge</code> , the others are counter values. See section 42.
LWR@colbeforespec	A data array of specifications for > columns.
LWR@colafterspec	A data array of specifications for < columns.
LWR@colbarspec	A data array of specifications for vertical rules.

71.5 Handling &, @, !, and bar

For technical discussion regarding problems redefining \&, See:

<http://tex.stackexchange.com/questions/11638/where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860>

```
\LWR@instertatbangcols
```

```
6773 \newcommand*{\LWR@instertatbangcols}{%
6774 \ifbool{LWR@skipatbang}{%
6775 {}{%
6776 {}{%
6777     \LWR@printatbang{at}{\arabic{LWR@tableLaTeXcolindex}}{%
6778     \LWR@printatbang{bang}{\arabic{LWR@tableLaTeXcolindex}}{%
6779 }{%
6780 }}
```

\LWR@closetabledatacell If LWR@skippingmrowcell or LWR@skippingmcolrowcell then there is no data tag to close. Otherwise, close any paragraphs, then close the data tag.

```
6781 \newcommand*{\LWR@closetabledatacell}{%
6782 \LWR@stoppars{%
6783 \global\booltrue{LWR@intabularmetadata}{%
6784 \ifbool{LWR@exittingtabular}{}{%
6785 % not exiting tabular
6786     \ifboolexpr{bool{LWR@skippingmrowcell} \or bool{LWR@skippingmcolrowcell}}{%
6787         {}{%
```

If not skipping a \multicolumnrow cell, insert the @ and ! columns after this non-existent column.

```
6788     \ifbool{LWR@skippingmcolrowcell}{%
6789         {}{%
```

```

6790           {\LWR@insertatbangcols}%
6791       }%
6792   {%
6793     not skippingmrowcell

```

Insert any < then any @ and ! column contents, unless muted for the \bottomrule or a \multicolumn:

```

6793   \unskip%
6794   \ifboolexpr{%
6795     bool{\LWR@tabularmutemods} or
6796     bool{\LWR@skipatbang} or
6797     bool{\LWR@emptyatbang}%
6798   }%
6799   {}%
6800   {\LWR@getexpparray{\LWR@col afterspec}{\arabic{\LWR@tableLaTeXcolindex}}}%

```

Close paragraphs:

```

6801   \ifbool{\LWR@tableparcell}{\LWR@stoppars}{}%
6802   \global\boolfalse{\LWR@tableparcell}%

```

Close the table data cell.

Close any color <div>s.

```

6803   \whileboolexpr{test {\ifnumcomp{\value{\LWR@cellcolordepth}}{>}{0}}}{{%
6804     \LWR@htmlltag{/div}\LWR@orignewline%
6805     \addtocounter{\LWR@cellcolordepth}{-1}%
6806   }}%

```

Skip the @ and ! cells if are closing a multicolumn cell.

```

6807   \leavevemode\unskip\LWR@htmlltag{/td}\LWR@orignewline%
6808   \global\booltrue{\LWR@tabularcelladded}%
6809   \LWR@insertatbangcols%
6810 }% not skipping mrowcell
6811 }% not exiting tabular
6812 \global\boolfalse{\LWR@skippingmrowcell}%
6813 \global\boolfalse{\LWR@skippingmcolrowcell}%
6814 \global\boolfalse{\LWR@skipatbang}%

```

Color control. Column is set by >{} for each cell, so it must be cleared here.

```

6815 \gdef{\LWR@cellHTMLcolor{}}%
6816 \gdef{\LWR@columnHTMLcolor{}}%
6817 \setcounter{\LWR@cellcolordepth}{0}%
6818 }%

```

When not used inside a `tabular`, & performs its original function as recorded here (with catcode 4).

```
6819 \let\LWR@origampmacro&
```

```
6820 \end{warpHTML}
```

71.5.1 Handling &

for HTML output: 6821 `\begin{warpHTML}`

& Will behave depending on whether it is being used inside `tabular`.

& is redefined to test whether it is inside a `tabular` environment, in which case it performs special processing for HTML conversion. If not, it behaves normally.

```
6822 \newcommand*\LWR@tabularampersand{%
6823 \LWR@traceinfo{\LWR@tabularampersand}%
6824 \ifnumcomp{\value{\LWR@tabulardepth}}{>}{0}%
6825 {%
```

If not skipping a multirow cell, close the current data cell.

```
6826 \unskip%
6827 \LWR@closetabledatamax%
```

Move to the next column.

```
6828 \addtocounter{\LWR@tableLaTeXcolindex}{1}%
```

Have not yet added data in this column:

```
6829 \global\boolfalse{\LWR@tabularcelladded}%
```

Look at the next token to decide multi or single column data tag.

```
6830 \LWR@getmynexttoken%
6831 }%
```

If not inside a `tabular`, performs the original action:

```
6832 {\LWR@origampmacro}%
6833 }
```

& is left with its original catcode for now.

tikz package seems to require & be left alone until after tikz has been loaded. Also, cleveref uses the ampersand in one of its options.

& is made active inside a tabular.

& is left alone when in math alignments.

71.5.2 Filling an unfinished row

\LWR@tabularfinishrow Adds empty table cells if necessary to finish the row.

At the end of the table, if any bottom rules are requested then an empty row must be generated to form the borders which show the rules.

```
6834 \newcommand*\LWR@tabularfinishrow{%
```

If not exiting the tabular, or doing a rule, or have already started a row, finish this row:

```
6835 \ifboolexpr{%
6836   not bool {LWR@exittingtabular} or%
6837   bool{LWR@doingtbrule} or%
6838   bool{LWR@doingcmidrule} or%
6839   test{\ifnumcomp{\value{LWR@hlines}}{>}{0}} or%
6840   test{\ifnumcomp{\value{LWR@hdashedlines}}{>}{0}} or%
6841   bool{LWR@startedrow}%
6842 }{%
```

To temporarily turn off LWR@exittingtabular so that table data tags will still be generated:

If generating a final row for the \bottomrule borders, turn off the @, !, <, and > column output:

```
6843 \ifbool{LWR@exittingtabular}{%
6844   \global\booltrue{LWR@tabularmutemods}%
6845 }{%
6846   \global\boolfalse{LWR@tabularmutemods}%
6847 }%
```

Locally reenable the table data tags until finished with the final row:

```
6848 \global\boolfalse{LWR@exittingtabular}%
```

Generate table data tags and ampersands until the right edge:

```

6849 \whileboolexpr{%
6850     test {
6851         \ifnumcomp{\value{LWR@tableLaTeXcolindex}}{<}{\value{LWR@tabletotalLaTeXcols}}
6852     } or %
6853     (%
6854         bool{LWR@intabularmetadata} and%
6855         not bool{LWR@tabularcelladded} and%
6856         test {
6857             \ifnumcomp{\value{LWR@tableLaTeXcolindex}}{=}{\value{LWR@tabletotalLaTeXcols}}
6858         }%
6859     )%
6860 }%
6861 {%
6862     \LWR@tabledatainglecolumntag%

```

The following is essentially `\LWR@tabularampersand` with `LWR@emptyatbang` added to empty the following cells:

```

6863     \LWR@closetabledatacell%
6864     \addtocounter{LWR@tableLaTeXcolindex}{1}%
6865     \global\boolfalse{LWR@tabularcelladded}%
6866     \global\booltrue{LWR@emptyatbang}%

```

Starts the next cell:

```

6867     \ifnumcomp{\value{LWR@tableLaTeXcolindex}}{<}{\value{LWR@tabletotalLaTeXcols}}{%
6868         \LWR@getmynexttoken}%
6869     {}%
6870 }%

```

Reenable the original `LWR@exittingtabular` to close the entire table:

```

6871 \ifbool{LWR@tabularmutemods}{%
6872     \global\booltrue{LWR@exittingtabular}%
6873 }{%
6874     \global\boolfalse{LWR@exittingtabular}%
6875 }%
6876 \global\boolfalse{LWR@tabularmutemods}%

6877 \global\boolfalse{LWR@emptyatbang}%
6878 }{}% ifboolexpr
6879 }

```

71.6 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendofline

Throws away options \\[dim] or *

\LWR@tabularendofline

```
6880 \NewDocumentCommand{\LWR@tabularendofline}{s o}{%
```

Finish the row:

```
6881 \ifnumcomp{\value{LWR@tableLaTeXcolindex}}{<}{\value{LWR@tabletotalLaTeXcols}}%
6882     {\LWR@tabularfinishrow}%
6883     {\LWR@closetabledatacell}%
6884 \LWR@htmlltag{/tr}\LWR@orignewline%
```

xcolor row color support:

```
6885 @rowc@lors%
```

No longer inside a data cell:

```
6886 \global\booltrue{LWR@intabularmetadata}%
```

Not yet started a table row:

```
6887 \global\boolfalse{LWR@startedrow}%
```

Additional setup:

```
6888 \setcounter{LWR@hlines}{0}%
6889 \setcounter{LWR@hdashedlines}{0}%
6890 \global\boolfalse{LWR@doingtbrule}%
6891 \global\boolfalse{LWR@doingcmidrule}%
6892 \LWR@clearmidrules%
6893 \gdef\LWR@rowHTMLcolor{}%
```

Start at first column:

```
6894 \setcounter{LWR@tableLaTeXcolindex}{1}%
```

Have not yet added data in this column:

```
6895 \global\boolfalse{LWR@tabularcelladded}%
```

Allow TEX to flush the pending paragraph. Not doing so causes a slowdown for very large tables.

```
6896 \LWR@stoppars
6897 \LWR@origpar
```

Look at the next token to decide between single column data tag or a special case:

```
6898 \LWR@getmynexttoken%
6899 }
```

71.7 Looking ahead in the column specifications

```
\LWR@columnspecklookahead {<offset>}
```

Looks offset tokens ahead in the column specification, setting \LWR@strresulttwo.

The w column alignment will be seen as a single unit such as {c}.

```
6900 \newcommand*{\LWR@columnspecklookahead}[1]{%
6901 \setcounter{\LWR@tempcountone}{\value{\LWR@tablecolspeccindex}}%
6902 \addtocounter{\LWR@tempcountone}{#1}%
6903 \fullexpandarg%
6904 \StrChar{\LWR@origcolspec}{\arabic{\LWR@tempcountone}}[\LWR@strresulttwo]%
```

Get the contents of the first group in \LWR@strresulttwo:

```
6905 \exploregroups%
6906 \StrChar{\LWR@strresulttwo}{1}[\LWR@strresulttwo]%
6907 \noexploregroups%
6908 }
```

71.8 Parsing @, >, <, !, bar columns

Holds the parsed argument for @, >, <, or ! columns:

```
6909 \newcommand*{\LWR@colparameter}{}%
```

```
\LWR@parseatcolumn Handles {@{text}} columns.
```

```
6910 \newcommand*{\LWR@parseatcolumn}{}%
```

Move to the next token after the '@':

```
6911 \LWR@traceinfo{at column}%
6912 \addtocounter{LWR@tablecolsindex}{1}%
```

Read the next token into \LWR@colparameter, expanding once:

```
6913 \LWR@traceinfo{about to read the next token:}%
6914 \expandarg%
6915 \StrChar{\LWR@origcolspec}%
6916   {\arabic{LWR@tablecolsindex}}[\LWR@colparameter]
6917 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
6918 \LWR@traceinfo{have now read the next token}%
6919 \ifnumcomp{\value{LWR@tabletotalLaTeXcols}}{=}{0}%
6920 {%
  left edge of the table:
  \LWR@traceinfo{at the left edge}%
  \LWR@setexpparray{\LWR@colatspec}{leftedge}[\LWR@colparameter]%
  \LWR@traceinfo{at the left edge: \%}
  \LWR@getexpparray{\LWR@colatspec}{leftedge}%
}%
6926 {%
  not at the left edge:
  \LWR@traceinfo{not at the left edge}%
  \LWR@setexpparray{\LWR@colatspec}%
    {\arabic{LWR@tabletotalLaTeXcols}}[\LWR@colparameter]%
  \LWR@traceinfo{at \arabic{LWR@tabletotalLaTeXcols}: \%}
  \LWR@getexpparray{\LWR@colatspec}{\arabic{LWR@tabletotalLaTeXcols}}%
}%
6932 }%
6933 \let\LWR@colparameter\relax%
6934 \global\booltrue{LWR@validtablecol}%
6935 }
```

\LWR@parsebangcolumn Handles !{text} columns.

```
6936 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
6937 \LWR@traceinfo{bang column}%
6938 \addtocounter{LWR@tablecolsindex}{1}%
```

Read the next token into \LWR@colparameter, expanding once:

```
6939 \LWR@traceinfo{about to read the next token:}%
6940 \expandarg%
```

```

6941 \StrChar{\LWR@origcolspect}%
6942   {\arabic{LWR@tablecolspectindex}}[\LWR@colparameter]
6943 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@colparameter:

```

6944 \LWR@traceinfo{have now read the next token}%
6945 \ifnumcomp{\value{LWR@tabletotalLaTeXcols}}{=}{0}%
6946 {%
6947   \LWR@traceinfo{at the left edge}%
6948   \LWR@setexparray{\LWR@colbangspec}{leftedge}{\LWR@colparameter}%
6949 }%
6950 {%
6951   \LWR@traceinfo{not at the left edge}%
6952   \LWR@setexparray{\LWR@colbangspec}%
6953     {\arabic{LWR@tabletotalLaTeXcols}}{\LWR@colparameter}%
6954   \LWR@traceinfo{bang \arabic{LWR@tabletotalLaTeXcols}: \LWR@colparameter!}%
6955 }%
6956 \let\LWR@colparameter\relax%
6957 \global\booltrue{LWR@validtablecol}%
6958 }

```

\LWR@parsebeforecolumn Handles >{text} columns.

```
6959 \newcommand*\LWR@parsebeforecolumn{%
```

Move to the next token after the '>':

```
6960 \addtocounter{LWR@tablecolspectindex}{1}%

```

Read the next token, expanding once into \LWR@colparameter:

```

6961 \expandarg%
6962 \StrChar{\LWR@origcolspect}%
6963   {\arabic{LWR@tablecolspectindex}}[\LWR@colparameter]%
6964 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@colparameter:

```

6965 \LWR@setexparray{\LWR@colbeforespec}%
6966   {\arabic{LWR@tabletotalLaTeXcolsnext}}{\LWR@colparameter}%
6967 \let\LWR@colparameter\relax%
6968 \global\booltrue{LWR@validtablecol}%
6969 }

```

\LWR@parseaftercolumn Handles <{text} columns.

```
6970 \newcommand*\LWR@parseaftercolumn{%
```

Move to the next token after the '<':

```
6971 \addtocounter{LWR@tablecolsindex}{1}%
```

Read the next token, expanding once into \LWR@colparameter:

```
6972 \expandarg%
6973 \StrChar{\LWR@origcolspec}%
6974 {\arabic{LWR@tablecolsindex}}[\LWR@colparameter]%
6975 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
6976 \LWR@setexpparray{LWR@colaferspec}%
6977 {\arabic{LWR@tabletotalLaTeXcols}}[\LWR@colparameter]%
6978 \let\LWR@colparameter\relax%
6979 \global\booltrue{LWR@validtablecol}%
6980 }
```

\LWR@parsebarcolumn Handles vertical rules.

```
6981 \newcommand*{\LWR@parsebarcolumn}{%
6982 \LWR@traceinfo{LWR@parsebarcolumn}%
}
```

Remember the bar at this position:

```
6983 \ifnumcomp{\value{LWR@tabletotalLaTeXcols}}{=}{0}%
6984 {%
6985   \edef\LWR@tempone{\LWR@getexpparray{LWR@colbarspec}{leftedge}}%
6986   \ifdefstring{\LWR@tempone}{tvertbarl}%
6987     {\LWR@setexpparray{LWR@colbarspec}{leftedge}{tvertbarldouble}}%
6988     {\LWR@setexpparray{LWR@colbarspec}{leftedge}{tvertbarl}}%
6989 }%
6990 {%
6991   \edef\LWR@tempone{%
6992     \LWR@getexpparray{LWR@colbarspec}{\arabic{LWR@tabletotalLaTeXcols}}%
6993   }%
6994   \ifdefstring{\LWR@tempone}{tvertbarr}%
6995   {%
6996     \LWR@setexpparray{LWR@colbarspec}%
6997       {\arabic{LWR@tabletotalLaTeXcols}}{tvertbarrdouble}}%
6998   }%
6999   {%
7000     \LWR@setexpparray{LWR@colbarspec}%
7001       {\arabic{LWR@tabletotalLaTeXcols}}{tvertbarr}}%
7002   }%
7003 }%
7004 \global\booltrue{LWR@validtablecol}%
}
```

7005 }

\LWR@parsecoloncolumn Handles vertical rules.

7006 \newcommand*\{\LWR@parsecoloncolumn}{%
 7007 \LWR@traceinfo{\LWR@parsecoloncolumn}{%

Remember the bar at this position:

7008 \ifnumcomp{\value{\LWR@tabletotalLaTeXcols}}{=}{0}{%
 7009 { % left edge of the table:
 7010 \edef\LWR@tempone{\LWR@getexpparray{\LWR@colbarspec}{leftedge}}%
 7011 \ifdefstring{\LWR@tempone}{tvertbarldash}{%
 7012 {\LWR@setexpparray{\LWR@colbarspec}{leftedge}{tvertbarldoubledash}}%
 7013 {\LWR@setexpparray{\LWR@colbarspec}{leftedge}{tvertbarldash}}%
 7014 }%
 7015 { % not at the left edge:
 7016 \edef\LWR@tempone{
 7017 \LWR@getexpparray{\LWR@colbarspec}{\arabic{\LWR@tabletotalLaTeXcols}}}%
 7018 }%
 7019 \ifdefstring{\LWR@tempone}{tvertbarrdash}{%
 7020 {\LWR@setexpparray{\LWR@colbarspec}{
 7021 \arabic{\LWR@tabletotalLaTeXcols}{tvertbarrdoubledash}}}%
 7022 {\LWR@setexpparray{\LWR@colbarspec}{
 7023 \arabic{\LWR@tabletotalLaTeXcols}{tvertbarrdash}}}%
 7024 }%
 7025 \global\booltrue{\LWR@validtablecol}%
 7026 }

\LWR@parsingsemicoloncolumn Handles vertical rules.

7027 \newcommand*\{\LWR@parsingsemicoloncolumn}{%

Treat ; as a : column:

7028 \LWR@parsecoloncolumn%

Skip the following width token:

7029 \addtocounter{\LWR@tablecolspecindex}{1}{%
 7030 }

71.9 Parsing 'l', 'c', or 'r' columns

\LWR@parsenormalcolumn {\langle thiscolumn \rangle}

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```

7031 \newcommand*{\LWR@parsenormalcolumn}[1]{%
7032 \addtocounter{\LWR@tabletotalLaTeXcols}{1}%
7033 \addtocounter{\LWR@tabletotalLaTeXcolsnext}{1}%

7034 \LWR@setexpparray{\LWR@tablecolspec}{\arabic{\LWR@tabletotalLaTeXcols}}{#1}%

7035 \LWR@traceinfo{normal column \arabic{\LWR@tabletotalLaTeXcols}: #1}%
7036 \LWR@setexpparray{\LWR@colatspec}{\arabic{\LWR@tabletotalLaTeXcolsnext}}{}%
7037 \LWR@setexpparray{\LWR@colbangspec}{\arabic{\LWR@tabletotalLaTeXcolsnext}}{}%
7038 \LWR@setexpparray{\LWR@colbeforespec}{\arabic{\LWR@tabletotalLaTeXcolsnext}}{}%
7039 \LWR@setexpparray{\LWR@colafterspec}{\arabic{\LWR@tabletotalLaTeXcolsnext}}{}%
7040 \LWR@setexpparray{\LWR@colbarspec}{\arabic{\LWR@tabletotalLaTeXcolsnext}}{}%
7041 \global\booltrue{\LWR@validtablecol}%
7042 }

```

71.10 Parsing ‘p’, ‘m’, or ‘b’ columns

\LWR@parsepcolumn {<thiscolumn>} The width will be ignored.

```
7043 \newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type:

```
7044 \LWR@parsenormalcolumn{#1}%
```

Skips the following width token:

```

7045 \addtocounter{\LWR@tablecolspecindex}{1}%
7046 }
```

71.11 Parsing ‘w’ columns

\LWR@parsewcolumn The width will be ignored.

```
7047 \newcommand*{\LWR@parsewcolumn}{%
```

```

7048 \LWR@columnspeckookahead{1}%
7049 \expandafter\expandafter\expandafter{\LWR@strresulttwo}%

```

Table 10: Tabular baseline

l	p	m	b	r
			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

Skips the following width and alignment tokens:

```
7050 \addtocounter{LWR@tablecolspecindex}{2}%
7051 }
```

71.12 Parsing ‘D’ columns

From the `dcolumn` package.

`\LWR@parseDcolumn {<thiscolumn>}` The three parameters will be ignored.

```
7052 \newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
7053 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
7054 \addtocounter{LWR@tablecolspecindex}{3}%
7055 }
```

71.13 Parsing the column specifications

⚠ tabular baselines

HTML CSS cannot exactly match the L^AT_EX concept of a baseline for a table row. Table 10 shows the L^AT_EX results for various vertical-alignment choices, with the baseline of the first column drawn across all the columns for comparison. See the `p` column specification in table 11 for details.

Table 11 describes how each kind of column is converted to HTML.

Table 11: Tabular HTML column conversions

l, r, c: Converted to table cells without paragraph tags.

Uses css `vertical-align:middle` so that top or bottom-aligned cells may go above or below this cell.

p: Converted to table cells with paragraph tags. Ref: Table 10, LATEX places the top line of a parbox aligned with the rest of the text line, so css `vertical-align:bottom` is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the css may be desired, changing `td.tdp`, `td.tdP`, `td.tdprule`, and `td.tdPrule` to `vertical-align: middle`. Another possibility is to change L,R,C, and P to `vertical-align: top` and not worry about the alignment of B and M cells or trying to approximate LATEX baselines.

m: With paragraph tags, css `vertical-align:middle`.

b: With paragraph tags, css `vertical-align:top` so that the bottom of the text is closest to the middle of the text line.

P, M, B: Horizontally-centered versions.

S: Converted to 'r'. Ignores optional argument. From the `siunitx` package.

D: Converted to 'c'. From the `dcolumn` package.

@, !, >, <: One each, in that order.

|: Vertical rule.

Unknown: Converted to 'l'.

\newcolumn: Currently treated as unknown.

Bool LWR@validtablecol True if found a valid table column type.

```
7056 \newbool{LWR@validtablecol}
```

Bool LWR@opttablecol True if found a table column optional argument.

```
7057 \newbool{LWR@opttablecol}
```

\LWR@parsetablecols {*colspeccs*}

Scans the column specification left to right.

Builds \LWR@tablecols with the final specification, one L^AT_EX column per entry. The final number of L^AT_EX columns in each row is stored in LWR@tabletotalLaTeXcols, which is the number of & and \\ in each line, but which does not include @, !, <, > specifications in the count.

```
7058 \newcommand*{\LWR@parsetablecols}[1]{%
7059 \LWR@traceinfo{\LWR@parsetablecols}%
}
```

Remember the original supplied column spec:

```
7060 \renewcommand*{\LWR@origcols}{#1}%

```

Remove spaces:

```
7061 \expandarg%
7062 \StrSubstitute{\LWR@origcols}{ }{}[\LWR@origcols]%
```

The parsed column spec data array, LWR@tablecols, will be overwritten with new values.

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
7063 \setcounter{LWR@tabletotalLaTeXcols}{0}%
7064 \setcounter{LWR@tabletotalLaTeXcolsnext}{1}%
7065 \LWR@setexpparray{\LWR@colatspec}{leftedge}{}%
7066 \LWR@setexpparray{\LWR@colatspec}{1}{}%
7067 \LWR@setexpparray{\LWR@colatspec}{2}{}%
7068 \LWR@setexpparray{\LWR@colatspec}{3}{}%
7069 \LWR@setexpparray{\LWR@colbangspec}{leftedge}{}%
7070 \LWR@setexpparray{\LWR@colbangspec}{1}{}%
7071 \LWR@setexpparray{\LWR@colbangspec}{2}{}%
7072 \LWR@setexpparray{\LWR@colbangspec}{3}{}%
7073 \LWR@setexpparray{\LWR@colbeforespec}{1}{}%
7074 \LWR@setexpparray{\LWR@colbeforespec}{2}{}%
```

```

7075 \LWR@setexpparray{\LWR@colbeforespec}{3}{}%
7076 \LWR@setexpparray{\LWR@colafterspec}{1}{}%
7077 \LWR@setexpparray{\LWR@colafterspec}{2}{}%
7078 \LWR@setexpparray{\LWR@colafterspec}{3}{}%
7079 \LWR@setexpparray{\LWR@colbarspec}{leftedge}{}%
7080 \LWR@setexpparray{\LWR@colbarspec}{1}{}%
7081 \LWR@setexpparray{\LWR@colbarspec}{2}{}%
7082 \LWR@setexpparray{\LWR@colbarspec}{3}{}%

```

Starting at the first column specification:

```
7083 \setcounter{\LWR@tablecolspecindex}{1}%
```

Place the colspecs string length into \LWR@strresult, and remember the number of characters in the column specification:

```

7084 \expandarg%
7085 \StrLen{\LWR@origcolspect}{\LWR@strresult}%
7086 \fullexpandarg%
7087 \LWR@traceinfo{original column spec length: \LWR@strresult}%
7088 \setcounter{\LWR@tablecolspecwidth}{\LWR@strresult}%

```

Haven't seen any optional arguments so far

```
7089 \global\boolfalse{\LWR@opttablecol}%
```

Scan through the column specifications:

```

7090 \whileboolexpr{%
7091     not test{%
7092         \ifnumcomp{\value{\LWR@tablecolspecindex}}{>}{%
7093             {\value{\LWR@tablecolspecwidth}}{%
7094         }{%
7095     }{%
7096 }{%

```

Place the next single-character column type into \LWR@strresult:

```

7097 \expandarg%
7098 \StrChar{\LWR@origcolspect}{\arabic{\LWR@tablecolspecindex}}{\LWR@strresult}%
7099 \LWR@traceinfo{position \arabic{\LWR@tablecolspecindex}: \LWR@strresult}%
7100 \fullexpandarg%

```

Not yet found a valid column type:

```
7101 \global\boolfalse{\LWR@validtablecol}%
```

Skip over any optional arguments, such as siunitx S column:

```
7102 \IfStrEq{\LWR@strresult}{[]}{\global\booltrue{LWR@opttablecol}}{}%
```

Throw away anything found inside the optional argument:

```
7103 \ifbool{LWR@opttablecol}%
7104 {}% inside an optional argument
7105 {}% not an optional tabular argument
```

Not inside an optional argument, so consider the column type:

```
7106 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{}%
7107 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{}%
7108 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{}%
7109 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{}%
7110 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{}%
7111 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{}%
7112 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{}%

7113 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{c}}{}%
7114 \IfStrEq{\LWR@strresult}{s}{\LWR@parsenormalcolumn{c}}{}%

7115 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
7116 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
7117 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
7118 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
7119 \IfStrEq{\LWR@strresult}{|}{\LWR@parsebarcolumn}{}%
7120 \IfStrEq{\LWR@strresult}{:}{\LWR@parsecoloncolumn}{}%
7121 \IfStrEq{\LWR@strresult}{;}{\LWR@parsesemicoloncolumn}{}%

7122 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
7123 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
7124 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%

7125 \IfStrEq{\LWR@strresult}{w}{\LWR@parsewcolumn}{}%
7126 \IfStrEq{\LWR@strresult}{W}{\LWR@parsewcolumn}{}%
```

From the dcolumn package:

```
7127 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{}%
```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```
7128 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%
```

Many people define centered versions “P”, “M”, and “B”:

```
\newcolumntype{P}[1]{>{\centering\arraybackslash}p{#1}}
```

```
7129 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
7130 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
7131 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%
```

If this column was an invalid column type, convert it to an l column:

```
7132 \ifbool{\LWR@validtablecol}{%
7133   \LWR@traceinfo{invalid column type: \LWR@strresult}%
7134   \LWR@parsenormalcolumn{l}%
7135 }%
7136 }% not an optional column argument
```

If read the closing bracket, no longer inside the optional argument:

```
7137 \IfStrEq{\LWR@strresult}{}{\global\boolearn{\LWR@opttablecol}}%
```

Move to the next character:

```
7138 \addtocounter{\LWR@tablecolsindex}{1}%
7139 }% whiledo
7140 }%
```

71.14 colortbl and xparse tabular color support

These macros provide a minimal emulation of some colortbl macros which might appear between table cells. If colortbl is loaded, these macros will be replaced with functional versions.

For each of the HTML colors below, the text for the HTML color is set if requested, but the macro is empty if none has been set.

`\rownum` Reserve a counter register.

```
7141 @ifundefined{rownum}{\newcount\rownum}{}%
```

`\@rowcolors` Emulated in case xcolor is not used.

```
7142 \newcommand*\@rowcolors{}
```

\@rowcolors Emulated in case xcolor is not used.

7143 \newcommand*{\@rowcolors}{}{}

\LWR@xcolorrowHTMLcolor Emulated xcolor row color.

7144 \newcommand*{\LWR@xcolorrowHTMLcolor}{}{}

\LWR@columnHTMLcolor HTMLstyle code for the column color.

7145 \def\LWR@columnHTMLcolor{}{}

\LWR@rowHTMLcolor HTMLstyle code for the row color.

7146 \def\LWR@rowHTMLcolor{}{}

\LWR@cellHTMLcolor HTMLstyle code for the cell color.

7147 \def\LWR@cellHTMLcolor{}{}

\LWR@ruleHTMLcolor HTMLstyle code for the cell color.

7148 \newcommand*{\LWR@ruleHTMLcolor}{}{}

\rowcolor [⟨model⟩] {⟨color⟩} [⟨left overhang⟩] [⟨right overhang⟩] Print version. The HTML version is in lwarf-colortbl. Used before starting a tabular data cell, thus \LWR@getmynexttoken.

7149 \newcommand*{\rowcolor}{\LWR@getmynexttoken} %

\arrayrulecolor [⟨model⟩] {⟨color⟩}

\arrayrulecolornexttoken [⟨model⟩] {⟨color⟩}

Print versions for use outside and inside a tabular:

7150 \newcommand{\arrayrulecolor}[2][named]{}{}

7151 \newcommand{\arrayrulecolornexttoken}[2][named]{\LWR@getmynexttoken}

\doublerulesepcolor [⟨model⟩] {⟨color⟩}

```
\doublerulesepcolornexttoken [<model>] {<color>}
```

Print versions for use inside and outside a tabular:

```
7152 \newcommand{\doublerulesepcolor}[2][named]{}
7153 \newcommand{\doublerulesepcolornexttoken}[2][named]{\LWR@getmynexttoken}
```

71.15 Starting a new row

\LWR@maybenewtablerow If have not yet started a new table row, begin one now. Creates a new row tag, adding a class for hline or tbrule if necessary.

```
7154 \newcommand*\LWR@maybenewtablerow{%
7155 {%
7156 \ifbool{\LWR@startedrow}{%
7157 }% started the row
7158 }% not started the row
```

Remember that now have started the row:

```
7159 \global\booltrue{\LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```
7160 \global\booltrue{\LWR@intabularmetadata}%
7161 \ifboolexpr{%
7162     \test{\ifnumcomp{\value{\LWR@hlines}}{>}{0}}\or%
7163     \test{\ifnumcomp{\value{\LWR@hdashedlines}}{>}{0}}%
7164 }%
7165 {%
7166     \LWR@htmlltag{tr class="hline"}%
7167     \LWR@orignewline%
7168 }%
7169 {%
7170     \ifbool{\LWR@doingtbrule}{%
7171         \ifdefvoid{\LWR@ruleHTMLcolor}{%
7172             \LWR@htmlltag{tr class="tbrule"}%
7173         }{%
7174             \LWR@htmlltag{%
7175                 tr class="tbrule" % space
7176                 style="border-top: 1px solid % space
7177                     \LWR@origpound\LWR@ruleHTMLcolor "%%
7178             }%
7179         }%
7180     }%
7181     \LWR@orignewline%
```

```

7182      }%
7183      {\LWR@htmntag{tr}\LWR@orignewline}%
7184      }% end of not doing hline
7185 }% end of not started the row
7186 }

```

71.16 Printing vertical bar tags

\LWR@printbartag {<index>}

Adds to a tabular data cell an HTML class name for a left/right vertical bar.

```

7187 \newcommand*{\LWR@printbartag}[1]{%
7188 \LWR@traceinfo{\LWR@printbartag !#1!}%
7189 \ifboolexpr{bool{\LWR@tabularmutemods} or bool{\LWR@emptyatbang}}{%
7190 }% muting or empty
7191 }% not muting
7192     \edef\tempone{\LWR@getexparray{\LWR@colbarspec}{#1}}%
7193     \ifdefempty{\tempone}{}{\LWR@tempone}%
7194 }% not muting
7195 \LWR@traceinfo{\LWR@printbartag done}%
7196 }

```

71.17 Printing at or bang tags

\LWR@printatbang {<at-or-bang>} {<index>}

```
7197 \newcommand*{\LWR@printatbang}[2]{%
```

Fetch the column at or bang spec:

```

7198 \xdef\LWR@atbangspec{\LWR@getexparray{\LWR@col#1spec}{#2}}%
7199 \LWR@traceinfo{atbang: #2 !\LWR@atbangspec!}%

```

Only generate if is not empty;

```

7200 \ifdefempty{\LWR@atbangspec}{%
7201 }%
7202 }% not empty
7203     \LWR@htmntag{%
7204         td class="td#1%
7205             \LWR@subaddcmidruletrim{}{}%
7206             \LWR@printbartag{#2}%

```

```

7207      "%
7208      \LWR@tdstartstyles%
7209      \LWR@addcmidrulewidth%
7210      \LWR@addcdashline%
7211      \LWR@addtabularrulecolors%
7212      \LWR@tdendstyles%
7213  }%

```

Create an empty cell if muting for the \bottomrule:

```

7214      \ifboolexpr{bool{\LWR@tabularmutemods} or bool{\LWR@emptyatbang}}{%
7215      {}%
7216      {\LWR@atbangspec}%
7217 }%
7218      \LWR@htmlltag{/td}\LWR@orignewline%
7219      \global\booltrue{\LWR@tabularcelladded}%
7220 }% not empty
7221 }%

```

\LWR@addleftmostbartag

```

7222 \newcommand*{\LWR@addleftmostbartag}{%
7223 \ifnumcomp{\value{\LWR@tableLaTeXcolindex}}{=}{1}{%
7224     \LWR@printbartag{leftedge}%
7225 }{}%
7226 }

```

\LWR@tabularleftedge

```

7227 \newcommand*{\LWR@tabularleftedge}{%
7228 \ifnumcomp{\value{\LWR@tableLaTeXcolindex}}{=}{1}{%
7229 {}%
7230     \LWR@printatbang{at}{leftedge}%
7231     \LWR@printatbang{bang}{leftedge}%
7232 }% left edge
7233 {}% not left edge
7234 }

```

71.18 Data opening tag

\LWR@thiscolspec Temporary storage.

```
7235 \newcommand*{\LWR@thiscolspec}{}%
```

\LWR@tabledatasinglecolumntag Print a table data opening tag with style for alignment and color.

```
7236 \newcommand*{\LWR@tabledatasinglecolumntag}{%
7237 {%
7238 \LWR@traceinfo{\LWR@tabledatasinglecolumntag}%
7239 \LWR@maybenewtablerow%
```

Don't start a new paragraph tag if have already started one:

```
7240 \ifbool{\LWR@intabularmetadata}{%
7241 {%
```

If have found the end of tabular command, do not create the next data cell:

```
7242     \ifbool{\LWR@existingtabular}{%
7243     {%
7244     \LWR@tabularleftedge%
```

Print the @ and ! contents before first column:

```
7244     \LWR@tabularleftedge%
```

Fetch the current column's alignment character into \LWR@strresult:

```
7245     \xdef\LWR@strresult{%
7246         \LWR@getexpparray{\LWR@tablecolspe}{{\arabic{\LWR@tableLaTeXcolindex}}}}%
7247 }
```

print the start of a new table data cell:

```
7248 \LWR@traceinfo{\LWR@tabledatasinglecolumntag: about to print td tag}%
7249     \LWR@htmlltag{td class="td"}
```

append this column's spec:

```
7250     \LWR@strresult%
```

If this column has a cmidrule, add "rule" to the end of the HTML class tag. Also add vertical bar tags.

```
7251     \LWR@addcmidruletrim%
7252     \LWR@addleftmostbartag%
7253     \LWR@printbartag{{\arabic{\LWR@tableLaTeXcolindex}}}%
7254     "%
```

Add styles for rules, alignment:

```
7255     \LWR@tdstartstyles%
```

```

7256      \LWR@addcmidrulewidth%
7257      \LWR@addcdashline%

7258      \xdef\LWR@thiscolspec{%
7259          \LWR@getexpparray{\LWR@tablecolspec}{\arabic{\LWR@tableLaTeXcolindex}}%
7260      }%
7261      \LWR@addformatwpalignment{\LWR@thiscolspec}%

```

Add styles for cell and rule colors:

```

7262      \LWR@addtabularrowcolor%
7263      \LWR@addtabularrulecolors%

7264      \LWR@tdendstyles%
7265  }% HTML td
7266      \LWR@traceinfo{\LWR@tabledatasinglecolumntag: done printing td tag}%

```

If this is a p, m, b, or X column, allow paragraphs:

```

7267      \ifboolexpr{%
7268          test{ \ifdefstring{\LWR@strresult}{p} } or
7269          test{ \ifdefstring{\LWR@strresult}{m} } or
7270          test{ \ifdefstring{\LWR@strresult}{b} } or
7271          test{ \ifdefstring{\LWR@strresult}{P} } or
7272          test{ \ifdefstring{\LWR@strresult}{M} } or
7273          test{ \ifdefstring{\LWR@strresult}{B} } or
7274          test{ \ifdefstring{\LWR@strresult}{X} }%
7275      }%
7276  {%
7277      \allowpars
7278      \LWR@traceinfo{\LWR@tabledatasinglecolumntag: about to \LWR@startpars}%
7279      \global\booltrue{\LWR@tableparcell}%
7280      \LWR@startpars
7281      \LWR@traceinfo{\LWR@tabledatasinglecolumntag: done with \LWR@startpars}%
7282  }%
7283  {}%
7284  }%

```

Print the > contents unless muted for the \bottomrule:

```

7283      \ifboolexpr{bool{\LWR@tabularmutemods} or bool{\LWR@emptyatbang}}{%
7284      }%
7285  {%
7286      \LWR@getexpparray{\LWR@colbeforespec}{\arabic{\LWR@tableLaTeXcolindex}}%
7287  }%
7288  \global\boolfalse{\LWR@intabularmetadata}%
7289  }% not exiting tabular
7290 }{}% in tabular metadata
7291 \LWR@traceinfo{\LWR@tabledatasinglecolumntag: done}%
7292 }%

```

71.19 Midrules

LWR@midrules LWR@midrules is a data array (section 42) of columns each containing a non-zero width if a midrule should be created for this column.

LWR@trimlrules LWR@trimlrules is a data array (section 42) of columns containing 1 if a midrule should be left trimmed for each column.

LWR@trimrrules LWR@trimrrules is a data array (section 42) of columns containing r if a midrule should be right trimmed for each column.

LWR@cdashlines LWR@cdashlines is a data array (section 42) of columns each containing a Y if an arydshln package "cdashed line" should be created for this column.

Ctr LWR@midrulecounter Indexes across the LWR@midrules and LWR@trim<l/r>rules data arrays.

```
7293 \newcounter{LWR@midrulecounter}
```

Len \LWR@heavyrulewidth The default width of the rule.

```
7294 \newlength{\LWR@heavyrulewidth}
7295 \setlength{\LWR@heavyrulewidth}{.08em}
```

Len \LWR@lightrulewidth The default width of the rule.

```
7296 \newlength{\LWR@lightrulewidth}
7297 \setlength{\LWR@lightrulewidth}{.05em}
```

Len \LWR@cmidrulewidth The default width of the rule.

```
7298 \newlength{\LWR@cmidrulewidth}
7299 \setlength{\LWR@cmidrulewidth}{.03em}
```

Len \LWR@thiscmidrulewidth The width of the next rule, defaulting to \LWR@cmidrulewidth.

If not \LWR@cmidrulewidth, a style will be used to generate the custom width.

Assigned from the LWR@midrules array.

```
7300 \newlength{\LWR@thiscmidrulewidth}
7301 \setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}
```

\LWR@clearmidrules Start new midrules. Called at beginning of tabular and also at \\.

Clears all LWR@midrules and LWR@trimrules markers for this line.

```
7302 \newcommand*{\LWR@clearmidrules}
```

```

7303 {%
7304 \setcounter{LWR@midrulecounter}{1}%
7305 \whileboolexpr{%
7306   not test{%
7307     \ifnumcomp{\value{LWR@midrulecounter}}{>}{}
7308     {\value{LWR@tabletotalLaTeXcols}}%
7309   }%
7310 }%
7311 {%
7312   \LWR@setexpparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{0pt}%
7313   \setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
7314   \LWR@setexpparray{LWR@trimlrules}{\arabic{LWR@midrulecounter}}{}%
7315   \LWR@setexpparray{LWR@trimrrules}{\arabic{LWR@midrulecounter}}{}%
7316   \LWR@setexpparray{LWR@cdashlines}{\arabic{LWR@midrulecounter}}{N}%
7317   \addtocounter{LWR@midrulecounter}{1}%
7318 }%
7319 }

```

\LWR@subcmidrule {*width*} {*trim*} {*leftcolumn*} {*rightcolumn*}

Marks LWR@midrules data array elements to be non-zero widths from left to right columns. Also marks trimming for the L and/or R columns.

LWR@doingcmidrule is set to force an empty row at the end of the tabular to create the rule.

```

7320 \newcommand*{\LWR@subcmidrule}[4]{%
7321 \setcounter{LWR@midrulecounter}{#3}%
7322 \whileboolexpr{%
7323   not test {%
7324     \ifnumcomp{\value{LWR@midrulecounter}}{>}{#4}{}
7325   }%
7326 }%
7327 {%
7328   \LWR@setexpparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{#1}%
7329   \addtocounter{LWR@midrulecounter}{1}%
7330 }% whiledo
7331 \IfSubStr{#2}{1}{\LWR@setexpparray{LWR@trimlrules}{#3}{1}}{%
7332 \IfSubStr{#2}{r}{\LWR@setexpparray{LWR@trimrrules}{#4}{r}}{%
7333 \global\booltrue{LWR@doingcmidrule}}%
7334 }

```

\LWR@docmidrule [*width*] (*trim*) {*leftcolumn-rightcolumn*}

Marks LWR@midrules array elements to be a non-zero width from left to right columns. Also marks trimming for the L and/or R columns.

```
7335 \NewDocumentCommand{\LWR@docmidrule}
```

```
7336 {0{\LWR@cmidrulewidth} D(){} >{\SplitArgument{1}{-}m}
7337 {\LWR@subcmidrule{#1}{#2}#3}
```

\LWR@subcdashline {⟨leftcolumn⟩} {⟨rightcolumn⟩}

Marks LWR@cdashlines data array elements to be Y from left to right columns.

LWR@doingcmidrule is set to force an empty row at the end of the tabular to create the rule.

```
7338 \newcommand*{\LWR@subcdashline}[2]{%
7339 \setcounter{LWR@midrulecounter}{#1}%
7340 \whileboolexpr{%
7341 not test {%
7342 \ifnumcomp{\value{LWR@midrulecounter}}{>}{#2}%
7343 }%
7344 }%
7345 {%
7346 \LWR@setexpparray{LWR@cdashlines}{\arabic{LWR@midrulecounter}}{Y}%
7347 \addtocounter{LWR@midrulecounter}{1}%
7348 }% whiledo
7349 \global\booltrue{LWR@doingcmidrule}%
7350 }
```

\LWR@docdashline {⟨leftcolumn-rightcolumn⟩}

Marks LWR@cdashlines data array elements to be Y from left to right columns.

```
7351 \NewDocumentCommand{\LWR@docdashline}{%
7352 >{\SplitArgument{1}{-}m}%
7353 {%
7354 \LWR@subcdashline#1%
7355 }}
```

Used to compute margins, tabular trims, column offsets:

```
7356 \newlength{\LWR@templengthone}
7357 \newlength{\LWR@templengthtwo}
7358 \newlength{\LWR@templengththree}
7359 \newcounter{LWR@tempcountone}
```

Used to add a style to a table data cell:

```
7360 \newbool{LWR@tdhavecellstyle}
```

\LWR@tdstartstyles Begins possibly adding a table data cell style.

```
7361 \newcommand*{\LWR@tdstartstyles}{\global\boolfalse{LWR@tdhavecellstyle}}
```

\LWR@tdaddstyle Starts adding a table data cell style.

```
7362 \newcommand*{\LWR@tdaddstyle}{%
7363 \ifbool{LWR@tdhavecellstyle}{%
7364 {; }%
7365 { style=""}%
7366 \booltrue{LWR@tdhavecellstyle}%
7367 }
```

\LWR@tdendstyles Finishes possibly adding a table data cell style. Prints the closing quote.

```
7368 \newcommand*{\LWR@tdendstyles}{%
7369 \ifbool{LWR@tdhavecellstyle}{%
7370   "%%
7371   \global\boolfalse{LWR@tdhavecellstyle}%
7372 }{}%
7373 }
```

\LWR@subaddcmidruletrim {<lefttrim>} {<righttrim>} Adds a \cmidrule with optional trim.

```
7374 \newcommand*{\LWR@subaddcmidruletrim}[2]{%
7375 \setlength{\LWR@templengthone}{%
7376   \LWR@getexparray[LWR@midrules]{\arabic{LWR@tableLaTeXcolindex}}%
7377 }%
7378 \ifdimcomp{\LWR@templengthone}{>}{0pt}%
7379 {%
```

Print the class with left and right trim letters appended:

```
7380 \LWR@origtilde tdrule#1#2%
```

Remember the width of the rule:

```
7381 \setlength{\LWR@thiscmidrulewidth}{\LWR@templengthone}%
7382 }%
7383 {%
7384 \setlength{\LWR@thiscmidrulewidth}{0pt}%
7385 }%
7386 }
```

\LWR@addcmidruletrim Adds left or right trim to a \cmidrule.

```
7387 \newcommand*{\LWR@addcmidruletrim}{%
7388 \LWR@subaddcmidruletrim}%
```

```
7389 {\LWR@getexparray{\LWR@trimlrules}{\arabic{\LWR@tableLaTeXcolindex}}}%  
7390 {\LWR@getexparray{\LWR@trimrrules}{\arabic{\LWR@tableLaTeXcolindex}}}%  
7391 }
```

\LWR@addrulewidth {*thiswidth*} {*defaultwidth*}

If not default width, add a custom style with width and color depending on *thiswidth*.

Must be placed between \LWR@tdstartstyles and \LWR@tdendstyles.

```
7392 \newcommand{\LWR@addrulewidth}[2]{%
```

Only add a custom width if *thiswidth* is different than the *defaultwidth*, or if a color is being used:

```
7393 \ifboolexpr{  
7394     test{\ifdimcomp{#1}{=}{0pt} or  
7395     (  
7396         ( test{\ifdimcomp{#1}{=}{#2} and not bool{FormatWP} )  
7397         and ( test {\ifdefvoid{\LWR@ruleHTMLcolor}} )  
7398     )  
7399 }%  
7400 {}% default width and color  
7401 {}% custom width and/or color
```

Ensure that the width is wide enough to display in the browser:

```
7402 \LWR@forceminwidth{#1}%
```

Begin adding another style:

```
7403 \LWR@tdaddstyle%
```

The style itself:

```
7404 border-top:\LWR@printlength{\LWR@atleastonept} solid %
```

If default gray, the darkness of the color depends on the thickness of the rule:

```
7405 \ifdefvoid{\LWR@ruleHTMLcolor}{%  
7406     \ifdimcomp{#1}{<}{\LWR@lightrulewidth}{%  
7407         {\LWR@origpound{}AOAOAO}{%  
7408             \% lightrule or heaver  
7409             \ifdimcomp{#1}{<}{\LWR@heavyrulewidth}{%  
7410                 {\LWR@origpound{}808080}{%  
7411                     {black}{%
```

```

7412      }% lightrule or heavier
7413      }{%
7414          \LWR@origpound\LWR@ruleHTMLcolor%
7415      }
7416 }% custom width and/or color
7417 }

```

`\LWR@addcmidrulewidth` Adds a style for the rule width.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```

7418 \newcommand{\LWR@addcmidrulewidth}{%
7419 \LWR@addrulewidth{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
7420 }

```

`\LWR@addcdashline` Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```

7421 \newcommand{\LWR@addcdashline}{%
7422 \edef\LWR@tempone{%
7423     \LWR@getexpparray{\LWR@cdashlines}{\arabic{LWR@tableLaTeXcolindex}}%
7424 }%
7425 \ifdefstring{\LWR@tempone}{Y}{%
7426     \LWR@tdaddstyle%
7427     border-top: 1pt dashed %
7428     \ifdefvoid{\LWR@ruleHTMLcolor}%
7429         {black}%
7430         {\LWR@origpound\LWR@ruleHTMLcolor}%
7431 }{}%
7432 }

```

`\LWR@WPcell` $\{\langle text-align \rangle\} \{\langle vertical-align \rangle\}$

```

7433 \newcommand*{\LWR@WPcell}[2]{%
7434 \LWR@tdaddstyle%
7435 \LWR@print@mbox{text-align:#1}; \LWR@print@mbox{vertical-align:#2}%
7436 }

```

`\LWR@addformatwpalignment` If FormatWP, adds a style for the alignment.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```

7437 \newcommand*{\LWR@addformatwpalignment}[1]{%
7438 \ifbool{FormatWP}{%
7439 \IfSubStr{#1}{1}{\LWR@WPcell{left}{middle}}{}%
7440 \IfSubStr{#1}{c}{\LWR@WPcell{center}{middle}}{}%
7441 \IfSubStr{#1}{r}{\LWR@WPcell{right}{middle}}{}%

```

```

7442 \IfSubStr{#1}{p}{\LWR@WPcell{left}{bottom}}{}%
7443 \IfSubStr{#1}{m}{\LWR@WPcell{left}{middle}}{}%
7444 \IfSubStr{#1}{b}{\LWR@WPcell{left}{top}}{}%
7445 \IfSubStr{#1}{P}{\LWR@WPcell{center}{bottom}}{}%
7446 \IfSubStr{#1}{M}{\LWR@WPcell{center}{middle}}{}%
7447 \IfSubStr{#1}{B}{\LWR@WPcell{center}{top}}{}%
7448 }{}%
7449 }

```

71.20 Cell colors

\LWR@addtabulararrowcolor Adds a cell's row color style, if needed.

No color is added for the final row of empty cells which finishes each tabular.

```

7450 \newcommand*\LWR@addtabulararrowcolor{%
7451 \ifbool{\LWR@tabularmutemods}{}{%
7452   \ifdefvoid{\LWR@rowHTMLcolor}{%
7453     \ifdefvoid{\LWR@xcolorrowHTMLcolor}{%
7454       \% xcolor row color
7455       \LWR@tdaddstyle%
7456       background:\LWR@origpound\LWR@xcolorrowHTMLcolor%
7457     }%
7458   }%
7459   \% explicit row color
7460   \LWR@tdaddstyle%
7461   background:\LWR@origpound\LWR@rowHTMLcolor%
7462 }%
7463 }%
7464 }

```

\LWR@addtabularhrulecolor Adds a cell's horizontal rule color style, if needed.

```
7465 \newcommand*\LWR@addtabularhrulecolor{%
```

If either form of horizontal rule is requested:

```

7466 \ifboolexpr{%
7467   test{\ifnumcomp{\value{\LWR@hlines}}{>}{0}} or%
7468   test{\ifnumcomp{\value{\LWR@hdashedlines}}{>}{0}} or%
7469   bool{\LWR@doingtbrule}%
7470 }{%

```

If there is no custom color:

```

7471     \ifdefvoid{\LWR@ruleHTMLcolor}%
7472     {%
7473         \ifnumcomp{\value{LWR@hlines}}{>}{1}%
7474         {%
7475             \LWR@tdaddstyle%
7476             border-top: 4px double%
7477         }{%
7478             \ifnumcomp{\value{LWR@hdashedlines}}{>}{1}%
7479             {%
7480                 \LWR@tdaddstyle%
7481                 border-top: 2px dashed%
7482             }{%
7483                 \ifnumcomp{\value{LWR@hdashedlines}}{=}{1}%
7484                 {%
7485                     \LWR@tdaddstyle%
7486                     border-top: 1px dashed%
7487                 }{}}%}

```

If no color and not doubled or dashed, then add nothing, since a simpler rule is the default.

```
7488     }%
```

If there is a custom color:

```

7489     {%
7490         \ifnumcomp{\value{LWR@hlines}}{>}{1}%
7491         {%
7492             \LWR@tdaddstyle%
7493             border-top: 4px double \LWR@origpound\LWR@ruleHTMLcolor%
7494         }{%
7495             \ifnumcomp{\value{LWR@hdashedlines}}{>}{1}%
7496             {%
7497                 \LWR@tdaddstyle%
7498                 border-top: 2px dashed \LWR@origpound\LWR@ruleHTMLcolor%
7499             }{%
7500                 \ifnumcomp{\value{LWR@hdashedlines}}{=}{1}%
7501                 {%
7502                     \LWR@tdaddstyle%
7503                     border-top: 1px dashed \LWR@origpound\LWR@ruleHTMLcolor%
7504                 }{%
7505                     \LWR@tdaddstyle%
7506                     border-top: 1px solid \LWR@origpound\LWR@ruleHTMLcolor%
7507                 }{}}%}
7508     }%
7509 }{}}%
7510 }

```

`\LWR@addtabularrulecolors` Adds a cell's rule color styles, if needed.

No color is added for the final row of empty cells which finishes each tabular.

```
7511 \newcommand*{\LWR@addtabularrulecolors}{%
```

Custom horizontal rule color:

```
7512 \LWR@addtabularhrulecolor%
```

No vertical rules if finishing the tabular with a row of empty cells:

```
7513 \ifbool{\LWR@tabularmutemods}{ }{%
```

If at the leftmost cell, possibly add a leftmost vertical rule:

```
7514 \ifnumequal{\value{\LWR@tableTeXcolindex}}{1}{%
```

Fetch the left edge's vertical bar specification:

```
7515 \edef\tempone{\LWR@getexarray{\LWR@colbarspec}{leftedge}}%
```

Add a custom style if a vertical bar was requested:

```
7516 \ifdefstring{\tempone}{tvertbarl}{%
7517   \LWR@tdaddstyle%
7518   border-left: 1px solid % space
7519   \LWR@vertruleHTMLcolor%
7520 }
7521 \ifdefstring{\tempone}{tvertbarldouble}{%
7522   \LWR@tdaddstyle%
7523   border-left: 4px double % space
7524   \LWR@vertruleHTMLcolor%
7525 }
7526 \ifdefstring{\tempone}{tvertbarldash}{%
7527   \LWR@tdaddstyle%
7528   border-left: 1px dashed % space
7529   \LWR@vertruleHTMLcolor%
7530 }
7531 \ifdefstring{\tempone}{tvertbarldoubledash}{%
7532   \LWR@tdaddstyle%
7533   border-left: 2px dashed % space
7534   \LWR@vertruleHTMLcolor%
7535 }
7536 }
```

Possibly add a right vertical rule for this cell:

```
7537 \edef\tempone{%
7538   \LWR@getexarray{\LWR@colbarspec}{\arabic{\LWR@tableTeXcolindex}}}%
```

```
7539 }%  
7540 \ifdefstring{\LWR@tempone}{tvertbarr}{%
```

Add a custom style if a vertical bar was requested:

```
7541             \LWR@taddstyle%
7542             border-right: 1px solid \LWR@vertruleHTMLcolor%
7543         }{%
7544         \ifdefstring{\LWR@tempone}{tvertbarrdouble}{%
7545             \LWR@taddstyle%
7546             border-right: 4px double \LWR@vertruleHTMLcolor%
7547         }{%
7548         \ifdefstring{\LWR@tempone}{tvertbarrdash}{%
7549             \LWR@taddstyle%
7550             border-right: 1px dashed \LWR@vertruleHTMLcolor%
7551         }{%
7552         \ifdefstring{\LWR@tempone}{tvertbarrdoubledash}{%
7553             \LWR@taddstyle%
7554             border-right: 2px dashed \LWR@vertruleHTMLcolor%
7555         }{%
7556     }%
7557 }
```

Ctr LWR@cellcolordepth Counts how many cell color <div>s were added to the current tabular data cell.

7558 \newcounter{LWR@cellcolordepth}

\LWR@subaddtabularcellcolor {<HTML color>}

```
7559 \newcommand*{\LWR@subaddtabularcellcolor}[1]{%
7560 \LWR@htmltag{div class="cellcolor" style="%"
7561     background:\LWR@origpound{}{}#1 %
7562 " }%
7563 \addtocounter{\LWR@cellcolordepth}{1}%
7564 }
```

\LWR@addtabularcellcolor Adds a cell color style, if needed.

```

7574     {\LWR@subaddtabularcellcolor{\LWR@columnHTMLcolor}}%
7575   }%
7576   {\LWR@subaddtabularcellcolor{\LWR@xcolorrowHTMLcolor}}%
7577 }%
7578 {\LWR@subaddtabularcellcolor{\LWR@rowHTMLcolor}}%
7579 }%
7580 {\LWR@subaddtabularcellcolor{\LWR@cellHTMLcolor}}%
7581 }

```

71.21 Multicolumns

71.21.1 Parsing multicolumns

7582 \newcounter{LWR@tablemulticolswidth}

Indexes into the multicolumn specification:

7583 \newcounter{LWR@tablemulticolspos}

Remembers multicolumn vertical rules if found in the column spec.

```

7584 \newcounter{LWR@mcolvertbarsl}
7585 \newcounter{LWR@mcolvertbarsr}
7586 \newcounter{LWR@mcolvertbarsldash}
7587 \newcounter{LWR@mcolvertbarsrdash}
7588 \newbool{LWR@mcolvertbaronleft}%

```

\LWR@printmccoltype {<cols>} 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.

```

7589 \newcommand*{\LWR@printmccoltype}[1]{%
7590 \LWR@traceinfo{lwr@printmccoltype -#1-}%

```

Get one token of the column spec:

7591 \StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%

Add to the HTML tag depending on which column type is found:

```

7592 \IfStrEq{\LWR@strresult}{l}{\{}%
7593 \IfStrEq{\LWR@strresult}{c}{\{}%
7594 \IfStrEq{\LWR@strresult}{r}{\{}%
7595 \IfStrEq{\LWR@strresult}{p}{\{}%

```

```

7596 \IfStrEq{\LWR@strresult}{m}{\}%
7597 \IfStrEq{\LWR@strresult}{b}{\b}{\}%
7598 \IfStrEq{\LWR@strresult}{P}{\P}{\}%
7599 \IfStrEq{\LWR@strresult}{M}{\M}{\}%
7600 \IfStrEq{\LWR@strresult}{B}{\B}{\}%

7601 \IfStrEq{\LWR@strresult}{w}{\w}{\}%
7602 \IfStrEq{\LWR@strresult}{W}{\W}{\}%

7603 \IfStrEq{\LWR@strresult}{S}{\c}{\}%
7604 \IfStrEq{\LWR@strresult}{s}{\c}{\}%

7605 \IfStrEq{\LWR@strresult}{X}{\p}{\}%

7606 \IfStrEq{\LWR@strresult}{|}{\}%
7607 {%
7608     \ifbool{\LWR@mcolvertbaronleft}{%
7609         {\addtocounter{\LWR@mcolvertbarsl}{1}}% left edge
7610         {\addtocounter{\LWR@mcolvertbarsr}{1}}% not left edge
7611     }%
7612 {%
7613     \IfStrEq{\LWR@strresult}{:}{\}%
7614     {%
7615         \ifbool{\LWR@mcolvertbaronleft}{%
7616             {\addtocounter{\LWR@mcolvertbarsldash}{1}}% left edge
7617             {\addtocounter{\LWR@mcolvertbarsrdash}{1}}% not left edge
7618         }%
7619     {%
7620         \IfStrEq{\LWR@strresult}{;}{\}%
7621         {%
7622             \ifbool{\LWR@mcolvertbaronleft}{%
7623                 {\addtocounter{\LWR@mcolvertbarsldash}{1}}% left edge
7624                 {\addtocounter{\LWR@mcolvertbarsrdash}{1}}% not left edge
7625             }%
7626             {\boolfalse{\LWR@mcolvertbaronleft}}%
7627         }%
7628     }%
7629 \LWR@traceinfo{lwr@printmccoltype done}%
7630 }

```

\LWR@multicolpartext {⟨num parameters⟩} Print the data with paragraph tags, advance to bypass the given number of parameters.

```

7631 \newcommand*{\LWR@multicolpartext}[1]{%
7632 \LWR@startpars%
7633 \LWR@multicoltext%
7634 \addtocounter{\LWR@tablemulticolspos}{#1}%

```

```
7635 \LWR@stoppars%
7636 }
```

\LWR@multicolother {*colspec*} For @, !, >, <, print the next token without paragraph tags:

```
7637 \newcommand*\LWR@multicolother}[1]{%
7638 \addtocounter{LWR@tablemulticolspos}{1}%
7639 \StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
7640 \LWR@strresult%
```

A valid column data type was found:

```
7641 \global\booltrue{LWR@validtablecol}%
7642 }
```

\LWR@multicolskip Nothing to print for this column type.

```
7643 \newcommand*\LWR@multicolskip}{%
```

A valid column data type was found:

```
7644 \global\booltrue{LWR@validtablecol}%
7645 }
```

\LWR@printmccoldata {*colspec*} Print the data for any valid column type found.

```
7646 \newcommand*\LWR@printmccoldata}[1]{%
7647 \LWR@traceinfo{lwr@printmccoldata -#1}%
```

Not yet found a valid column type:

```
7648 \global\boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
7649 \StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
7650 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
7651 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
7652 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
7653 \IfStrEq{\LWR@strresult}{D}{}%
7654 \addtocounter{LWR@tablemulticolspos}{3}%
7655 \LWR@multicoltext%
7656 }{}
```

```

7657 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolpartext{0}}{}%
7658 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolpartext{0}}{}%
7659 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolpartext{0}}{}%
7660 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolpartext{0}}{}%
7661 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolpartext{0}}{}%
7662 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolpartext{0}}{}%

7663 \IfStrEq{\LWR@strresult}{w}{\LWR@multicolpartext{3}}{}%
7664 \IfStrEq{\LWR@strresult}{W}{\LWR@multicolpartext{3}}{}%

7665 \IfStrEq{\LWR@strresult}{S}{\LWR@multicoltext}{}%
7666 \IfStrEq{\LWR@strresult}{s}{\LWR@multicoltext}{}%

7667 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolpartext{0}}{}%
7668 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
7669 \IfStrEq{\LWR@strresult}{:}{\LWR@multicolskip}{}%
7670 \IfStrEq{\LWR@strresult}{;}{\%
7671     \LWR@multicolskip%
7672     \addtocounter{LWR@tablemulticolspos}{1}%
7673 }{}%
7674 \IfStrEq{\LWR@strresult}{@}{\detokenize{@}}{\LWR@multicolother{#1}}{}%
7675 \IfStrEq{\LWR@strresult}{!}{\detokenize{!}}{\LWR@multicolother{#1}}{}%
7676 \IfStrEq{\LWR@strresult}{>}{\detokenize{>}}{\LWR@multicolother{#1}}{}%
7677 \IfStrEq{\LWR@strresult}{<}{\detokenize{<}}{\LWR@multicolother{#1}}{}%

```

If an invalid column type:

```
7678 \ifbool{LWR@validtablecol}{}{\LWR@multicoltext}%
```

Tracing:

```

7679 \LWR@traceinfo{lwr@printmccoldata done}%
7680 }
```

\parsemulticolumnalignment {⟨1: colspec⟩} {⟨2: printresults⟩}

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.

```

7681 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
7682 \setcounter{LWR@tablemulticolspos}{1}%
7683 \StrLen{#1}[\LWR@strresult]%
7684 \setcounter{LWR@tablemulticolwidth}{\LWR@strresult}%

```

Scan across the tokens in the column spec:

```

7685 \whileboolexpr{%
7686     not test {%
7687         \ifnumcomp{\value{LWR@tablemulticolspos}}{>}{%
7688             {\value{LWR@tablemulticolswidth}}{%
7689         }{%
7690     }{%
7691 }{%

```

Execute the assigned print function for each token in the column spec:

```
7692 #2{#1}{%
```

Move to the next token in the column spec:

```

7693 \addtocounter{LWR@tablemulticolspos}{1}{%
7694 }{%
7695 }
```

71.21.2 Multicolumn factored code

```
\LWR@addmulticolverrulecolor
```

```
7696 \newcommand*{\LWR@addmulticolverrulecolor}{%
```

No vertical rules if finishing the tabular with a row of empty cells:

```
7697 \ifbool{\LWR@tabularmutemods}{}{%
```

Left side:

```

7698 \ifnumcomp{\value{LWR@mcolvertbarsl}}{=}{1}{%
7699     \LWR@tdaddstyle{%
7700         border-left: 1px solid \LWR@verruleHTMLcolor{%
7701     }{%
7702 \ifnumcomp{\value{LWR@mcolvertbarsl}}{>}{1}{%
7703     \LWR@tdaddstyle{%
7704         border-left: 4px double \LWR@verruleHTMLcolor{%
7705     }{%
7706 \ifnumcomp{\value{LWR@mcolvertbarsldash}}{=}{1}{%
7707     \LWR@tdaddstyle{%
7708         border-left: 1px dashed \LWR@verruleHTMLcolor{%
7709     }{%
7710 \ifnumcomp{\value{LWR@mcolvertbarsldash}}{>}{1}{%
7711     \LWR@tdaddstyle{%
7712         border-left: 2px dashed \LWR@verruleHTMLcolor{%
7713     }{%

```

Right side:

```

7714      \ifnumcomp{\value{LWR@mcolvertbarsr}}{=}{1}{%
7715          \LWR@tdaddstyle%
7716          border-right: 1px solid \LWR@vertruleHTMLcolor%
7717      }{%
7718      \ifnumcomp{\value{LWR@mcolvertbarsr}}{>}{1}{%
7719          \LWR@tdaddstyle%
7720          border-right: 4px double \LWR@vertruleHTMLcolor%
7721      }{%
7722      \ifnumcomp{\value{LWR@mcolvertbarsrdash}}{=}{1}{%
7723          \LWR@tdaddstyle%
7724          border-right: 1px dashed \LWR@vertruleHTMLcolor%
7725      }{%
7726      \ifnumcomp{\value{LWR@mcolvertbarsrdash}}{>}{1}{%
7727          \LWR@tdaddstyle%
7728          border-right: 2px dashed \LWR@vertruleHTMLcolor%
7729      }{%
7730 }%
7731 }
7732 \newcommand{\LWR@multicoltext}{}}

```

To find multicolumn right trim:

```

7733 \newcounter{LWR@lastmulticolumn}

\LWR@domulticolumn  [{<1: vpos>} [{<2: #rows>} {<3: numLaTeXcols>} {<4: numHTMLcols>} {<5: colspec>} {<6: text>}]

7734 \NewDocumentCommand{\LWR@domulticolumn}{o o m m +m}{%
7735 \LWR@traceinfo{LWR@domulticolumn -#1- -#2- -#4- -#5-}%

```

Remember the text to be inserted, and remember that a valid column type was found:

```

7736 \renewcommand{\LWR@multicoltext}{%
7737 #6%
7738 \global\booltrue{LWR@validtablecol}%
7739 }%

```

Compute the rightmost column to be included. This is used to create the right trim.

```

7740 \setcounter{LWR@lastmulticolumn}{\value{LWR@tableLaTeXcolindex}}%
7741 \addtocounter{LWR@lastmulticolumn}{#3}%
7742 \addtocounter{LWR@lastmulticolumn}{-1}%

```

Row processing:

```
7743 \LWR@maybenewtablerow%
```

Begin the opening table data tag:

```
7744 \LWR@htmltag{td colspan="#4" %
7745 \IfValueT{#2}{ % rows?
7746 rowspan="#2" %

7747 \IfValueT{#1}{% vpos?
7748 \ifstrequal{#1}{b}{\style=\\"LWR@print@mbox{vertical-align:bottom}" }{}%
7749 \ifstrequal{#1}{t}{\style=\\"LWR@print@mbox{vertical-align:top}" }{}%
7750 }% vpos?
7751 }% rows?

7752 class="td%"
```

Print the column type and vertical bars:

```
7753 \setcounter{LWR@mcolvertbarsl}{0}%
7754 \setcounter{LWR@mcolvertbarsr}{0}%
7755 \setcounter{LWR@mcolvertbarsldash}{0}%
7756 \setcounter{LWR@mcolvertbarsrdash}{0}%
7757 \booltrue{LWR@mcolvertbaronleft}%
7758 \LWR@parsemulticolumnalignment{#5}{\LWR@printmccoltype}%

```

If this column has a cmidrule, add “rule” to the end of the HTML class tag.

If this position had a “Y” then add “rule” for a horizontal rule:

```
7759 \LWR@subaddcmidruletrim%
7760 {\LWR@getexparray{\LWR@trimrlrules}{\arabic{\LWR@tableLaTeXcolindex}}}%
7761 {\LWR@getexparray{\LWR@trimrrrules}{\arabic{\LWR@lastmulticolumn}}}%
```

Also add vertical bar class.

```
7762 \ifnumcomp{\value{LWR@mcolvertbarsl}}{=}{1}{ tvertbar1}{%
7763 \ifnumcomp{\value{LWR@mcolvertbarsl}}{>}{1}{ tvertbar1double}{%
7764 \ifnumcomp{\value{LWR@mcolvertbarsr}}{=}{1}{ tvertbarr}{%
7765 \ifnumcomp{\value{LWR@mcolvertbarsr}}{>}{1}{ tvertbarrdouble}{%
7766 \ifnumcomp{\value{LWR@mcolvertbarsldash}}{=}{1}{ tvertbar1dash}{%
7767 \ifnumcomp{\value{LWR@mcolvertbarsldash}}{>}{1}{ tvertbar1doubledash}{%
7768 \ifnumcomp{\value{LWR@mcolvertbarsrdash}}{=}{1}{ tvertbarrdash}{%
7769 \ifnumcomp{\value{LWR@mcolvertbarsrdash}}{>}{1}{ tvertbarrdoubledash}{%
```

Close the class tag’s opening quote: " NOT A TYPO

```
7770 %
```

```

7771 \LWR@tdstartstyles%
7772 \LWR@addtabulararrowcolor%
7773 \LWR@addcmidrulewidth%
7774 \LWR@addcdashline%
7775 \LWR@addtabularhrulecolor%
7776 \LWR@addmulticolvertrulecolor%
7777 \LWR@addformatwpalignment{#5}%
7778 \LWR@tdendstyles%
7779 }% end of the opening table data tag
7780 \global\boolfalse{LWR@intabularmetadata}%
7781 \LWR@parsemulticolumnalignment{#5}{\LWR@printmccoldata}%
7782 }

```

71.21.3 Multicolumn

```

\lWR@htmlemulticolumn {<numcols>} {<alignment>} {<text>}
7783 \NewDocumentCommand{\lWR@htmlemulticolumn}{m m +m}%
7784 {%

```

Figure out how many extra HTML columns to add for @ and ! columns:

```
7785 \lWR@tabularhtmlcolumns{\arabic{LWR@tableLaTeXcolindex}}{#1}
```

Create the multicolumn tag:

```
7786 \lWR@domulticolumn{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#3}%
```

Move to the next L^AT_EX column:

```
7787 \addtocounter{LWR@tableLaTeXcolindex}{#1}%
7788 \addtocounter{LWR@tableLaTeXcolindex}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
7789 \global\booltrue{LWR@skipatbang}%
7790 }
```

71.21.4 Longtable captions

longtable captions use \multicolumn.

Bool LWR@starredlongtable Per the caption pacakge, step the counter if longtable*.

```
7791 \newbool{LWR@starredlongtable}
7792 \boolfalse{LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
7793 \providecommand*{\LTcaptive}{table}
```

\LWR@longtabledatacaptiontag * [<toc entry>] {<caption>}

```
7794 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
7795 {%
```

Remember the latest name for \nameref:

```
7796 \IfValueTF{#2}{% optional given?
7797   \ifblank{#2}{% optional empty?
7798     {\LWR@setlatestname{#3}}% empty
7799     {\LWR@setlatestname{#2}}% given and non-empty
7800   }% optional given
7801 {\LWR@setlatestname{#3}}% no optional
```

Create a multicolumn across all the columns:

Figure out how many extra HTML columns to add for @ and ! columns found between the first and the last column:

```
7802 \LWR@tabularhtmlcolumns{1}{\arabic{LWR@tabletotalLaTeXcols}}
```

Create the multicolumn tag:

```
7803 \LWR@domulticolumn{\arabic{LWR@tabletotalLaTeXcols}}%
7804   {\arabic{LWR@tabhtmlcoltotal}}%
7805   {P}%
7806 {%
7807 \LWR@domulticolumn
7807 \IfBooleanTF{#1}{ star? }
```

Star version, show a caption but do not make a LOT entry:

```
7808 {%
7809   yes star
7810   \LWR@figcaption%
7810   \LWR@isolate{#3}%
7811   \endLWR@figcaption%
7812 }%
7813 {%
7813 No star:
```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```

7814      \ifbool{LWR@starredlongtable}%
7815      {%
7816          \ifblank{#2}{% TOC entry
7817              {}%
7818          {}%
7819              \refstepcounter{\LTcaptype}%
7820              \protected@edef{@currentlabel}{%
7821                  \nameuse{p@\LTcaptype}\nameuse{the\LTcaptype}%
7822              }%
7823          }%
7824      }{}%

```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```

7825      \LWR@figcaption%
7826      \LWR@isolate{\nameuse{fnum@\LTcaptype}}%
7827      \CaptionSeparator%
7828      \LWR@isolate{#3}%
7829      \endLWR@figcaption%

```

See if an optional caption was given:

```
7830      \ifblank{#2}{% TOC entry empty
```

if the optional caption was given, but empty, do not form a TOC entry

```
7831      {}%
```

If the optional caption was given, but might only be []:

```

7832      {%
7833          \IfNoValueTF{#2}{% No TOC entry?}
```

The optional caption is []:

```

7834      {%
7835          \addcontentsline{%
7836              \nameuse{ext@\LTcaptype}}{%
7837                  \LTcaptype}%
7838          {}%
7839              \protect\numberline{%
7840                  \LWR@isolate{\nameuse{p@\LTcaptype}}\nameuse{the\LTcaptype}}%
7841                  \ignorespaces \LWR@isolate{#3}\protect\relax}%
7842          {}%
7843      }% end of No TOC entry
```

The optional caption has text enclosed:

```

7844      {%
7845          yes TOC entry
7846          \addcontentsline%
7847          {\@nameuse{ext@\LTcaptype}}%
7848          {\LTcaptype}%
7849          {%
7850              \protect\newline%
7851              {\LWR@isolate{\@nameuse{p@\LTcaptype}}\@nameuse{the\LTcaptype}}%
7852              {\ignorespaces \LWR@isolate{\#2}\protect\relax}%
7853          }%
7854      }% end of yes TOC entry
7855  }% end of TOC entry not empty
7855 }% end of no star

```

Skip any trailing @ or ! columns for this cell:

```

7856 \global\booltrue{\LWR@skipatbang}%
7857 }% end of \LWR@domulticolumn
7858
7859 \addtocounter{\LWR@tableLaTeXcolindex}{\arabic{\LWR@tabletotalLaTeXcols}}
7860 \addtocounter{\LWR@tableLaTeXcolindex}{-1}
7861
7862 }

```

71.21.5 Counting HTML tabular columns

The L^AT_EX specification for a table includes a number of columns separated by the & character. These columns differ in content from line to line. Additional virtual columns may be specified by the special @ and ! columns. These columns are identical from line to line, but may be skipped during a multicolumn cell.

For HTML output, @ and ! columns are placed into their own tabular columns. Thus, a L^AT_EX \multicolumn command may span several additional @ and ! columns in HTML output. These additional columns must be added to the total number of columns spanned by an HTML multi-column data cell.

```

7863 \newcounter{\LWR@tabhtmlcolindex}
7864 \newcounter{\LWR@tabhtmlcolend}
7865 \newcounter{\LWR@tabhtmlcoltotal}

```

\LWR@subtabularhtmlcolumns {<*index*>}

Factored from \LWR@tabularhtmlcolumns, which follows.

```

7866 \newcommand*{\LWR@subtabularhtmlcolumns}[1]{%

```

Temporarily define a macro equal to the @ specification for this column:

```
7867 \edef\LWR@atbangspec{\LWR@getexparray{\LWR@colatspec}{#1}}%
```

If the @ specification is not empty, add to the count:

```
7868 \ifdefempty{\LWR@atbangspec}%
7869   {}%
7870   {\addtocounter{\LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the ! columns:

```
7871 \edef\LWR@atbangspec{\LWR@getexparray{\LWR@colbangspec}{#1}}%
7872 \ifdefempty{\LWR@atbangspec}%
7873   {}%
7874   {\addtocounter{\LWR@tabhtmlcoltotal}{1}}%
7875 }
```

\LWR@tabularhtmlcolumns {*starting L^AT_EX column*} {*number L^AT_EX columns*}

Compute the total number of HTML columns being spanned, considering the starting L^AT_EX table column and the number of L^AT_EX tabular columns being spanned. Any @ and ! columns within this span are included in the total count. The resulting number of HTML columns is returned in the counter LWR@tabhtmlcoltotal.

```
7876 \newcommand*\LWR@tabularhtmlcolumns[2]{%
```

Count the starting index, compute ending index, and begin with the count being the L^AT_EX span, to which additional @ and ! columns may be added:

```
7877 \setcounter{\LWR@tabhtmlcolindex}{#1}%
7878 \setcounter{\LWR@tabhtmlcoltotal}{#2}%
7879 \setcounter{\LWR@tabhtmlcolend}{#1}%
7880 \addtocounter{\LWR@tabhtmlcolend}{#2}%
```

If at the left edge, add the at/bang columns for the left edge:

```
7881 \ifnumcomp{\value{\LWR@tabhtmlcolindex}}{=}{1}{%
7882   \LWR@subtabularhtmlcolumns{leftedge}%
7883 }{}}
```

Walk across the L^AT_EX columns looking for @ and ! columns:

```
7884 \whileboolexpr{%
7885   test {%
7886     \ifnumcomp{\value{\LWR@tabhtmlcolindex}}{<}{\value{\LWR@tabhtmlcolend}}{%
7887   }}{}}
```

```

7888 }%
7889 {%
7890     \LWR@subtabularhtmlcolumns{\arabic{LWR@tabhtmlcolindex}}%
7891     \addtocounter{LWR@tabhtmlcolindex}{1}%
7892 }% whiledo
7893 }

7894 \end{warpHTML}

```

71.22 Multirow if not loaded

A default defintion in case multirow is not loaded. This is used during table parsing.

```

7895 \begin{warpHTML}
7896 \newcommand{\multirow}[2][c]{}
7897 \end{warpHTML}

```

71.23 Multicolumnrow

A print-mode version is defined here, and is also used during HTML output while inside a `lateximage`.

See section [299](#) for the HTML versions.

for HTML & PRINT: 7898 `\begin{warpall}`

```
\multicolumnrow {\langle 1:cols \rangle} {\langle 2:halign \rangle} [\langle 3:vpos \rangle] {\langle 4:numrows \rangle} [\langle 5:bigstruts \rangle] {\langle 6:width \rangle} [\langle 7:fixup \rangle]
{\langle 8:text \rangle}
```

For discussion of the use of `\DeclareExpandableDocumentCommand`, see:
<https://tex.stackexchange.com/questions/168434/problem-with-abbreviation-of-multirow-and-multicolumn-latex>

`\AtBeginDocument` to adjust after the user may have loaded multirow, which requires several tests to determine which version is loaded and thus which options are available.

7899 `\AtBeginDocument{`

`\@ifundefined{@xmultirow}` determines if multirow was never loaded.

Null action if not loaded:

7900 `\@ifundefined{@xmultirow}`

```

7901 {
7902 \DeclareExpandableDocumentCommand{\LWR@print@multicolumnrow}{%
7903     {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
7904     {}%
7905 }% no version of multirow was loaded
7906 {% \xmultirow defined, so some version of multirow was loaded

```

\@ifpackageloaded{multirow} determines if v2.0 or later of multirow was used, which included the \ProvidesPackage macro.

The print version:

```

7907 \@ifpackageloaded{multirow}{% v2.0 or newer
7908 \@ifpackagelater{multirow}{2016/09/01}{2016/09/27 for v2.0
7909 {% v2.0+:
7910 \DeclareExpandableDocumentCommand{\LWR@print@multicolumnrow}{%
7911     {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
7912     {\multicolumn{#1}{#2}{\@xmultirow{#3}{#4}{#5}{#6}{#7}{#8}}}%
7913 }
7914 {% loaded but older, probably not executed:
7915 \DeclareExpandableDocumentCommand{\LWR@print@multicolumnrow}{%
7916     {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
7917     {\multicolumn{#1}{#2}{\@xmultirow{#4}{#5}{#6}{#7}{#8}}}%
7918 }
7919 }% packageloaded{multirow}

```

If not \@ifpackageloaded{multirow} but \xmultirow is defined, then this must be v1.6 or earlier, which did not \ProvidesPackage{multirow}, and did not have the vposn option.

```

7920 {% v1.6 or older did not \ProvidePackage
7921 \DeclareExpandableDocumentCommand{\LWR@print@multicolumnrow}{%
7922     {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
7923     {\multicolumn{#1}{#2}{\@xmultirow{#4}{#5}{#6}{#7}{#8}}}%
7924 }
7925
7926 }% \@ifundefined{\xmultirow}
7927
7928 \providecommand*\multicolumn{\LWR@print@multicolumnrow}
7929
7930 }% AtBeginDocument
7931 \end{warpall}

```

71.24 Utility macros inside a table

for HTML output: 7932 \begin{warpHTML}

Used to prevent opening a tabular data cell if the following token is one which does not create tabular data:

```
7933 \newcommand*{\LWR@donothing}{}
```

In case array is not loaded:

```
7934 \let\firsthline\relax  
7935 \let\lasthline\relax  
7936 \newcommand*{\firsthline}{}  
7937 \newcommand*{\lasthline}{}{}
```

In case bigdelim is not loaded:

```
7938 \newcommand*{\ldelim}{}  
7939 \newcommand*{\rdelim}{}  
  
7940 \end{warpHTML}
```

71.25 Special-case tabular markers

for HTML & PRINT: 7941 \begin{warpall}

\TabularMacro Place this just before inserting a custom macro in a table data cell. Doing so tells lwarp not to automatically start a new HTML table data cell yet. See section 9.9.

```
7942 \newcommand*{\TabularMacro}{}  
  
7943 \end{warpall}
```

\ResumeTabular Used to resume tabular entries after resuming an environment.

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

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

for HTML output: 7944 \begin{warpHTML}

```
7945 \newcommand*\{\ResumeTabular\}{%
7946 \global\boolfalse{LWR@exittingtabular}%
7947 \global\boolfalse{LWR@tabularmutemods}%
7948 \LWR@getmynexttoken%
7949 }
```

```
7950 \end{warpHTML}
```

for PRINT output: 7951 \begin{warpprint}

```
7952 \newcommand*\{\ResumeTabular\}{}
7953 \end{warpprint}
```

71.26 Checking for a new table cell

for HTML output: 7954 \begin{warpHTML}

Bool LWR@exittingtabular When \end is found, turns off the next opening data tag.

```
7955 \newbool{LWR@exittingtabular}
```

Bool LWR@tabularmutemods Mutes HTML output for @, !, < and >.

This is used while printing the final row to generate \bottomrules.

```
7956 \newbool{LWR@tabularmutemods}
```

- \LWR@tabledatacolumntag Open a new HTML table cell unless the next token is for a macro which does not create data, such as \hline, \toprule, etc:

```

7957 \newcommand*\LWR@tabledatacolumntag{%
7958 {%
7959 \LWR@traceinfo{\LWR@tabledatacolumntag}{%
    \show\LWR@mynexttoken to see what tokens to look for
}

```

If not any of the below, start a new table cell:

```
7960 \global\let\LWR@mynextaction\LWR@tabledatasinglecolumntag%
```

If exiting the tabular:

```

7961 \ifdefequal{\LWR@mynexttoken}{\end}{%
7962     {\global\booltrue{\LWR@exittingtabular}}{}%
}

```

`longtable` can have a caption in a cell

```

7963 \ifdefequal{\LWR@mynexttoken}{\caption}{%
7964     {\global\let\LWR@mynextaction\LWR@donothing}{}%
}

```

Look for other things which would not start a table cell:

```

7965 \ifdefequal{\LWR@mynexttoken}{\multicolumn}{%
7966     {\global\let\LWR@mynextaction\LWR@donothing}{}%
7967 \ifdefequal{\LWR@mynexttoken}{\multirow}{%
7968     {\global\let\LWR@mynextaction\LWR@donothing}{}%
7969 \ifdefequal{\LWR@mynexttoken}{\multicolumnrow}{%
7970     {\global\let\LWR@mynextaction\LWR@donothing}{}%
7971 \ifdefequal{\LWR@mynexttoken}{\noalign}{%
7972     {\global\let\LWR@mynextaction\LWR@donothing}{}%
}

```

If an `\mrowcell`, this is a cell to be skipped over:

```

7973 \ifdefequal{\LWR@mynexttoken}{\mrowcell}{%
7974     {\global\let\LWR@mynextaction\LWR@donothing}{}%
}

```

If an `\mcolrowcell`, this is a cell to be skipped over:

```

7975 \ifdefequal{\LWR@mynexttoken}{\mcolrowcell}{%
7976     {\global\let\LWR@mynextaction\LWR@donothing}{}%
7977 \ifdefequal{\LWR@mynexttoken}{\TabularMacro}{%
7978     {\global\let\LWR@mynextaction\LWR@donothing}{}%
}

```

```
7979 \ifdefeql{\LWR@mynexttoken}{\hline}%
7980     {\global\let\LWR@mynextaction\LWR@donothing}{}%
7981 \ifdefeql{\LWR@mynexttoken}{\firsthline}%
7982     {\global\let\LWR@mynextaction\LWR@donothing}{}%
7983 \ifdefeql{\LWR@mynexttoken}{\lasthline}%
7984     {\global\let\LWR@mynextaction\LWR@donothing}{}%
7985 \ifdefeql{\LWR@mynexttoken}{\toprule}%
7986     {\global\let\LWR@mynextaction\LWR@donothing}{}%
7987 \ifdefeql{\LWR@mynexttoken}{\midrule}%
7988     {\global\let\LWR@mynextaction\LWR@donothing}{}%
7989 \ifdefeql{\LWR@mynexttoken}{\cmidrule}%
7990     {\global\let\LWR@mynextaction\LWR@donothing}{}%
7991 \ifdefeql{\LWR@mynexttoken}{\morecmidrules}%
7992     {\global\let\LWR@mynextaction\LWR@donothing}{}%
7993 \ifdefeql{\LWR@mynexttoken}{\specialrule}%
7994     {\global\let\LWR@mynextaction\LWR@donothing}{}%
7995 \ifdefeql{\LWR@mynexttoken}{\cline}%
7996     {\global\let\LWR@mynextaction\LWR@donothing}{}%
7997 \ifdefeql{\LWR@mynexttoken}{\bottomrule}%
7998     {\global\let\LWR@mynextaction\LWR@donothing}{}%
7999 \ifdefeql{\LWR@mynexttoken}{\rowcolor}%
8000     {\global\let\LWR@mynextaction\LWR@donothing}{}%
8001 \ifdefeql{\LWR@mynexttoken}{\arrayrulecolor}%
8002     {\global\let\LWR@mynextaction\LWR@donothing}{}%
8003 \ifdefeql{\LWR@mynexttoken}{\doublerulesepcolor}%
8004     {\global\let\LWR@mynextaction\LWR@donothing}{}%
8005 \ifdefeql{\LWR@mynexttoken}{\warpprintonly}%
8006     {\global\let\LWR@mynextaction\LWR@donothing}{}%
8007 \ifdefeql{\LWR@mynexttoken}{\warpHTMLonly}%
8008     {\global\let\LWR@mynextaction\LWR@donothing}{}%
```

```

8009 \ifdefequal{\LWR@mynexttoken}{\ldelim}%
8010   {\global\let\LWR@mynextaction\LWR@donothing}{}%
8011 \ifdefequal{\LWR@mynexttoken}{\rdelim}%
8012   {\global\let\LWR@mynextaction\LWR@donothing}{}%

```

For arydshln:

```

8013 \ifdefequal{\LWR@mynexttoken}{\hdashline}%
8014   {\global\let\LWR@mynextaction\LWR@donothing}{}%
8015 \ifdefequal{\LWR@mynexttoken}{\cdashline}%
8016   {\global\let\LWR@mynextaction\LWR@donothing}{}%
8017 \ifdefequal{\LWR@mynexttoken}{\firstdashline}%
8018   {\global\let\LWR@mynextaction\LWR@donothing}{}%
8019 \ifdefequal{\LWR@mynexttoken}{\lastdashline}%
8020   {\global\let\LWR@mynextaction\LWR@donothing}{}%

```

Ignore an empty line between rows:

```

8021 \ifdefequal{\LWR@mynexttoken}{\par}%
8022   {\global\let\LWR@mynextaction\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:

```

8023 \LWR@traceinfo{LWR@tabledatacolumntag: about to do mynext}%
8024 \LWR@mynextaction%
8025 \LWR@traceinfo{LWR@tabledatacolumntag: done}%
8026 }

8027 \end{warpHTML}

```

71.27 \mrowcell

for HTML & PRINT: 8028 \begin{warpall}

- \mrowcell The user must insert \mrowcell into any \multirow cells which must be skipped.
- ⚠ multirow cells** This command has no action during print output.

```
8029 \newcommand*{\mrowcell}{}{}
```

```
8030 \end{warpall}
```

71.28 \mcolrowcell

for HTML & PRINT: 8031 \begin{warpall}

- ⚠ **\mcolrowcell** The user must insert **\mcolrowcell** into any **\multicolumnrow** cells which must be skipped. This command has no action during print output.

```
8032 \newcommand*{\mcolrowcell}{}{}
```

```
8033 \end{warpall}
```

71.29 HTML tabular environment

for HTML output: 8034 \begin{warpHTML}

These are default definitions in case booktabs is not loaded, and are not expected to be used, but must exist as placeholders. They are pre-deleted in case memoir has already loaded booktabs.

```
8035 \LetLtxMacro\toprule\relax
8036 \LetLtxMacro\midrule\relax
8037 \LetLtxMacro\cmidrule\cline
8038 \LetLtxMacro\bottomrule\relax
8039 \LetLtxMacro\addlinespace\relax
8040 \LetLtxMacro\morecmidrules\relax
8041 \LetLtxMacro\specialrule\relax
8042
8043 \newcommand*{\toprule}[1][]{\hline}
8044 \newcommand*{\midrule}[1][]{\hline}
8045 \LetLtxMacro\cmidrule\cline
8046 \newcommand*{\bottomrule}[1][]{\hline}
8047 \newcommand*{\addlinespace}[1][]{}
8048 \newcommand*{\morecmidrules}{}{}
8049 \newcommand*{\specialrule}[3]{\hline}
```

\noalign {<text>} Redefined for use inside tabular.

```

8050 \LetLtxMacro{\LWR@orignoalign}{\noalign}
8051
8052 \newcommand{\LWR@tabularnoalign}[1]{%
8053 \begingroup%
8054 \global\advance\rownum\m@ne%
8055 \renewcommand*{\LWR@xcolorrowHTMLcolor}{}%
8056 \multicolumn{\value{LWR@tabletotalLaTeXcols}}{1}{#1} \\
8057 \endgroup%
8058 % \rowc@lors%
8059 \LWR@getmynexttoken%
8060 }

```

`\LWR@HTMLhline` The definition of `\hline` depends on whether `tbls` has been loaded. If so, optional space below the line may be specified, but will be ignored.

```

8061 \AtBeginDocument{
8062 @ifpackageloaded{l warp-tables}
8063 {
8064 \newcommand*{\LWR@HTMLhline}[1][]{%
8065     \ifbool{FormatWP}{%
8066         {\LWR@docmidrule{1-\arabic{LWR@tabletotalLaTeXcols}}}%
8067         {\addtocounter{LWR@hlines}{1}}%
8068         \LWR@getmynexttoken}%
8069 }
8070 {
8071 \newcommand*{\LWR@HTMLhline}[]{%
8072     \ifbool{FormatWP}{%
8073         {\LWR@docmidrule{1-\arabic{LWR@tabletotalLaTeXcols}}}%
8074         {\addtocounter{LWR@hlines}{1}}%
8075         \LWR@getmynexttoken}%
8076 }
8077 }% AtBeginDocument

```

`\LWR@HTMLcline` {*<columns>*}

```

8078 \NewDocumentCommand{\LWR@HTMLcline}{m}{%
8079 {\LWR@docmidrule{#1}\LWR@getmynexttoken}%

```

`\LWR@nullifyNoAutoSpacing` For `babel-french`, turn off auto spacing at the start of the tabular, then nullify the autospacing commands inside the tabular, since they were not compatible with the tabular column parsing code, which uses `xstring`.

```

8080 \AtBeginDocument{
8081 @ifundefined{frenchbsetup}{%
8082 % no babel-french
8083 \newcommand*{\LWR@nullifyNoAutoSpacing}{}%
8084 }% no babel-french

```

```

8085 {%
8086   yes babel-french
8087   \newcommand*{\LWR@nullifyNoAutoSpacing}{%
8088     \NoAutoSpacing%
8089   }%
8090   \renewcommand*{\NoAutoSpacing}{}%
8091   \renewcommand*{\LWR@FBcancel}{}%
8092 }
8091 }%
8092 yes babel-french
8092 }%
8092 AtBeginDocument

```

Env tabular <direction> [(*vertposition*)] {<colspeсs>}

The <direction> is from plett for Japanese documents, and is ignored.

```

8093 \StartDefiningTabulars
8094
8095 \NewDocumentCommand{\LWR@HTML@@tabular}{d<> o m}
8096 {%
8097 \LWR@traceinfo{LWR@HTML@@tabular started}%
8098 \addtocounter{LWR@tabulardepth}{1}%

```

Not yet started a table row:

```
8099 \global\boolfalse{LWR@startedrow}%
```

Not yet doing any rules:

```

8100 \setcounter{LWR@hlines}{0}%
8101 \setcounter{LWR@hdashedlines}{0}%
8102 \global\boolfalse{LWR@doingtbrule}%
8103 \global\boolfalse{LWR@doingcmidrule}%

```

For babel-french, turn off auto spacing one time, then nullify the autospace commands since were not compatible with the tabular parsing code.

```
8104 \LWR@nullifyNoAutoSpacing%
```

Have not yet found the end of tabular command. Unmute the @ and ! columns.

```

8105 \global\boolfalse{LWR@exittingtabular}%
8106 \global\boolfalse{LWR@tabularmutemods}%

```

Create the table tag:

```

8107 \global\booltrue{LWR@intabularmetadata}%
8108 \LWR@traceinfo{LWR@tabular: About to LWR@forecenewpage.}%
8109 \LWR@forcenewpage
8110 \LWR@htmlblocktag{table}%

```

Parse the table columns:

```
8111 \LWR@parsetablecols{#3}%
```

Table col spec is: \LWR@tablecolspec which is a string of llccrr, etc.

Do not place the table inside a paragraph:

```
8112 \LWR@stoppars%
```

Track column #:

```
8113 \setcounter{LWR@tableLaTeXcolindex}{1}%
```

Have not yet added data in this column:

```
8114 \global\boolfalse{LWR@tabularcelladded}%
```

Start looking for midrules:

```
8115 \LWR@clearmidrules%
```

\\" becomes a macro to end the table row:

```
8116 \LetLtxMacro{\\"}{\LWR@tabularendofline}%
```

The following adjust for colortbl.

```
8117 \LetLtxMacro{\arrayrulecolor}{\arrayrulecolor\nexttoken}%
8118 \LetLtxMacro{\doublerulesepcolor}{\doublerulesepcolor\nexttoken}%
8119 \gdef{\LWR@columnHTMLcolor}{}%
8120 \gdef{\LWR@rowHTMLcolor}{}%
8121 \gdef{\LWR@cellHTMLcolor}{}%
8122 \rowcolors%
```

The vertical rules are set to the color active at the start of the tabular. \arrayrulecolor will then affect horizontal rules inside the tabular, but not the vertical rules.

```
8123 \ifdefvoid{\LWR@ruleHTMLcolor}%
8124   {\edef{\LWR@vertruleHTMLcolor}{black}}%
8125   {\edef{\LWR@vertruleHTMLcolor}{\LWR@origpound\LWR@ruleHTMLcolor}}%
```

Tracking the depth of cell color <div>s:

```
8126 \setcounter{LWR@cellcolordepth}{0}%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with `\LWR@getmynexttoken` to see if the next token might create a new data cell:

The optional parameter for `\hline` supports the `tabls` package.

```

8127 \LWR@traceinfo{\LWR@@HTML@tabular: redefining macros}%
8128 \LetLtxMacro\noalign{\LWR@tabularnoalign}%
8129 \LetLtxMacro\hline{\LWR@HTMLhline}%
8130 \LetLtxMacro\cline{\LWR@HTMLcline}%

8131 \DeclareDocumentCommand{\hdashline}{o}{%
8132     \ifbool{FormatWP}{%
8133         {\LWR@docdashline{1-\arabic{LWR@tabletotalLaTeXcols}}}{%
8134             {\addtocounter{LWR@hdashedlines}{1}}{%
8135                 \LWR@getmynexttoken}%
8136 }{%
8137 \DeclareDocumentCommand{\cdashline}{m}{%
8138     \LWR@docdashline{##1}\LWR@getmynexttoken}%
8139 }{%
8140 \DeclareDocumentCommand{\firstdashline}{o}{%
8141     \ifbool{FormatWP}{%
8142         {\LWR@docdashline{1-\arabic{LWR@tabletotalLaTeXcols}}}{%
8143             {\addtocounter{LWR@hdashedlines}{1}}{%
8144                 \LWR@getmynexttoken}%
8145 }{%
8146 \DeclareDocumentCommand{\lastdashline}{o}{%
8147     \ifbool{FormatWP}{%
8148         {\LWR@docdashline{1-\arabic{LWR@tabletotalLaTeXcols}}}{%
8149             {\addtocounter{LWR@hdashedlines}{1}}{%
8150                 \LWR@getmynexttoken}%
8151 }{%

```

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`.

```

8152 \renewcommand{\multicolumn}{\LWR@htmlmulticolumn}%
8153 \renewcommand*{\mrowcell}{%
8154     \LWR@maybenewtablerow}%
8155     \LWR@tabularleftedge}%
8156     \global\booltrue{\LWR@skippingmrowcell}%
8157 }{%
8158 \renewcommand*{\mcolrowcell}{%
8159     \LWR@maybenewtablerow}%
8160     \global\booltrue{\LWR@skippingmcolrowcell}%

```

```
8161 }%
8162 \LetLtxMacro\caption\LWR@longtabledatacaptiontag%
```

Reset for new processing:

```
8163 \global\boolfalse{LWR@tableparcell}%
8164 \global\boolfalse{LWR@skippingmrowcell}%
8165 \global\boolfalse{LWR@skippingmcolrowcell}%
8166 \global\boolfalse{LWR@skipatbang}%
8167 \global\boolfalse{LWR@emptyatbang}%
```

Set & for its special meaning inside the tabular:

```
8168 \StartDefiningTabulars%
8169 \protected\gdef&{\LWR@tabularampersand}%
```

Nest one level deeper of tabular paragraph handling:

```
8170 \addtocounter{LWR@tabularpardepth}{1}%

```

Look ahead for a possible table data cell:

```
8171 \LWR@traceinfo{LWR@HTML@tabular: about to LWR@getmynexttoken}%
8172 \LWR@getmynexttoken%
8173 }%
```

Ending the environment:

```
8174 \newcommand*\LWR@HTML@endtabular}%
8175 {%
8176 \LWR@traceinfo{LWR@HTML@endtabular}%

```

Unnest one level of tabular paragraph handling:

```
8177 \addtocounter{LWR@tabularpardepth}{-1}%
8178 \ifboolexpr{%
8179   test {%
8180     \ifnumcomp{\value{LWR@tableLaTeXcolindex}}{<}{\value{LWR@tabletotalLaTeXcols}}%
8181   } or %
8182   (%
8183     bool{LWR@intabularmetadata} and%
8184     not bool{LWR@tabularcelladded} and%
8185     test {%
8186       \ifnumcomp{\value{LWR@tableLaTeXcolindex}}{=}{\value{LWR@tabletotalLaTeXcols}}%
8187     }%
8188   )%
8189 }%
8190 {%
```

```

8191      \LWR@tabularfinishrow%
8192 }%
8193 {%
8194      \LWR@closetabledatacell%
8195 }%
8196 \LWR@htmlblocktag{/tr}%

```

xcolor row color support:

```

8197 \@rowc@lors%
8198 \LWR@htmlblocktag{/table}%
8199 \global\boolfalse{LWR@intabularmetadata}%

```

Unnest one level of tabular:

```
8200 \addtocounter{LWR@tabulardepth}{-1}%

```

Restore & to its usual meaning:

```

8201 \protected\gdef&{\LWR@origampmacro}%
8202 \StopDefiningTabulars%
8203 \LWR@traceinfo{LWR@HTML@endtabular finished}%
8204 }%
8205
8206 \csletcs{LWR@HTML@endtabular*}{LWR@HTML@endtabular}%
8207
8208 \StopDefiningTabulars

```

siunitx may redefine tabular, so set the following later:

```

8209 \AtBeginDocument{
8210     \LetLtxMacro{\LWR@origendtabular}{\endtabular}
8211     \csletcs{\LWR@origendtabular*}{\endtabular*}%
8212     \LWR@formatted{@tabular}%
8213     \LWR@formatted{endtabular}%
8214     \LWR@formatted{endtabular*}%
8215 }
8216 \end{warpHTML}

```

72 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 12 shows the data structures related to cross-referencing.

for HTML output: 8217 `\begin{warpHTML}`

72.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

8218 `\providecommand*\@currentlabelname{()}`

`\LWR@stripperiod {<text>} [<.>]`

Removes a trailing period.

8219 `\def\LWR@stripperiod#1.\ltx@empty#2\@nil{#1}%`

`\LWR@setlatestname {<object name>}`

Removes `\label`, strips any final period, and remembers the result.

8220 `\newcommand*\LWR@setlatestname[1]{%`

Remove `\label` and other commands from the name, the strip any final period. See `zref-titleref` and `gettitlestring`.

```
8221 \GetTitleStringExpand{#1}%
8222 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
8223 \edef\@currentlabelname{%
8224   \expandafter\LWR@stripperiod\@currentlabelname%
8225   \ltx@empty.\ltx@empty\@nil%
8226 }%
8227 }
```

Table 12: Cross-referencing data structures

Original LATEX:	(print and HTML)
\refstepcounter: Steps the counter and sets \currentlabel.	
\currentlabel: \p@<ctr>\the<ctr> Updated by \refstepcounter.	
\label: Writes to the .aux file: \newlabel{<label>}{{\currentlabel}{\thepage}}	
\newlabel: When the .aux file is read, sets \r@<label>.	
\r@<label>: Set to: {{\currentlabel}{\thepage}}	
\ref: Returns the first part of \r@<label>.	
\pageref: Returns the second part of \r@<label>.	
Added by l warp:	(HTML only)
\label: Adds HTML tags (section 72.3), plus \splabel data (section 72.2):	
zLWR@name: The section name for this label.	
zLWR@htmlfilenumber: The filenumber or name for this label.	
zLWR@teximagedepth: The teximagedepth for this label.	
zLWR@teximagenumber: The teximagenumber for this label.	
\nameref: Emulated from hyperref for l warp. See section 72.4.	
\ref and \nameref: Adds HTML tags. See section 72.4.	
Added by amsmath:	(print and HTML)
\label: Execution is delayed until the math environment is completed.	
\ltx@label: LATEX \label, (HTML: patched by l warp,) 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 58.4 for use with footnotes.	
\label: Writes to the .aux file: \newlabel{<label>@\cref}{{\cref@currentlabel}{\thepage}}	
\newlabel: (Unchanged.) When the .aux file is read, sets \r@<label>@\cref.	
\r@<label>@\cref: Set to: {{\cref@currentlabel}{\thepage}}	
Utility functions: See \cref@getlabel, \cref@gettype, \cref@getcounter, \cref@getprefix.	
Cross-referencing names: \crefname and \Crefname assign human-readable names for references to this counter type.	
Additionally patched by l warp:	(HTML only)
\cref, etc.: Modified for l warp. See section 86.	
\label inside math: See section 78.7.1.	
Footnotes: See \noteentry in section 58.4.	

72.2 zref setup

See:

<http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference>

Create a new property list called special:

```
8228 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
8229 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
8230 \zref@newprop{zLWR@htmlfilename}{%
8231 \ifbool{FileSectionNames}{\LWR@thisfilename}{\arabic{LWR@htmlfilename}}%
8232 }%
```

Additional properties for lateximages:

```
8233 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
8234 \zref@newprop{zLWR@lateximagenumber}{\arabic{LWR@lateximagenumber}}
```

zLWR@htmlfilename property holds the file number or name

Add a LWR@htmlfilename property, and lateximage properties to special:

```
8235 \zref@addprop{special}{zLWR@name}
8236 \zref@addprop{special}{zLWR@htmlfilename}
8237 \zref@addprop{special}{zLWR@lateximagedepth}
8238 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
8239 \newcommand*{\LWR@spref}[2]{%
8240 \zref@extractdefault{#1}{#2}{??}%
8241 }
```

\LWR@nameref {<label>} Returns the section name for this label:

```
8242 \newcommand*{\LWR@nameref}[1]{%
8243 \LWR@spref{#1}{zLWR@name}%
8244 }
```

\LWR@htmlfileref {*label*} Returns the file number or name for this label:

8245 \newcommand*\{\LWR@htmlfileref\}[1]{%

DO NOT USE \LWR@traceinfo HERE! Will be expanded.

8246 \LWR@sref{\#1}{z\LWR@htmlfilenumber}%

8247 }

\LWR@lateximagedepthref {*label*} Returns the `lateximagedepth` for this label:

8248 \newcommand*\{\LWR@lateximagedepthref\}[1]{%

8249 \LWR@sref{\#1}{z\LWR@lateximagedepth}%

8250 }

\LWR@lateximagenumberref {*label*} Returns the `lateximagenumber` for this label:

8251 \newcommand*\{\LWR@lateximagenumberref\}[1]{%

8252 \LWR@sref{\#1}{z\LWR@lateximagenumber}%

8253 }

\LWR@splabel {*label*} Sanitize the name and then creates the label:

8254 \newcommand*\{\LWR@splabel\}[1]{%

8255 \LWR@traceinfo{\LWR@splabel !#1!}%

8256 \LWR@setlatestname{\@currentlabelname}%

8257 \zref@labelbylist{\#1}{special}%

8258 }

72.3 Labels

\LWR@sublabel {*label*} Creates an HTML id tag.

\detokenize is used to allow underscores in the labels.

8259 \newcommand*\{\LWR@sublabel\}[1]{%

8260 \LWR@traceinfo{\LWR@sublabel !#1!}%

Create an HTML id tag unless are inside a `lateximage`, since it would appear in the image:

8261 \ifnumcomp{\value{\LWR@lateximagedepth}}{>}{0}%

8262 {}%

8263 {}% not `lateximage`

If not doing a `lateximage`, create an HTML ID tag: (To be factored...)

```

8264      \LWR@sanitize{#1}%
8265      \ifbool{\LWR@doingstartpars}{%
8266      {%
8267          \ifbool{\LWR@doingapar}{%
8268              {%
8269                  \LWR@htmltag{a \LWR@print@mbox{id="\LWR@sanitized"} }\LWR@htmltag{/a}}%
8270              }% par started
8271              {%
8272                  \LWR@stoppars%
8273                  \LWR@htmltag{a \LWR@print@mbox{id="\LWR@sanitized"} }\LWR@htmltag{/a}}%
8274                  \LWR@startpars%
8275              }% par not started
8276          }% pars allowed
8277          {%
8278              \LWR@htmltag{a \LWR@print@mbox{id="\LWR@sanitized"} }\LWR@htmltag{/a}}%
8279          }% pars not allowed
8280      }% not lateximage
8281 }

```

`\LWR@new@label` (*bookmark*) {*label*} [*type*]

label during HTML output when not in SVG math mode, removing extra spaces around the label, as done by regular L^AT_EX `\label`.

The is also used during a `lateximage`, including SVG math, since the special label handling is required, but `\LWR@sublabel` does not generate HTML tags inside a `lateximage`.

`cleveref` later encases this to add its own cross-referencing.

The optional *bookmark* is per the memoir class, and is ignored.

The optional *type* is per the ntheorem package, and is ignored.

```

8282 \NewDocumentCommand{\LWR@new@label}{d() m o}{%
8283 \LWR@traceinfo{\LWR@new@label: starting}%
8284 \LWR@traceinfo{\LWR@new@label: !#2!}%
8285 % \@bsphack%

```

Create a traditional L^AT_EX label, as modified by `cleveref`:

```
8286 \LWR@orig@label{#2}%
```

Create a special label which holds the section number, `LWR@htmlfilename`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```

8287 \LWR@traceinfo{LWR@new@label: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
8288 \LWR@traceinfo{LWR@new@label: LWR@thisfilename is !\LWR@thisfilename!}%
8289 \LWR@traceinfo{LWR@new@label: LWR@htmlfilename is \arabic{LWR@htmlfilename}}%
8290 \LWR@splabel{#2}%
8291 \LWR@sublabel{#2}%
8292 % \@esphack%
8293 \LWR@traceinfo{LWR@new@label: done}%
8294 }

```

72.4 References

\LWR@startref {*label*} (Common code for \ref and \nameref.)

Open an HTML tag reference to a filename, # character, and a label.

```

8295 \newcommand*\LWR@startref[1]
8296 {%
8297 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
8298 \LWR@sanitize{#1}%
8299 \LWR@traceinfo{LWR@startref A: !#1!}%

```

Create the filename part of the link:

```

8300 \LWR@htmlltag{a href="%"
8301 \LWR@traceinfo{LWR@startref B}%
8302 \LWR@print@mbox{\LWR@htmlrefsectionfilename{#1}}%
8303 \LWR@traceinfo{LWR@startref C}%
8304 \LWR@origpound%

```

Create the destination id:

See if LWR@lateximagedepth is unknown:

```

8305 \LWR@traceinfo{LWR@startref D: !#1!}%
8306 \ifthenelse{\equal{\LWR@lidref}{??}}%

```

“??” if LWR@lateximagedepth is unknown, so create a link with an unknown destination:

```

8307 {%
8308     \LWR@traceinfo{LWR@startref D0: ??}%
8309     ??%
8310 }%

```

If LWR@lateximagedepth is known. Use a lateximage if the depth is greater than zero, or a regular link otherwise:

```

8311 {%
8312   \LWR@traceinfo{LWR@startref D1: \LWR@lidref}%
8313   \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}{%
8314     {%
8315       \LWR@traceinfo{LWR@startref D2: \LWR@lidref}%
8316       lateximage\LWR@lateximagenumberref{#1}%
8317     }%
8318   {%
8319     \LWR@traceinfo{LWR@startref D3}%

```

\detokenize is used to allow underscores in the labels:

```

8320   \LWR@print@mbox{\LWR@sanitized}%
8321   }%
8322 }%
8323 \LWR@traceinfo{LWR@startref E}%

```

Closing quote:

```

8324 "}%
8325 \LWR@traceinfo{LWR@startref F}%
8326 }

```

\LWR@subnewref {<label>} {<label or sub@label>}

Factored for the subfig package. Uses the original label for the hyper-reference, but prints its own text, such as “1(b)”.

```

8327 \NewDocumentCommand{\LWR@subnewref}{m m}{%
8328 \LWR@traceinfo{LWR@subnewref #1 #2}%
8329 \LWR@startref{#1}%
8330 \LWR@print@ref{#2}%
8331 \LWR@htmntag{/a}%
8332 }

```

\ref * {<label>} \ref is redefined to \LWR@HTML@ref, except inside the text part of a \hyperref, where it is redefined to \LWR@ref@ignorestar.

\LWR@HTML@ref * {<label>} Create an internal document reference link, or without a link if starred per hyperref.

```

8333 \NewDocumentCommand{\LWR@HTML@ref}{s m}{%
8334 \LWR@traceinfo{LWR@HTML@ref !#2!}%
8335 \IfBooleanTF{#1}%

```

```
8336 {\LWR@print@ref{#2}}%
8337 {\LWR@subnewref{#2}{#2}}%
8338 }
8339
8340 \LWR@formatted{ref}
```

\LWR@ref@ignorestar * {*label*} For use inside \hyperref. Ignores the star, then uses the original \ref.

```
8341 \NewDocumentCommand{\LWR@ref@ignorestar}{s m}{%
8342     \LWR@print@ref{#2}}%
8343 }
```

\pagerefPageFor Text for page references.

```
8344 \newcommand*{\pagerefPageFor}{see }
```

\pageref * {*label*} Create an internal document reference, or just the unlinked number if starred, per hyperref.

```
8345 \NewDocumentCommand{\LWR@new@pageref}{s m}{%
8346 \IfBooleanTF{#1}{%
8347 {(\pagerefPageFor\LWR@print@ref{#2})}}{%
8348 {(\cpageref{#2})}}%
8349 }
```

\nameref {*label*}

```
8350 \newrobustcmd*{\nameref}[1]{%
8351 \LWR@traceinfo{nameref}%
8352 \LWR@startref{#1}%
8353 \LWR@traceinfo{nameref B}%
8354 \LWR@nameref{#1}%
8355 \LWR@traceinfo{nameref C}%
8356 \LWR@htmlltag{/a}%
8357 \LWR@traceinfo{nameref: done}%
8358 }
```

\Nameref {*label*} In print, adds the page number. In HTML, does not.

```
8359 \LetLtxMacro{\Nameref}{\nameref}
```

72.5 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.

```
8360 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
8361 % \EmulatesPackage{hyperref}[2015/08/01]%
% Disabled. Do not do this.
```

Emulates `hyperref`:

`\@currentHref` Added to support backref.

```
8362 \AtBeginDocument{
8363 \def\@currentHref{%
8364 autopage-\theLWR@currentautosec%
8365 }
8366 }
```

`\LWR@subhyperref` $\{\langle URL \rangle\}$

Starts a link for `\LWR@hrefb`. A group must have been opened first, with nullified catcodes. The text name is printed afterwards, after the group is closed and catcodes restored.

```
8367 \NewDocumentCommand{\LWR@subhyperref}{m}{%
8368 \LWR@traceinfo{\LWR@subhyperref !#1!}%
8369 \LWR@sanitize{#1}%
8370 \LWR@htmntag{%
8371     a href="\LWR@sanitized" %
8372     target="_blank"\LWR@orignewline%
8373 }%
8374 }
```

`\LWR@subhyperreftext` $\{\langle text \rangle\}$

Finishes the hyperref for `\LWR@hrefb`. Catcodes must have been restored already. To be used after `\LWR@subhyperref`, and after its group has been closed.

```
8375 \newcommand{\LWR@subhyperreftext}[1]{%
```

```

8376 #1%
8377 \LWR@htmltag{/a}%
8378 \LWR@ensuredoingapar%
8379 }

\LWR@subhyperrefclass {⟨URL⟩} {⟨text⟩} {⟨htmlclass⟩}

8380 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
8381 \LWR@htmltag{%
8382     a href="#"%
8383         \begingroup\@sanitize#1\endgroup%
8384     " %
8385     class="#3"\LWR@orignewline%
8386 }%
8387 #2%
8388 \LWR@htmltag{/a}%
8389 \LWR@ensuredoingapar%
8390 }

```

`\href [⟨options⟩] {⟨URL⟩}`

Create a link with accompanying text:

```

8391 \DeclareDocumentCommand{\LWR@hrefb}{O{} m}{%
8392 \LWR@ensuredoingapar%
8393 \LWR@subhyperref{#2}%
8394 \endgroup% restore catcodes
8395 \LWR@subhyperreftext%
8396 }%
8397
8398 \newrobustcmd*\href{%
8399 \begingroup%
8400 \catcode`\#=12%
8401 \catcode`\%=12%
8402 \catcode`\&=12%
8403 \catcode`\~=12%
8404 \catcode`\_=12%
8405 \LWR@hrefb%
8406 }

```

`\nolinkurl {⟨URL⟩}`

Print the name of the link without creating the link:

```

8407 \newcommand*\LWR@nolinkurlb[1]{%
8408 \LWR@ensuredoingapar%
8409 \def\LWR@templink{#1}%

```

```

8410 \@onelvel@sanitize\LWR@templink%
8411 \LWR@templink%
8412 \endgroup%
8413 }
8414
8415 \newrobustcmd*\{nolinkurl}{%
8416 \begingroup%
8417 \catcode`\#=12%
8418 \catcode`\%=12%
8419 \catcode`\&=12%
8420 \catcode`\~=12%
8421 \catcode`\_=12%
8422 \LWR@nolinkurlb%
8423 }

```

\url {⟨URL⟩}

Create a link whose text name is the address of the link.

The url package may redefine \url, so it is \let to \LWR@urlahere and also redefined by lwarp-url.

```

8424 \DeclareDocumentCommand{\LWR@urlb}{m}{%
8425 \LWR@ensuredoingapar%
8426 \def\LWR@templink[#1]{%
8427 \@onelvel@sanitize\LWR@templink%
8428 \href{\LWR@templink}{\LWR@templink}%
8429 \endgroup%
8430 }
8431
8432 \newrobustcmd*\{url}{%
8433 \begingroup%
8434 \catcode`\#=12%
8435 \catcode`\%=12%
8436 \catcode`\&=12%
8437 \catcode`\~=12%
8438 \catcode`\_=12%
8439 \LWR@urlb%
8440 }

```

\LWR@subinlineimage [⟨alttag⟩] {⟨class⟩} {⟨filename⟩} {⟨extension⟩} {⟨style⟩}

```

8441 \newcommand*\{LWR@subinlineimage}[5][]{%
8442 \ifblank[#1]{%
8443 {\LWR@htmltag{img src="#3.\#4" alt="#3" style="#5" class="#2"};}%
8444 {\LWR@htmltag{img src="#3.\#4" alt="#1" style="#5" class="#2"};}%
8445 }

```

```
8446 \end{warpHTML}
```

Table 13: 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>` “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.

73 Floats

Floats are supported, although partially through emulation.

Table 13 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”.

73.1 Float environment

for HTML output: 8447 \begin{warpHTML}

\LWR@floatbegin {*type*} [*placement*] Begins a \newfloat environment.

```
8448 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
8449 \ifbool{FormatWP}{\newline}{}%
8450 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
8451 \addtocounter{\LWR@thisautoid}{1}%
8452 \booltrue{\LWR@freezethisautoid}%
8453 \begingroup%
```

Settings while inside the environment:

```
8454 \LWR@print@raggedright%
```

Open an HTML figure tag. The figure is assigned a class equal to its type, and another class according to the float package style, if used. Note that \csuse returns an empty string if \LWR@floatstyle@<type> is not defined.

```
8455 \LWR@htmlltag{%
8456     figure id="\LWR@print@mbox{autoid-\arabic{\LWR@thisautoid}}" % space
8457     class="#1 \nameuse{\LWR@floatstyle@#1}"%
8458 }%
8459 \ifbool{FormatWP}{%
8460     \LWR@orignewline%
8461     \LWR@BlockClassWP{}{}{wp#1}%
8462 }{}%
```

Update the caption type:

```
8463 \renewcommand*{\@capttype}{#1}%
8464 \caption@settype{#1}%
```

Mark the float for a word processor conversion:

```
8465 \LWR@startpars%
8466 \ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%
8467
8468 === begin #1 ===
8469
8470 }{}%
```

Look for `\centering`, etc:

```
8471 \LWR@futureonospacelet\LWR@mynexttoken\LWR@floatalignment%
8472 }
```

For koma-script. The following does not work for tables.

```
8473 \AtBeginDocument{
8474 @ifpackageloaded{tocbasic}{
8475 \appto\figure@atbegin{%
8476 \LWR@futureonospacelet\LWR@mynexttoken\LWR@floatalignment%
8477 }
8478 }{}}
8479 }
```

`\@float` Support packages which create floats directly.
`\@dblfloat`

```
8480 \let@\float\LWR@floatbegin
8481 \let@\dblfloat\LWR@floatbegin
```

`\LWR@floatend` Ends a `\newfloat` environment.

```
8482 \newcommand*{\LWR@floatend}{%
```

If saw a `\centering`, finish the center environment:

```
8483 \LWR@endfloatalignment%
```

Mark the float end for a word processor conversion:

```
8484 \ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%
8485
8486 === end ===
8487
8488 }{}%
8489 \LWR@stopars%
```

Close an HTML figure tag:

```
8490 \ifbool{FormatWP}{\endLWR@BlockClassWP}{%
8491 \LWR@htmlelementend{figure}%
8492 \endgroup%
8493 \boolfalse{\LWR@freezethisautoid}%
8494 \LWR@startpars%
8495 \ifbool{FormatWP}{\newline}{%
8496 }
```

\end@float Support packages which create floats directly.
\end@dblfloat
8497 \let\end@float\LWR@floatend
8498 \let\end@dblfloat\LWR@floatend

73.2 Float tracking

Ctr LWR@thisautoid A sequential counter for all floats and theorems. This is used to identify the float or theorem then reference it from the List of Figures and List of Tables.

8499 \newcounter{LWR@thisautoid}

Ctr LWR@thisautoidWP A sequential counter for all word processor conversion <div>s. This is used to convince LIBREOFFICE to form a frame around this element.

8500 \newcounter{LWR@thisautoidWP}

Bool LWR@freezethisautoid Prevents multiple increments of \LWR@thisautoid inside a float.

8501 \newbool{LWR@freezethisautoid}
8502 \boolfalse{LWR@freezethisautoid}

\LWR@newautoidanchor Adds a new <autoid> anchor.

8503 \newcommand*{\LWR@newautoidanchor}{%
8504 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}{%
8505 {}%
8506 {
8507 \ifbool{LWR@freezethisautoid}{}{
8508 \addtocounter{LWR@thisautoid}{1}%
8509 \LWR@htmlltag{a id="\LWR@print@mbox{autoid-\arabic{LWR@thisautoid}}"}%
8510 \LWR@htmlltag{/a}%
8511 }%
8512 }
8513 }

\@capttype Remembers which float type is in use.

8514 \newcommand*{\@capttype}{}{}

\LWR@floatalignmentname Set to center, flushleft, or flushright if saw \centering, \raggedright, or \raggedleft.

8515 \newcommand*{\LWR@floatalignmentname}{}{}

\LWR@floatalignment If sees a \centering, \raggedleft, or \raggedright, creates a center, flushright, or flushleft environment.

```

8516 \newcommand*\LWR@floatalignment{%
8517 \ifdefstreq{\LWR@mynexttoken}{\centering}{%
8518     \center%
8519     \renewcommand*\LWR@floatalignmentname{center}%
8520 }{%
8521 \ifdefstreq{\LWR@mynexttoken}{\raggedright}{%
8522     \flushleft%
8523     \renewcommand*\LWR@floatalignmentname{flushleft}%
8524 }{%
8525 \ifdefstreq{\LWR@mynexttoken}{\raggedleft}{%
8526     \flushright%
8527     \renewcommand*\LWR@floatalignmentname{flushright}%
8528 }{%
8529 }

```

\LWR@endfloatalignment Closes an environment from \LWR@floatalignment.

```

8530 \newcommand*\LWR@endfloatalignment{%
8531 \ifdefvoid{\LWR@floatalignmentname}{}{\@nameuse{end\LWR@floatalignmentname}}%
8532 \renewcommand*\LWR@floatalignmentname{}%
8533 }

```

73.3 Caption inside a float environment

\CaptionSeparator How to separate the float number and the caption text.

```
8534 \AtBeginDocument{\providecommand*\CaptionSeparator{:\~}}
```

\@makecaption {\langle name and num\rangle} {\langle text\rangle}

Prints the float type and number, the caption separator, and the caption text.

```

8535 \AtBeginDocument{\renewcommand{\@makecaption}[2]{%
8536     \LWR@traceinfo{@makecaption}%
8537     \LWR@isolate{\#1}\CaptionSeparator\LWR@isolate{\#2}%
8538     \LWR@traceinfo{@makecaption: done}%
8539 }%
8540 }

```

73.4 Caption and lof linking and tracking

When a new HTML file is marked in the L^AT_EX PDF file, the L^AT_EX page number at that point is stored in `LWR@latestautopage`, (and the associated filename is remembered by the special L^AT_EX labels). This page number is used to generate an autopage HTML `<id>` in the HTML output at the start of the new HTML file. Meanwhile, there is a float counter used to generate an HTML autooid `<id>` at the start of the float itself in the HTML file. The autopage and autooid 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@nextautoid` Tracks autoid for floats. Tracks autopage for floats.
- Ctr `LWR@nextautopage` These are updated per float as the `.lof`, `.lot` file is read.

```
8541 \newcounter{LWR@nextautoid}
8542 \newcounter{LWR@nextautopage}
```

```
\LWRsetnextfloat {<autopage>} {<autooid>}
```

This is written to the `.lof`, `.lot` file just before each float's usual entry. The autopage and autooid are remembered for `\l@figure` to use when creating the HTML links.

```
8543 \newcommand*{\LWRsetnextfloat}[2]{%
8544 \setcounter{LWR@nextautopage}{#1}%
8545 \setcounter{LWR@nextautoid}{#2}%
8546 }
```

- Ctr `LWR@latestautopage` Updated each time a new HTML file is begun. `\LWRsetnextfloat` is written with this and the autooid by the modified `\addcontentsline` just before each float's entry.

```
8547 \newcounter{LWR@latestautopage}
8548 \setcounter{LWR@latestautopage}{1}
```

- Env `LWR@figcaption` An HTML `<figcaption>` is not allowed in places where L^AT_EX does allow a figure caption, such as inside a `longtable` where the tabular has already started, or inside a `center` environment. Therefore, a `<div>` of class `figurecaption` is used instead.

```
8549 \newenvironment*{LWR@figcaption}
8550 {
8551     \ifbool{FormatWP}{%
8552         \BlockClass{font-style:italic}{figurecaption}
8553         \LWR@print@vspace*\{\baselineskip\}
8554     }{
8555         \BlockClass{figurecaption}
8556     }%
```

```
8557      }
8558      {\endBlockClass}
```

```
\LWR@HTML@caption@begin {<type>}
```

Low-level patches to create HTML tags for captions.

```
8559 \newcommand*{\LWR@HTML@caption@begin}[1]
8560 {
8561 \LWR@traceinfo{\LWR@HTML@caption@begin}%
```

Keep par and minipage changes local:

```
8562 \begingroup%
```

The caption code was not allowing the closing par tag:

```
8563 \@setpar{\LWR@closeparagraph\@par}%
```

No need for a minipage or \parbox inside the caption:

```
8564 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{}{%
8565 \RenewDocumentCommand{\parbox}{O{t} O{} O{t} m +m}{{\#5}}%
```

Enclose the original caption code inside an HTML tag:

```
8566 \LWR@figcaption%
8567 \LWR@traceinfo{\LWR@HTML@caption@begin: about to \LWR@origcaption@begin}%
8568 \LWR@print@caption@begin{\#1}%
8569 \LWR@traceinfo{\LWR@HTML@caption@begin: done}%
8570 }
```

```
\LWR@HTML@caption@end Low-level patches to create HTML tags for captions.
```

```
8571 \newcommand*{\LWR@HTML@caption@end}
8572 {%
8573 \LWR@traceinfo{\LWR@HTML@caption@end}%
8574 \LWR@print@caption@end%
```

Closing tag:

```
8575 \endLWR@figcaption%
8576 \endgroup%
8577 % \leavevmode% avoid bad space factor (0) error
8578 \LWR@traceinfo{\LWR@HTML@caption@end: done}%
8579 }
```

\caption@begin Low-level patches to create HTML tags for captions. These are assigned \AtBeginDocument so that other packages which modify captions will have already been loaded before saving the print-mode version.

```
8580 \AtBeginDocument{  
8581 \LWR@formatted{caption@begin}  
8582 \LWR@formatted{caption@end}  
8583 }
```

\captionlistentry Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```
8584 \let\LWR@origcaptionlistentry\captionlistentry  
8585  
8586 \renewcommand*\captionlistentry{  
8587 \LWR@ensuredoingapar%  
8588 \LWR@origcaptionlistentry%  
8589 }  
8590  
8591 \def\LWR@LTcaptionlistentry{  
8592 \LWR@ensuredoingapar%  
8593 \LWR@htmntag{a id="\LWR@print@mbox{autoid-\arabic{\LWR@thisautoid}}"\}\LWR@htmntag{/a}%  
8594 \bgroup  
8595 \@ifstar{\egroup\LWR@LT@captionlistentry}{ gobble *  
8596 {\egroup\LWR@LT@captionlistentry}}%  
8597 \def\LWR@LT@captionlistentry#1{  
8598 \caption@listentry@\firstoftwo[\LTcaptype]{#1}}%
```

\addcontentsline Patched to write the autopage and autoid before each float's entry. No changes if writing .toc For a theorem, automatically defines \ext@<type> as needed, to mimic and reuse the float mechanism.

f

```
8599 \let\LWR@origaddcontentsline\addcontentsline  
8600  
8601 \renewcommand*\addcontentsline}[3]{  
8602 \ifstreq{\#1}{toc}{}{ not TOC  
  
8603 \ifnumcomp{\value{\LWR@lateimagedepth}}{>}{0}{  
8604 {}%  
8605 {\LWR@newautoidanchor}}%  
  
8606 \ifcsvvoid{\ext@#2}{\csdef{\ext@#2}{\#1}}{}%  
  
8607 \addtocontents{\@nameuse{\ext@#2}}{}}
```

```

8608     \protect\LWRsetnextfloat%
8609     {\arabic{LWR@latestautopage}}%
8610     {\arabic{LWR@thisautoid}}%
8611     }%
8612 }% not TOC
8613 \LWR@origaddcontentsline{\#1}{\#2}{\#3}%
8614 }

```

Pkg `capt-of` Either package provides `\captionof`, which is later patched at the beginning of the document.

`\captionof` Patched to handle paragraph tags.

```

8615 \AtBeginDocument{
8616 \let\LWR@origcaptionof\captionof
8617
8618 \renewcommand*{\captionof}{%
8619 \LWR@stoppars
8620 \LWR@origcaptionof%
8621 }
8622 }

8623 \end{warpHTML}

```

74 Table of Contents, LOF, LOT

This section controls the generation of the TOC, LOF, and LOT.

The `.toc`, `.lof`, and `.lot` files are named by the source code `\jobname`.

In HTML, the printed tables are placed inside a `<div>` of class `toc`, `lof`, or `lot`.

A “`sitetoc`” is provided which prints a subset of the TOC on the side of each page other than the homepage.

The regular L^AT_EX infrastructure is used for TOC, along with some patches to generate HTML output.

for HTML output: 8624 `\begin{warpHTML}`

74.1 Reading and printing the TOC

`\LWR@myshorttoc` `\{<toc/lof/lot/sitetoc>\}`

Reads in and prints the TOC/LOF/LOT at the current position. While doing so, makes the @ character into a normal letter to allow formatting commands in the section names.

Unlike in regular L^AT_EX, the file is not reset after being read, since the sidetoc may be referred to again in each HTML page.

```
8625 \newcommand*\LWR@myshorttoc[1]{%
8626 \LWR@traceinfo{\LWR@myshorttoc: #1}%
8627 \LWR@ensuredoingapar%
```

Only if the file exists:

```
8628 \IfFileExists{\jobname.\#1}{%
8629 \LWR@traceinfo{\LWR@myshorttoc: loading}%

```

 Many of the commands in the file will have @ characters in them, so @ must be made a regular letter.

```
8630 \begingroup%
8631 \makeatletter%
```

Read in the TOC file:

```
8632 \@input{\jobname.\#1}%
8633 \endgroup%
8634 }%
8635 {}%
8636 \LWR@traceinfo{\LWR@myshorttoc: done}%
8637 }
```

\LWR@subtableofcontents {<toc/lof/lot>} {<sectionstarname>}

Places a TOC/LOF/LOT at the current position.

```
8638 \NewDocumentCommand{\LWR@subtableofcontents}[m m]{%
```

Closes previous levels:

```
8639 \@ifundefined{chapter}%
8640 {\LWR@closeprevious{\LWR@depthsection}}%
8641 {\LWR@closeprevious{\LWR@depthchapter}}%
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
8642 \LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
8643 \@ifundefined{chapter}{\section*{\#2}}{\chapter*{\#2}}
```

Create a new HTML nav containing the TOC/LOF/LOT:

```
8644 \LWR@htmlelementclass{nav}{#1}
```

Create the actual list:

```
8645 \LWR@myshorttoc{#1}
```

Close the nav:

```
8646 \LWR@htmlelementclassend{nav}{#1}
8647 }
```

\@starttoc {<ext>}

Patch \@starttoc to encapsulate the TOC inside HTML tags:

```
8648 \let\LWR@orig@starttoc\@starttoc
8649
8650 \renewcommand{\@starttoc}[1]{
8651 \LWR@htmlelementclass{nav}{#1}
8652 \LWR@orig@starttoc{#1}
8653 \LWR@htmlelementclassend{nav}{#1}
8654 }
```

Bool LWR@copiedsidetoc Used to only copy the TOC file to the sidetoc a single time.

(listings and perhaps other packages would re-use \tableofcontents for their own purposes, causing the sidetoc to be copied more than once, and thus end up empty.)

```
8655 \newbool{LWR@copiedsidetoc}
8656 \boolfalse{LWR@copiedsidetoc}
```

\tableofcontents Patch \tableofcontents, etc. to print footnotes first. newfloat uses \listoffigures for all future float types.

```
8657 \AtBeginDocument{
8658 \let\LWR@origtableofcontents\tableofcontents
8659
8660 \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:

```
8661 \ifboolexpr{bool{FormatWP} and bool{WPMarkTOC}}{
8662
8663 === table of contents ===
8664
8665 }
8666 {
```

Copy the .toc file to .sidetoc for printing the sidetoc. The original .toc file is renewed when \tableofcontents is finished.

```
8667     \ifboolexpr{LWR@copiedsidetoc}{}{%
8668         \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
8669         \booltrue{LWR@copiedsidetoc}%
8670     }%
8671     \LWR@printpendingfootnotes
8672     \LWR@origtableofcontents
8673 }
8674 }% \tableofcontents
8675 }% AtBeginDocument
```

\listoffigures

```
8676 \let\LWR@origlistoffigures\listoffigures
8677
8678 \renewcommand*{\listoffigures}{
8679 \ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{
8680
8681 === list of figures ===
8682
8683 }
8684 {
8685     \LWR@printpendingfootnotes
8686     \LWR@origlistoffigures
8687 }
8688 }
```

\listoftables

```
8689 \let\LWR@origlistoftables\listoftables
8690
8691 \renewcommand*{\listoftables}{
8692 \ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{
8693
8694 === list of tables ===
8695 }
```

```

8696 }
8697 {
8698     \LWR@printpendingfootnotes
8699     \LWR@origlistoftables
8700 }
8701 }

```

74.2 High-level toc commands

\listof {⟨type⟩} {⟨title⟩}

Emulate the \listof command from the float package (section 213). Used to create lists of custom float types. Also used to redefine the standard L^AT_EX \listoffigures and \listoftables commands.

```

8702 \NewDocumentCommand{\listof}{m +m}{%
8703     @ifundefined{l@#1}{%
8704         \csdef{l@#1}##1##2{\hypertocfloat{1}{#1}{\@nameuse{ext@#1}}{##1}{##2}}%
8705     }{}%
8706     \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}%
8707     \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
8708     \immediate\openout\csname tf@\csname ext@#1\endcsname\endcsname
8709     \jobname.\@nameuse{ext@#1}\relax
8710 }

```

74.3 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:

div.sidetoccontainer: The entire sidetoc.
div.sidetoctitle: The title.
div.sidetoccontents: The table of contents.

```
8711 \end{warpHTML}
```

for HTML & PRINT: 8712 \begin{warpall}

Ctr SideTOCDepth Controls how deep the side-TOC gets. Use a standard L^AT_EX section level similar to tocdepth.

8713 \newcounter{SideTOCDepth}
8714 \setcounter{SideTOCDepth}{1}

\sidetocname Holds the default name for the sidetoc.

8715 \newcommand{\sidetocname}{Contents}
8716 \end{warpall}

for HTML output: 8717 \begin{warpHTML}

\LWR@sidetoc Creates the actual side-TOC.

8718 \newcommand*{\LWR@sidetoc}{
8719 \LWR@forcenewpage
8720 \LWR@stoppars
8721 }

The entire sidetoc is placed into a nav of class sidetoc.

8722 \LWR@htmlelementclass{div}{sidetoccontainer}
8723 \LWR@htmlelementclass{nav}{sidetoc}
8724
8725 \setcounter{tocdepth}{\value{SideTOCDepth}}
8726

The title is placed into a <div> of class sidetoctitle, and may contain paragraphs.

8727 \begin{BlockClass}{sidetoctitle}
8728 \ifcvoid{\thetitle}{}{\InlineClass{sidetocthetitle}{\thetitle}\par}
8729 \sidetocname
8730 \end{BlockClass}

The table of contents is placed into a <div> of class sidetoccontents.

8731 \begin{BlockClass}{sidetoccontents}
8732 \LinkHome
8733
8734 \LWR@myshorttoc{sidetoc}
8735 \end{BlockClass}
8736 \LWR@htmlelementclassend{nav}{sidetoc}
8737 \LWR@htmlelementclassend{div}{sidetoccontainer}
8738 }

74.4 Low-level toc line formatting

\numberline {⟨number⟩}

(Called from each line in the .aux, .lof files.)

Record this section number for further use:

```
8739 \newcommand*{\LWR@numberline}[1]{%
8740 \LWR@sectionnumber{#1}\quad%
8741 }
8742
8743 \LetLtxMacro\numberline\LWR@numberline
```

\hypertoc {⟨1: depth⟩} {⟨2: type⟩} {⟨3: name⟩} {⟨4: page⟩}

Called by \l@section, etc. to create a hyperlink to a section.

The autopage label is always created just after the section opens.

#1 is depth

#2 is section, subsection, etc.

#3 the text of the caption

#4 page number

```
8744 \NewDocumentCommand{\hypertoc}{m m +m m}{%
8745 \LWR@traceinfo{hypertoc !#1!#2!#3!#4!}}%
```

Respond to tocdepth:

```
8746 \ifthenelse{\cnttest{#1}{<=}{\value{tocdepth}}}{%
8747     \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with the name, of the given HTML class.

```
8748     \LWR@subhyperrefclass{%
8749         \LWR@htmlrefsectionfilename{autopage-#4}\LWR@origpound\LWR@print@mbox{autosec-#4}%
8750     }{#3}{toc#2}%
8751     \LWR@stoppars%
8752 }%
8753 {}%
8754 \LWR@traceinfo{hypertoc done}%
8755 }
```

Ctr `lofdepth` TOC depth for figures.

```
8756 \@ifclassloaded{memoir}{}{  
8757 \newcounter{lofdepth}  
8758 \setcounter{lofdepth}{1}  
8759 }
```

Ctr `lotdepth` TOC depth for tables.

```
8760 \@ifclassloaded{memoir}{}{  
8761 \newcounter{lotdepth}  
8762 \setcounter{lotdepth}{1}  
8763 }
```

`\hypertocfloat {<1: depth>} {<2: type>} {<3: ext of parent>} {<4: caption>} {<5: page>}`

#1 is depth

#2 is figure, table, etc.

#3 is lof, lot, of the parent.

#4 the text of the caption

#5 page number

```
8764 \newcommand{\hypertocfloat}[5]{%  
8765 \LWR@startpars
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
8766 \@ifundefined{c@#3depth}{%  
8767 \newcounter{#3depth} %  
8768 \setcounter{#3depth}{1} %  
8769 }{}%
```

Respond to `lofdepth`, etc.:

```
8770 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%  
8771 \ifthenelse{\cnttest{#1}{<=}{\arabic{#3depth}}}{%  
8772 \LWR@startpars%
```

Create an `HTML` link to `filename#autoid-(float number)`, with text of the caption, of the given `HTML` class.

```

8773   \LWR@subhyperrefclass{%
8774     \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
8775     \LWR@origpound\LWR@print@mbox{autoid-\arabic{LWR@nextautoid}}}}%
8776   {#4}{toc#2}%
8777   \LWR@stoppars%
8778 }{}%
8779 }

```

Automatically called by \contentsline:

```
\l@part {\langle name\rangle} {\langle page\rangle}
```

Uses \DeclareDocumentCommand in case the class does not happen to have a \part.

```
8780 \DeclareDocumentCommand{\l@part}{m m}{\hypertoc{-1}{part}{#1}{#2}}
```

```
\l@chapter {\langle name\rangle} {\langle page\rangle}
```

Uses \DeclareDocumentCommand in case the class does not happen to have a \chapter.

```

8781 \@ifundefined{chapter}
8782 {}
8783 {
8784 \DeclareDocumentCommand{\l@chapter}{m m}
8785   {\hypertoc{0}{chapter}{#1}{#2}}
8786 }

```

```
\l@section {\langle name\rangle} {\langle page\rangle}
```

```
8787 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
```

```
\l@subsection {\langle name\rangle} {\langle page\rangle}
```

```
8788 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
```

```
\l@subsubsection {\langle name\rangle} {\langle page\rangle}
```

```
8789 \renewcommand{\l@subsubsection}[2]{\hypertoc{3}{subsubsection}{#1}{#2}}
```

```
\l@paragraph {\langle name\rangle} {\langle page\rangle}
```

```
8790 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
```

```
\l@subparagraph  {\langle name\rangle} {\langle page\rangle}

8791 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
```

```
\l@figure  {\langle name\rangle} {\langle page\rangle}

8792 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
```

```
\l@table  {\langle name\rangle} {\langle page\rangle}

8793 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}
```

```
8794 \end{warpHTML}
```

75 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: 8795 \begin{warpHTML}

8796 \newcounter{LWR@autoindex}
8797 \setcounter{LWR@autoindex}{0}
8798
8799 \newcounter{LWR@autoglossary}
8800 \setcounter{LWR@autoglossary}{0}

Env   theindex

8801 \@ifundefined{chapter}
8802 {\newcommand*{\LWR@indexsection}[1]{\section*{#1}}}
8803 {\newcommand*{\LWR@indexsection}[1]{\chapter*{#1}}}
8804
8805 \AtBeginDocument{
8806 \renewenvironment*{theindex}{%
8807     \LWR@indexsection{\indexname}%
8808     \let\item\LWR@indexitem%
```

```
8809     \let\subitem\LWR@indexsubitem%
8810     \let\subsubitem\LWR@indexsubsubitem%
8811 }{%
8812 }% AtBeginDocument
```

\LWR@indexitem [*index key*] The optional argument is added to support repeatindex.

```
8813 \newcommand{\LWR@indexitem}[1][\emptyset]{%
8814   \InLineClass{\indexitem}{}#1%
8815 }%
```

\LWR@indexsubitem

```
8817 \newcommand{\LWR@indexsubitem}{%
8818   \InLineClass{\indexsubitem}{}%
8819 }%
```

\LWR@indexsubsubitem

```
8821 \newcommand{\LWR@indexsubsubitem}{%
8822   \InLineClass{\indexsubsubitem}{}%
8823 }%
```

\@wrindex {*term*} Redefined to write the LWR@autoindex counter instead of page.

```
8825 \def\LWR@wrindex#1{%
8826   \addtocounter{LWR@autoindex}{1}%
8827   \LWR@new@label{LWRindex-\arabic{LWR@autoindex}}%
8828   \protected@write{\indexfile}{}{%
8829     \string\indexentry{#1}{\arabic{LWR@autoindex}}}%
8830   \endgroup
8831   \esphack
8832 }
8833 \AtBeginDocument{%
8834   \let\@wrindex\LWR@wrindex
8835 }
```

\@wrglossary {*term*} Redefined to write the LWR@latestautopage counter instead of page.

```
8836 \def\@wrglossary#1{%
8837   \addtocounter{LWR@autoglossary}{1}%
8838   \LWR@new@label{LWRglossary-\theLWR@autoglossary}}%
8839 \protected@write{\glossaryfile}{}{
```

```
8840 {\string\glossaryentry{\#1}{\theLWR@autoglossary}}%
8841 \endgroup
8842 \esphack}
```

\LWR@indexnameref {*LWR@autoindex*}

Creates a hyperlink based on the given entry's autoindex.

```
8843 \newcommand*{\LWR@indexnameref}[1]{\nameref{LWRindex-\#1}}
```

\LWR@doindexentry {*LWR@autoindex, or macros.*}

Creates a hyperlink, or handles \see, \textbf{, etc.}

```
8844 \newrobustcmd{\LWR@doindexentry}[1]{%
8845 \IfInteger{\#1}{%
8846   {\LWR@indexnameref{\#1}}{%
8847   {\#1}}{%
8848 }}
```

\LWR@hyperindexrefnullified Handles macros commonly seen inside an \index entry.

To handle additional macros:

```
\appto{\LWR@hyperindexrefnullified}{...}
```

```
8849 \newcommand{\LWR@hyperindexrefnullified}{%
8850 \renewrobustcmd{\emph}[1]{\LWR@HTML@emph{\LWR@doindexentry{\##1}}}{%
8851 \renewrobustcmd{\textbf}[1]{\LWR@HTML@textbf{\LWR@doindexentry{\##1}}}{%
8852 \renewrobustcmd{\textrm}[1]{\LWR@HTML@textrm{\LWR@doindexentry{\##1}}}{%
8853 \renewrobustcmd{\textsf}[1]{\LWR@HTML@textsf{\LWR@doindexentry{\##1}}}{%
8854 \renewrobustcmd{\texttt}[1]{\LWR@HTML@texttt{\LWR@doindexentry{\##1}}}{%
8855 \renewrobustcmd{\textup}[1]{\LWR@HTML@textup{\LWR@doindexentry{\##1}}}{%
8856 \renewrobustcmd{\textsc}[1]{\LWR@HTML@textsc{\LWR@doindexentry{\##1}}}{%
8857 \renewrobustcmd{\textsi}[1]{\LWR@HTML@textsi{\LWR@doindexentry{\##1}}}{%
8858 \renewrobustcmd{\textit}[1]{\LWR@HTML@textit{\LWR@doindexentry{\##1}}}{%
8859 \renewrobustcmd{\textsl}[1]{\LWR@HTML@textsl{\LWR@doindexentry{\##1}}}{%
8860 }
```

\hyperindexref {*LWR@autoindex*}

\hyperindexref{LWR@autoindex} is inserted into *.ind by the makeindex style file lwarp.ist or the xindy style file lwarp.xdy.

```
8861 \newcommand{\hyperindexref}[1]{%
8862 \IfInteger{\#1}{%
```

```

8863      {\LWR@indexnameref{\#1}}%
8864      {%
8865          \begingroup%
8866          \LWR@hyperindexrefnullified
8867          #1%
8868          \endgroup%
8869      }%
8870 }
8871 \end{warpHTML}

```

for PRINT output: A null command for print mode, in case hyperref was not used:

```

8872 \begin{warpprint}
8873 \newcommand{\hyperindexref}[1]{#1}
8874 \end{warpprint}

```

for HTML & PRINT: For the glossaries package, try to prevent an error where \glo@name was not found:

```

8875 \begin{warpall}
8876 \providecommand{\glo@name}{}
8877 \end{warpall}

```

76 Bibliography presentation

for HTML output: 8878 \begin{warpHTML}

```
\bibliography {\langle filenames\rangle}
```

Modified to use the base jobname instead of the _html jobname.

```

8879 \def\bibliography#1{%
8880     \if@filesw
8881         \immediate\write\auxout{\string\bibdata{\#1}}%
8882     \fi
8883 %     \input{\jobname.bbl}%
8884     \begingroup%
8885     \input{\BaseJobname.bbl}%
8886     \endgroup%
8887 }

```

```
\@biblabel {\langle text-refnumber\rangle}
```

```
8888 \renewcommand{\@biblabel}[1]{[\#1]\quad}
```

Env thebibliography To emphasize document titles in the bibliography, the following redefines `\em` inside `thebibliography` to gather everything until the next closing brace, then display these tokens with `\textit`.

Adapted from `embracedef.sty`, which is by TAKAYUKI YATO:

<https://gist.github.com/zr-tex8r/b72555e3e7ad2f0a37f1>

```

8889 \AtBeginDocument{
8890 \AtBeginEnvironment{thebibliography}{
8891 \providecommand*{\LWR@newem}[1]{\textit{\#1}}
8892
8893 \renewrobustcmd{\em}{%
8894   \begingroup
8895     \gdef\LWR@em@after{\LWR@em@finish\LWR@newem}%
8896     \afterassignment\LWR@em@after
8897     \toks@\bgroup
8898 }
8899
8900 \def\LWR@em@finish#1{%
8901   \xdef\LWR@em@after{\noexpand#1{\the\toks@}}%
8902   \endgroup
8903   \LWR@em@after\egroup
8904 }
8905 }% \AtBeginEnvironment{thebibliography}
8906 }% \AtBeginDocument

8907 \end{warpHTML}
```

77 Restoring original formatting

\LWR@restoreorigformatting Used to temporarily restore the print-mode meaning of a number of formatting, graphics, and symbols-related macros while generating SVG math or a `lateximage`.

Must be used inside a group.

Sets `\LWR@formatting` to print until the end of the group.

A number of packages will `\appto` additional actions to this macro.

Various packages add to this macro using `\appto`.

for HTML output: 8908 `\begin{warpHTML}`

```

8909 \newcommand*{\LWR@restoreorigformatting}{%
8910 \LWR@traceinfo{\LWR@restoreorigformatting}%
```

Numerous macros change their print/HTML meaning depending on \LWR@formatting:

```
8911 \renewcommand*{\LWR@formatting}{print}%
8912 \linespread{1}%

8913 \let\par\LWR@origpar%

8914 \LWR@select@print@hskip%  
  
8915 \LetLtxMacro\hfil\LWR@orighfil%
8916 \let\hss\LWR@orighss%
8917 \let\llap\LWR@origllap%
8918 \let\rlap\LWR@origrlap%
8919 \let\hfilneg\LWR@orighfilneg%  
  
8920 \let\,\LWR@origcomma% disable HTML short unbreakable space
8921 \let\thinspace\LWR@origthinspace% disable HTML short unbreakable space
8922 \let\negthinspace\LWR@orignegthinspace% disable HTML negative short unbreakable space
8923 \let\textellipsis\LWR@origtextellipsis%
8924 \let\textless\LWR@origtextless%
8925 \let\textgreater\LWR@origtextgreater%
8926 \LetLtxMacro\rmfamily\LWR@origrmfamily%
8927 \LetLtxMacro\sffamily\LWR@origsffamily%
8928 \LetLtxMacro\ttfamily\LWR@origttfamily%
8929 \LetLtxMacro\bfseries\LWR@origbfseries%
8930 \LetLtxMacro\mdseries\LWR@origmdseries%
8931 \LetLtxMacro\upshape\LWR@origupshape%
8932 \LetLtxMacro\slshape\LWR@origslshape%
8933 \LetLtxMacro\scshape\LWR@origscshape%  
  
8934 \LetLtxMacro\sishape\LWR@origsishape%  
  
8935 \LetLtxMacro\itshape\LWR@origitshape%
8936 \LetLtxMacro\em\LWR@origem%
8937 \LetLtxMacro\normalfont\LWR@orignormalfont%
8938 \let\sp\LWR@origsp%
8939 \let\sb\LWR@origsb%
8940 \LetLtxMacro\textsuperscript\LWR@origtextsuperscript%
8941 \LetLtxMacro\@textsuperscript\LWR@orig@textsuperscript%
8942 \LetLtxMacro\textsubscript\LWR@origtextsubscript%
8943 \LetLtxMacro\@textsubscript\LWR@orig@textsubscript%
8944 \LetLtxMacro\underline\LWR@origunderline%
8945 \let~\LWR@origtilde%
8946 \let\enskip\LWR@origenskip%
8947 \let\quad\LWR@origquad%
8948 \let\qquad\LWR@origqquad%
```

\endtabular must be restored to its original, instead of relying on lwarp's \LWR@formatted mechanism:

```

8949 \LetLtxMacro\endtabular\LWR@origendtabular%
8950 \csletcs{endtabular*}{LWR@origendtabular*}%

8951 \LetLtxMacro\noalign\LWR@orignoalign%
8952 \LetLtxMacro\hline\LWR@orighline%

8953 \let\newline\LWR@orignewline%
8954 \LetLtxMacro\includegraphics\LWR@origincludegraphics%
8955 \let\TeX\LWR@origTeX%
8956 \let\LaTeX\TeX\origLaTeX%
8957 \let\LaTeXe\TeX\origLaTeXe%
8958 \renewcommand*\{X\textsubscript{E}}{X_{E}}%


8959 \LetLtxMacro\@ensuredmath\LWR@origensuredmath%
8960 %
8961 \LWR@restoreorigaccents%
8962 \LWR@restoreoriglists%
8963 %
8964 \LWR@FBcancel%
8965 }

8966 \end{warpHTML}

```

78 Math

78.1 Limitations

78.1.1 Rendering tradeoffs

Math rendering	Math may be rendered as SVG graphics or using the MATHJAX JavaScript display engine.
SVG files	Rendering math as images creates a new SVG file for each expression, except that an MD5 hash is used to combine identical duplicates of the same inline math expression into a single file, which must be converted to SVG only once. Display math is still handled as individual files, since it may contain labels or references which are likely to change.
SVG inline	The SVG images are currently stored separately, but they could be encoded in-line directly into the HTML document. This may reduce the number of files and potentially

speed loading the images, but slows the display of the rest of the document before the images are loaded.

PNG files Others L^AT_EX-to-HTML converters have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but SVG files are the preferred approach for scalable graphics.

MathML Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 11 regarding EPUB output with MATHJAX.

78.1.2 SVG option

SVG math option For SVG math, math is rendered as usual by L^AT_EX into the initial PDF file using the current font¹⁸, then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by L^AT_EX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML alt attribute carries the L^AT_EX code which generated the math, allowing copy/paste of the L^AT_EX math expression into other documents.

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

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

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

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

SVG math copy/paste For SVG math, text copy/paste from the HTML `<alt>` tags lists the equation number or tag for single equations, along with the L^AT_EX code for the math expression. For *AMS* environments with multiple numbers in the same environment, only the first and last is copy/pasted, as a range. No tags are listed inside a starred *AMS* environment, although the `\tag` macro will still appear inside the L^AT_EX math expression.

⚠ **SVG math in TEX boxes** SVG math does not work inside TEX boxes, since a `\newpage` is required before and after each image.

¹⁸See section 456 regarding fonts and fractions.

78.1.3 MATHJAX option

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

Prog MathJax

When MATHJAX is enabled, math is rendered twice:

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

78.1.4 Customizing MATHJAX

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

Examples:

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

78.1.5 MATHJAX limitations

MATHJAX limitations Limitations when using MATHJAX include:

Prog MathJax

subequations

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

footnotes in math

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

lateximage

- Math appearing inside a lateximage, and therefore also inside a Tikz or picture

environment, is rendered as SVG math even if MATHJAX is used in the rest of the document.

siunitx

⚠ **siunitx inside an equation**

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

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

Also see section 9.6.10.

tabbing

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

⚠ **other macros and packages**

- Other math-related macros and packages are not supported by MATHJAX, including \ensuremath and bigdelim, along with occasionally-used macros such as \relax. lwarp emulates footnotes, units, and nicefrac for MathJax.

78.1.6 Catcode changes

preamble macros with math

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

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

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

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

78.1.7 Complicated inline math objects

\inlinemathnormal \inlinemathother

**changing contents
complicated alt tag**

An inline math expression is usually converted to a reusable hashed SVG math image, or a MathJAX expression. The hash or expression depends on the contents of the math expression. In most cases this math expression is static, such as \$x+1\$, so the image can be reused for multiple instances of the same expression. In some cases, the math expression includes a counter or other object which may change between uses. Another problem is complicated contents which do not expand well in an alt tag. The macro \inlinemathother may be used before a dynamic math expression, and \inlinemathnormal after. Doing so tells lwarp to use an unhashed SVG math image, even if MathJAX is in use. See section 44.

78.1.8 Complicated display math objects

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

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

78.2 HTML alt tag names

Redefinable names for the HTML alt tags, for translation according to the reader’s native language.

for HTML & PRINT: 8967 `\begin{warpall}`

`\mathimagename` The HTML alt tag for an SVG math image.
 Default: "math image"

8968 `\newcommand*{\mathimagename}{math image}`

`\packagediagramname` Appended to the `lateximage` HTML alt tag for the images generated by many packages.
 Default: "diagram"

8969 `\newcommand*{\packagediagramname}{diagram}`

8970 `\end{warpall}`

78.3 Inline and display math

for HTML output: 8971 `\begin{warpHTML}`

`Ctr LWR@externalfilecnt` Counter for the external files which are generated and then referenced from the HTML:

8972 `\newcounter{LWR@externalfilecnt}`

Bool LWR@indisplaymathimage True if processing display math for SVG output. Inside a `lateximage`, display math is only set to print-mode output if `LWR@indisplaymathimage` is false. Used to avoid nullifying display math before it has been completed.

```
8973 \newbool{LWR@indisplaymathimage}
```

\\$ Plain dollar signs appearing in the HTML output may be interpreted by MATHJAX to be math shifts. For a plain text dollar `\$`, use an HTML entity to avoid it being interpreted by MATHJAX, unless are inside a `lateximage`, in which case it will not be seen by MATHJAX.

```
8974 \let\LWR@origtextdollar\$
8975
8976 \renewcommand*{\$}{%
8977 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}{%
8978 {\LWR@origtextdollar}%
8979 {\HTMLentity{dollar}}%
8980 }}
```

File lwarp_baseline_marker.png A marker to be used to help `pdfcrop` identify the inline math baseline and width. If either `graphicx` or `graphics` is loaded, this marker is placed at the lower left and lower right corners of the inline math. `pdfcrop` is then able to identify the width of the image, and also the height of an image such as a horizontal dash which does not otherwise touch the baseline.

A marker with alpha or opacity of 0% is not registered by `pdfcrop`, so the marker is a small square block of 1% alpha, which seems to work while still being effectively invisible in the final SVG image.

If `graphicx` is loaded, this marker is sized as a tiny 1 sp square. If `graphics` is loaded, this marker is used at its default size of around .25 pt. If neither `graphics` package is loaded, the marker is replaced by a 10 sp horizontal space, and there is no assistance for determining baseline or width of the inline math image. The best results are obtained when using `graphicx`.

\LWR@addbaselinemarker Places a small marker in an SVG inline image. If `graphics` or `graphicx` are loaded, the marker is a mostly transparent image. If neither is loaded, no marker is used.

```
8981 \AtBeginDocument{
8982
8983 \ifpdf
8984   \newcommand*{\LWR@baselinename}{lwarp_baseline_marker.png}
8985 \else
8986   \ifXeTeX
8987     \newcommand*{\LWR@baselinename}{lwarp_baseline_marker.png}
8988   \else
8989     \newcommand*{\LWR@baselinename}{lwarp_baseline_marker.eps}
```

```

8990     \fi
8991 \fi
8992
8993 \IfFileExists{\LWR@baselinename}%
8994 {
8995     \@ifpackageloaded{graphicx}{%
8996         \newcommand*{\LWR@addbaselinemarker}{%
8997             \LWR@origincludegraphics{\LWR@baselinename}%
8998         }%
8999     }{%
9000         \@ifpackageloaded{graphics}{%
9001             \newcommand*{\LWR@addbaselinemarker}{%
9002                 \LWR@origincludegraphics{\LWR@baselinename}%
9003             }%
9004         }{%
9005             \PackageWarning{lwarp}{Load graphicx or graphics
9006             for improved SVG math baselines,}%
9007             \newcommand*{\LWR@addbaselinemarker}{}%
9008         }%
9009     }%
9010 }{%
9011     \PackageWarning{lwarp}{File \LWR@baselinename\space is not installed alongside
9012     the lwarp-*.sty files, so SVG math baselines may not be accurate,}%
9013     \newcommand*{\LWR@addbaselinemarker}{}%
9014 }%
9015
9016 }% AtBeginDocument

\LWR@subsingle$ * {\langle 2: alt text\rangle} {\langle 3: add'l hashing\rangle} {\langle 4: math expression\rangle}

```

For inline math. Uses MathJax, or for SVG math the image is measured and adjusted to the baseline of the HTML output, and placed inside a `teximage`.

image filename hashing If starred, a hashed filename is used. If so, the hash is based on the alt tag and also the additional hashing argument.

This may be used to provide an expression with a simple alt tag but also enough additional information to provide a unique hash.

An example is when the expression is a complicated TeX expression, which would not copy/paste well. A simplified tag may be used, while the complicated expression is duplicated in the additional hashing argument.

Another example is when the expression is simple, but the image depends on options. These options may be decoded into text form and included in the additional hashing argument in order to make the hash unique according to the set of options, even if the simple alt tag is still the same.

```
9017 \newlength{\LWR@singledollarwidth}
```

```

9018 \newlength{\LWR@singledollarheight}
9019 \newlength{\LWR@singledollardepth}
9020
9021 \newsavebox{\LWR@singledollarbox}
9022
9023 \NewDocumentCommand{\LWR@subsingledollar}{s m m m}{%
9024 \LWR@traceinfo{\LWR@subsingledollar}%
9025 \ifnumcomp{\value{\LWR@lateximagedepth}}{>}{0}{%
9026 {%
9027 \LWR@traceinfo{\LWR@subsingledollar: already in a lateximage}%
9028     #4% contents
9029 }%
9030 {%
9031 \begingroup%

```

MathJax cannot parse the often complicated TeX expressions which appear in the various uses of `\ensuredmath`. `\ensuremath` forces the alt tag to “(math image)”, as translated according to `\mathimagename`. If this is the case, force the use of a `lateximage` even if MathJax. Likewise for `siunitx` if `parse-numbers=false`.

If MathJax, or if formatting math for a word processor, and not `\ensuredmath`, and not a dynamic math expression, print the math expression:

```

9032 \ifboolexpr{%
9033     (
9034         bool{mathjax} or
9035         ( bool{FormatWP} and bool{WPMarkMath} )
9036     ) and
9037     ( not test { \ifstrelational{#2}{(\mathimagename)} } ) and % from \ensuredmath
9038     ( not bool{LWR@dynamicmath} )
9039 }%

```

For MATHJAX, print the math between `\(`` and `\)``:

```

9040 {%
9041     \LWR@traceinfo{\LWR@subsingledollar: Mathjax}%
9042     {\textbackslash(\textbackslashLWR@HTMLsanitize{\#4}\textbackslash)}%
9043 }% mathjax

```

For SVG, print the math inside a `lateximage`, with an `<alt>` tag of the LATEX code, and a css style to control the baseline adjustment.

```

9044 {%
9045 \LWR@traceinfo{\LWR@subsingledollar: NOT mathjax, or is ensuremath, or is dynamic}%

```

Measure the depth, width, and height of the math image:

```
9046 \begingroup%
```

Temporarily disable formatting while measuring the image parameters:

```
9047 \LWR@restoreorigformatting%
9048 \RenewDocumentEnvironment{lateximage}{s o o}{}{}% inside group
9049 \LWR@print@normalsize%
```

Temporarily set font for the HTML PDF output:

```
9050 \LWR@traceinfo{Using font family \LWR@f@family}%
9051 \cnameuse{\LWR@orig\LWR@f@family family}%
9052 \LWR@traceinfo{Using font series \LWR@f@series}%
9053 \cnameuse{\LWR@orig\LWR@f@series series}%
9054 \LWR@traceinfo{Using font shape \LWR@f@shape}%
9055 \cnameuse{\LWR@orig\LWR@f@shape shape}%
```

`lateximagedepth` must be nested to avoid generating paragraph tags. *AMS* math modifies the `\text` macro such that `\addtocounter` does not always occur as expected. Lower-level code is used instead.

```
9056 \global\advance\c@LWR@lateximagedepth 1\relax%
```

Typeset and save the contents, depending on how they were generated:

SVG math: `\LWR@origensuredmath` is part of argument #4.

SVG math \ensuremath: `\LWR@origensuredmath` is part of argument #4.

SVG dynamic math: `\LWR@origensuredmath` is part of argument #4.

Mathjax: Argument #4 is the contents of the math expression without `\LWR@origensuredmath`.
This case is handled above.

Mathjax \ensuremath: `\LWR@origensuredmath` is part of argument #4.

Mathjax dynamic math: Argument #4 is the contents of the math expression without `\LWR@origensuredmath`, so `\LWR@origensuredmath` is added below.

\ifmmode: Included “just in case”.

```
9057 \ifmmode%
9058   \global\sbox{\LWR@singledollarbox}{#4}%
9059 \else%
9060   \ifbool{\LWR@dynamicmath}{}{%
9061     \ifbool{\mathjax}{}{%
9062       \global\sbox{\LWR@singledollarbox}{\LWR@origensuredmath{#4}}%
9063     }%
9064   }%
```

```

9064           \global\sbox{\LWR@singledollarbox}{#4}%
9065       }%
9066   }{%
9067       \global\sbox{\LWR@singledollarbox}{#4}%
9068   }%
9069   \fi%

```

Add a small and almost transparent marker at the depth of the image.

A math minus sign has the same depth as a plus, even though it does not draw anything below the baseline. This means that *pdfcrop* would crop the image without depth. The marker below the baseline is seen by *pdfcrop* and preserves the depth.

```

9070   \global\sbox{\LWR@singledollarbox}{%
9071       \usebox{\LWR@singledollarbox}%
9072       \raisebox{-\dp\LWR@singledollarbox}{%
9073           \LWR@addbaselinemarker}%
9074   }%
9075 }%

```

More low-level code to undo the counter change.

```
9076   \global\advance\c@LWR@lateximagedepth -1\relax% Due to AmS \text macro.
```

Measure the depth:

```

9077   \setlength{\LWR@singledollardepth}{%
9078       \LateximageFontScale\dp\LWR@singledollarbox}%
9079 }%

```

Make the length a global change:

```
9080   \global\LWR@singledollardepth=\LWR@singledollardepth%
```

Likewise for width:

```

9081   \setlength{\LWR@singledollarwidth}{%
9082       \LateximageFontScale\wd\LWR@singledollarbox}%
9083 }%
9084   \global\LWR@singledollarwidth=\LWR@singledollarwidth%

```

Likewise for total height:

```

9085   \setlength{\LWR@singledollarheight}{%
9086       \LateximageFontScale\ht\LWR@singledollarbox}%
9087 }%
9088   \addtolength{\LWR@singledollarheight}{%
9089       \LateximageFontScale\dp\LWR@singledollarbox}%

```

```

9090      }%
9091      \global\let\singledollarheight=\LWR@singledollarheight%
9092      \endgroup%

```

Set a style for the the height or width. The `em` unit is used so that the math scales according to the user's selected font size.

Start with the greater of the width or the height, biased towards the width:

```

9093      \ifdimgreater{\LWR@singledollarwidth}{.7\LWR@singledollarheight}{%
9094          \def\LWR@singledollarstyle{%
9095              width:\LWR@convertto{em}{\the\LWR@singledollarwidth} em%
9096          }%
9097      }{%
9098          \def\LWR@singledollarstyle{%
9099              height:\LWR@convertto{em}{\the\LWR@singledollarheight} em%
9100          }%
9101      }%

```

If a very narrow width, use the height.

```

9102      \ifdimless{\LWR@singledollarwidth}{.2em}{%
9103          \def\LWR@singledollarstyle{%
9104              height:\LWR@convertto{em}{\the\LWR@singledollarheight} em%
9105          }%
9106      }%
9107      \{}%
9108      \}%

```

If very wide and short, use the width:

```

9109      \ifdimless{\LWR@singledollarheight}{.2em}{%
9110          \def\LWR@singledollarstyle{%
9111              width:\LWR@convertto{em}{\the\LWR@singledollarwidth} em%
9112          }%
9113      }%
9114      \{}%
9115      \}%

```

If there is significant text depth, add the depth to the style.

```

9116      \ifdimgreater{\LWR@singledollardepth}{0.05ex}{%
9117          \def\LWR@singledollardepthstyle{%
9118              \ ; % extra space
9119              \LWR@print@mbox{%
9120                  vertical-align:-\LWR@convertto{em}{\the\LWR@singledollardepth} em%
9121              } % extra space

```

```

9122      }%
9123  }{%
9124      \def\LWR@singledollardepthstyle{}%
9125  }%

```

Create the `lateximage` using the alternate tag and the computed size and depth. The star causes `lateximage` to use an MD5 hash as the filename. When hashing, also include the current font and color in the hash.

```

9126  \ifbool{LWR@dynamicmath}{%
9127      \LWR@traceinfo{subsingledollar: dynamic}%
9128      \begin{lateximage}% no hashing
9129          [(\mathimagername)]% alt tag
9130          []% no add'l hashing
9131          [\LWR@singledollarstyle \LWR@singledollardepthstyle]%
9132          CSS
9133  }{%
9134      \LWR@traceinfo{subsingledollar: static}%
9135      \IfValueTF{#1}{%
9136          \LWR@findcurrenttextcolor% sets \LWR@tempcolor
9137          \begin{lateximage}*% use hashing
9138              [#2]%
9139              [% addl' hashing
9140              #3%
9141              FM\LWR@f@family%
9142              SR\LWR@f@series%
9143              SH\LWR@f@shape%
9144              CL\LWR@tempcolor%
9145              ]%
9146          [\LWR@singledollarstyle \LWR@singledollardepthstyle]%
9147      }{%
9148          \begin{lateximage}% no hashing
9149              [#2]%
9150              [% no add'l hashing
9151              [\LWR@singledollarstyle \LWR@singledollardepthstyle]%
9152      }%
9153  }%

```

Place small and almost transparent markers on the baseline at the left and right edges of the image. These markers are seen by `pdfcrop`, and force vertically-centered objects such as a dash to be raised off the baseline in the cropped image, and also force the total width and left/right margins to be correct. (Except that in some fonts a character may exceed the bounding box, and thus may appear wider than expected when converted to an image.)

```
9153  \LWR@addbaselinemarker%
```

Typeset the contents:

```
9154     \usebox{\LWR@singledollarbox}%
```

The closing baseline marker:

```
9155     \LWR@addbaselinemarker%
```

```
9156     \end{lateximage}%
9157 %
9158 }% not mathjax
9159 \endgroup%
9160 }% not in a lateximage
9161 \LWR@traceinfo{\LWR@subsingle$done}%
9162 }
```

```
9163 \LetLtxMacro{\LWR@origsingle$}
9164 \LetLtxMacro{\LWR@secondorigsingle$}{% balance for editor syntax highlighting}
```

```
9165 \LetLtxMacro{\LWR@origopenparen}(%
9166 \LetLtxMacro{\LWR@origcloseparen}%
9167 \LetLtxMacro{\LWR@origopenbracket}[%
9168 \LetLtxMacro{\LWR@origclosebracket}]
```

\$ Redefine the dollar sign to place math inside a `lateximage`, or use MATHJAX:
\$\$

```
9169 \begingroup
9170 \catcode`$=\active%
9171 \protected\gdef${\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

Used by `chemformula` to escape single-dollar math:

```
9172 \protected\gdef\newsingle${\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

\LWR@doubledollar Redefine the double dollar sign to place math inside a `lateximage`, or use MATHJAX:

```
9173 \protected\gdef\double$##1${%
```

If MATHJAX or formatting for a word processor, print the L^AT_EX expression:

```
9174 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{%
```

For MATHJAX, print the math between `\[` and `\]`:

```
9175 {
9176
9177     \textbackslash[%
9178     \LWR@HTMLsanitize{#1}%
9179     \textbackslash]
```

```
9180
9181 }% mathjax
```

For SVG, print the math inside a `lateximage`, with an `<alt>` tag of the L^AT_EX code:

```
9182 {%
  not mathjax
  \begin{BlockClass}{displaymath}%
  \LWR@newautoidanchor%
  \booltrue{\LWR@indisplaymathimage}%
  \begin{lateximage}%
  [%
    \textbackslash{} % extra space
    \LWR@HTMLsanitize{\#1} % extra space
    \textbackslash{}%
  ]%
  \LWR@origdollar\LWR@origdollar\#1\LWR@origdollar\LWR@origdollar%
  \end{lateximage}%
  \end{BlockClass}%
}%
9195 }%
9196 }%
```

`\LWR@singledollar {<alt text>} {<math expression>}`

```
9197 \protected\gdef\lWR@singledollar#1${%
9198 \ifbool{mathjax}{%
9199   \LWR@subsingledollar*%
9200   {%
9201     \textbackslash{} %
9202     \LWR@HTMLsanitize{\#1} % extra space
9203     \textbackslash{}%
9204   }%
9205   {singledollar}% add'l hashing
9206   {\#1}% contents
9207 }%
9208   \LWR@subsingledollar*%
9209   {%
9210     \textbackslash{} %
9211     \LWR@HTMLsanitize{\#1} % extra space
9212     \textbackslash{}%
9213   }%
9214   {singledollar}% add'l hashing
9215   {\LWR@origensuredmath{\#1}}% contents
9216 }%
9217 }
```

`\(` Redefine to the above dollar macros.`

`\[`

```
9218 \AtBeginDocument{
```

```

9219 \protected\gdef\(#1\){$#1$}
9220 \protected\gdef\[#1]{${#1}$$}
9221 }
9222
9223 \endgroup

9224 \AtBeginDocument{
9225 \LetLtxMacro{\LWR@openbracketnormal}{[}
9226 \LetLtxMacro{\LWR@closebracketnormal}{]}
9227 }
```

\@ensuredmath {*expression*}

If MathJax, a `lateximage` is used, since `\ensuremath` is often used for complex TeX expressions which MathJax may not render. If SVG math, a hashed file is used with a simple `alt` tag, but additional hashing provided by the contents.

```

9228 \LetLtxMacro{\LWR@origensuredmath}{\@ensuredmath}
9229
9230 \renewcommand{\@ensuredmath}[1]{%
9231 \ifbool{mathjax}{%
9232     \LWR@subsingle dollar*{(\mathimagename)}{%
9233         \protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}{%
9234     }{%
9235         \LWR@origensuredmath{#1}{%
9236     }{%
9237 }}{%
9238 }{%
9239     SVG math
9240 }
```

If already inside a `lateximage` in math mode, continue as-is.

```

9238 \ifmmode{%
9239     \LWR@origensuredmath{#1}{%
9240 }\else{%
```

Create an inline math `lateximage` with a simple `alt` tag and additional hashing according to the contents.

```

9241 \ifnumcomp{\value{\LWR@lateximagedepth}}{>}{0}{%
9242 {\LWR@origensuredmath{#1}}{%
9243 {%
9244     \LWR@subsingle dollar*{(\mathimagename)}{%
9245         \protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}{%
9246     }{%
9247         \LWR@origensuredmath{#1}{%
9248     }{%
9249     }{%
9250 }\fi{%
```

```
9251 }%
9252 }
```

Remove the old `math` and `displaymath` environments:

```
9253 \let\math\relax
9254 \let\endmath\relax
9255 \let\displaymath\relax
9256 \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`.

```
9257 \NewEnviron{math}{\expandafter{(\BODY)}}
```

Env `LWR@displaymathnormal` Set math mode then typeset the body of what was between the begin/end. See the `environ` package for `\BODY`.

```
9258 \NewEnviron{LWR@displaymathnormal}{\expandafter[\BODY]\@ignoretrue}
```

Set the default `displaymath` to the normal version:

```
9259 \LetLtxMacro{[}\LWR@openbracketnormal%
9260 \LetLtxMacro{]}\LWR@closebracketnormal%
9261 \LetLtxMacro{\displaymath}{\LWR@displaymathnormal}%
9262 \LetLtxMacro{\enddisplaymath}{\endLWR@displaymathnormal}%
```

Env `LWR@displaymathother` A version of `displaymath` which can handle complicated objects, but does not supply MATHJAX or HTML alt tags.

```
9263 \newenvironment{LWR@displaymathother}
9264 {%
9265   \begin{BlockClass}{displaymath}%
9266   \LWR@newauto{danchor}%
9267   \booltrue{LWR@indisplaymathimage}%
9268   \begin{lateximage}%
9269   [(display math)]%
9270   \LWR@origdollar\LWR@origdollar%
9271 }%
9272 {%
9273   \LWR@origdollar\LWR@origdollar%
9274   \end{lateximage}%
9275   \end{BlockClass}%
9276 }
```

Env `LWR@equationother` A version of `displaymath` which can handle complicated objects, but does not supply MATHJAX or HTML alt tags.

```

9277 \newenvironment{LWR@equationother}
9278 {%
9279   \begin{BlockClass}{displaymathnumbered}%
9280   \LWR@newauto{danchor}%
9281   \booltrue{LWR@indisplaymathimage}%
9282   \begin{lateximage}%
9283     [(display math)]%
9284     \LWR@origequation%
9285   }%
9286 {%
9287   \LWR@origendequation%
9288   \end{lateximage}%
9289   \end{BlockClass}%
9290 }
```

78.4 MATHJAX support

Ctr `LWR@nextequation` Used to add one to compute the next equation number.

```
9291 \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.

```
9292 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```

9293 \ifcsdef{thechapter}%
9294 \InlineClass{hidden}%
9295 \textbackslash%
9296 \textbackslash{}seteqsection \{\thechapter\}%
9297 \textbackslash%
9298 }%
9299 }%
9300 {}% not using chapters
```

MATHJAX doesn't allow setting the equation number to 1:

```
9301 \ifthenelse{\cnttest{\value{equation}}>0}{  
9302 {
```

Tell MATHJAX that the next set of equations begins with the current L^AT_EX equation number, plus one.

```
9303     \setcounter{LWR@nextequation}{\value{equation}}  
9304     \addtocounter{LWR@nextequation}{1}
```

Place the MATHJAX command inside “\(” and “\)” characters, to be printed to HTML, not interpreted by L^AT_EX.

```
9305     \InLineClass{hidden}{  
9306         \textbackslash(  
9307             \textbackslash{}seteqnumber \{\arabic{LWR@nextequation}\}  
9308             \textbackslash)  
9309     }  
9310 }{\% not eq > 0  
9311 }
```

\LWR@hidelatexequation {*environment*} {*contents*}

Creates the L^AT_EX version of the equation inside an HTML comment.

```
9312 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```
9313 \LWR@stopars  
9314 \LWR@htmlopencomment  
9315
```

Start the L^AT_EX math environment inside the HTML comment:

```
9316 \begingroup  
9317 \nameuse{LWR@orig#1}
```

While in the math environment, restore various commands to their L^AT_EX meanings.

```
9318 \LWR@restoreorigformatting
```

See \LWR@htmlmathlabel in section 78.7.1.

Print the contents of the equation:

```
9319 #2
```

End the L^AT_EX math environment inside the HTML comment:

```
9320 \@nameuse{LWR@origend#1}
9321 \endgroup
9322
```

Close the HTML comment and resume HTML paragraph handling:

```
9323 \LWR@htmlclosecomment
9324 \LWR@startpars
9325 }
```

\LWR@addmathjax {*environment*} {*contents*}

Given the name of a math environment and its contents, create a MATHJAX instance. The contents are printed to HTML output, not interpreted by L^AT_EX.

```
9326 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
```

Enclose the MATHJAX environment inside printed “\(`” and “\)`” characters.

```
9327 \LWR@origtilde\LWR@orignewline
9328 \textbackslash{}begin{\#1\`}
```

Print the contents, sanitizing for HTML special characters.

```
9329 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\#2}}
```

Close the MATHJAX environment:

```
9330 \textbackslash{}end{\#1\`}
9331 \LWR@orignewline
9332 }
```

78.5 Equation environment

Remember existing equation environment:

```
9333 \AtBeginDocument{
9334 \let\LWR@origequation\equation
9335 \let\LWR@origendequation\endequation
9336 \csletcs{\LWR@origequation*}{\equation*}
9337 \csletcs{\LWR@origendequation*}{\endequation*}
9338 }
```

```
\LWR@doequation {\{env contents\}} {\{env name\}}
```

For SVG math output, the contents are typeset using the original equation inside a `lateximage`, along with an `<alt>` tag containing a detokenized copy of the L^AT_EX source for the math.

For MATHJAX output, the contents are typeset in an original equation environment placed inside a HTML comment, with special processing for `\labels`. The contents are also printed to the HTML output for processing by the MATHJAX script.

```
9339 \newcommand*\{\LWR@doequation\}[2]{%
9340 }
```

If `mathjax` or `FormatWP`, print the L^AT_EX expression:

```
9341 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{%
```

MATHJAX output:

```
9342 {
```

Print commands to synchronize MATHJAX's equation number and format to the current L^AT_EX chapter/section and equation number:

```
9343 \LWR@syncmathjax
```

Print the L^AT_EX math inside an HTML comment:

```
9344 \LWR@hidelatexequation{\#2}{\#1}
9345 }
```

SVG output: Create the `lateximage` along with an HTML `<alt>` tag having an equation number, the L^AT_EX equation environment commands, and the contents of the environment's `\BODY`.

```
9346 {%
  not mathjax
```

Begin the `lateximage` with an `<alt>` tag containing the math source:

```
9347 \ifstreq{\#2}{equation*}{%
9348   \begin{BlockClass}{displaymath}%
9349 }{%
9350   \begin{BlockClass}{displaymathnumbered}%
9351 }{%
9352 \LWR@newautoidanchor%
9353 \booltrue{\LWR@indisplaymathimage}%
9354 \begin{lateximage}[%
```

```

9356           \ifdefeq{\LWR@equationtag}{\theequation}{%
9357 %                           no tag was given
9358           }{%
9359             (\LWR@equationtag) % tag was given
9360           }{%
9361             }{%
9362               (\LWR@equationtag) % automatic numbering
9363             }{%
9364               \textbackslash{begin}\{\#2\}} % extra space
9365               \LWR@HTMLsanitizeexpand{\detokenize\expandafter\{\#1\}} % extra space
9366               \textbackslash{end}\{\#2\}}%
9367 ]% alt tag

```

Create the actual L^AT_EX-formatted equation inside the `lateximage` using the contents of the environment.

```

9368     \nameuse{LWR@orig#2}%
9369     #1% contents collected by \collect@body
9370     \nameuse{LWR@origend#2}%
9371     \end{lateximage}%
9372     \end{BlockClass}%
9373 }% not mathjax
9374 }

```

After the environment, if MATHJAX, print the math to the HTML output for MATHJAX processing:

```

9375 \newcommand*\LWR@doendequation}[1]{%
9376   \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{%
9377     }{%
9378       \LWR@addmathjax{\#1}{\BODY}%
9379     }{}}%
9380
9381 }

```

Remove existing equation environment:

```

9382 \AtBeginDocument{
9383 \let\equation\relax
9384 \let\endequation\relax
9385 \csletcs{equation*}{\relax}
9386 \csletcs{endequation*}{\relax}
9387 }

```

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`.

```

9388 \AtBeginDocument{
9389 \NewEnviron{equation}{%
9390 {\LWR@doequation{\BODY}{equation}}%
9391 [\LWR@doendequation{equation}]%
9392
9393 \LetLtxMacro{\LWR@equationnormal}{\equation}
9394 \LetLtxMacro{\endLWR@equationnormal}{\endequation}
9395 }

Env   equation*

9396 \AtBeginDocument{
9397 \NewEnviron{equation*}{%
9398 {\LWR@doequation{\BODY}{equation*}}%
9399 [\LWR@doendequation{equation*}]%
9400
9401 \csletcs{\LWR@equationnormalstar}{\equation*}
9402 \csletcs{\LWR@endequationnormalstar}{\endequation*}
9403 }

```

Remember the “less” version of `equation`, which uses MATHJAX and `alt` tags, but does not support complicated contents such as some `Tikz` expressions.

```

9404 \AtBeginDocument{
9405 \LetLtxMacro{\LWR@equationless}{\equation}
9406 \LetLtxMacro{\endLWR@equationless}{\endequation}
9407 \csletcs{\LWR@equationlessstar}{\equation*}
9408 \csletcs{\LWR@endequationlessstar}{\endequation*}
9409 }

```

78.6 \displaymathnormal and \displaymathother

- | | |
|--|---|
| <code>\displaymathnormal</code>
<code>\displaymathother</code>
MathJax unsupported
complicated alt tag | <p>By default, or when selecting <code>\displaymathnormal</code>, MATHJAX math display environments print their contents as text into HTML, and SVG display math environments render their contents as SVG images and use their contents as the <code>alt</code> tag of HTML output. To do so, the contents are loaded into a macro for reuse. In some cases, such as complicated <code>Tikz</code> pictures, compilation will fail.</p> <p>When selecting <code>\displaymathother</code>, it is assumed that the contents are more complicated than “pure” math. An example is an elaborate <code>Tikz</code> picture, which will not render in MATHJAX and will not make sense as an HTML <code>alt</code> tag. In this mode, MATHJAX is turned off, math display environments become SVG images, even if MATHJAX is selected, and the HTML <code>alt</code> tags become simple messages. The contents are internally processed as an environment instead of a macro argument, so complicated objects such as <code>Tikz</code> pictures are more likely to compile successfully.</p> |
|--|---|

\displaymathnormal simple math objects Use when display math environments have simple math which is to sent to MATHJAX or included in HTML alt tags.

```

9410 \newcommand*{\displaymathnormal}{%
9411 \ifbool{LWR@origmathjax}{\booltrue{mathjax}}{\boolfalse{mathjax}}%
9412 \LetLtxMacro{\[LWR@openbracketnormal}%
9413 \LetLtxMacro{\]}{LWR@closebracketnormal}%
9414 \LetLtxMacro{\displaymath}{LWR@displaymathnormal}%
9415 \LetLtxMacro{\enddisplaymath}{\endLWR@displaymathnormal}%
9416 \LetLtxMacro{\equation}{\LWR@equationnormal}%
9417 \LetLtxMacro{\endequation}{\endLWR@equationnormal}%
9418 \csletcs{equation*}{LWR@equationnormalstar}%
9419 \csletcs{endequation*}{LWR@endequationnormalstar}%
9420 }

```

\displaymathother complicated math objects Use when display math environments have complicated objects which will not work with MathJax or should not be included in HTML alt tags. Complicated contents are more likely to compile correctly.

```

9421 \newcommand*{\displaymathother}{%
9422 \boolfalse{mathjax}}%
9423 \LetLtxMacro{\displaymath}{LWR@displaymathother}%
9424 \LetLtxMacro{\enddisplaymath}{\endLWR@displaymathother}%
9425 \LetLtxMacro{\[}{\LWR@displaymathother}%
9426 \LetLtxMacro{\]}{\endLWR@displaymathother}%
9427 \LetLtxMacro{\equation}{\LWR@equationother}%
9428 \LetLtxMacro{\endequation}{\endLWR@equationother}%
9429 \csletcs{equation*}{displaymath}%
9430 \csletcs{endequation*}{enddisplaymath}%
9431 }

```

9432 \end{warpHTML}

for PRINT output: 9433 \begin{warpprint}

Print-mode versions:

```

9434 \newcommand*{\displaymathnormal}{}%
9435 \newcommand*{\displaymathother}{}%

```

9436 \end{warpprint}

for HTML output: 9437 \begin{warpHTML}

78.7 AMS Math environments

78.7.1 Support macros

Bool LWR@amsmultiline True if processing a multiline environment.

To compensate for `multiline`-specific code, LWR@amsmultiline is used to add extra horizontal space in \LWR@htmlmathlabel if it is used in an amsmath environment which is not a `multiline` environment and not an equation.

```
9438 \newbool{LWR@amsmultiline}
9439 \boolfalse{LWR@amsmultiline}
```

```
\LWR@htmlmathlabel {\langle label\rangle}
```

lwarp points \ltx@label here. This is used by \label when inside a L^AT_EX AMS math environment's math display environment.

\LWR@origltx@label points to the L^AT_EX original, modified by lwarp, then by amsmath, then by cleveref.

```
9440 \newcommand*{\LWR@htmlmathlabel}[1]{%
9441 \LWR@traceinfo{\LWR@htmlmathlabelb #1}%
}
```

If `mathjax` or `FormatWP`, print the L^AT_EX expression:

```
9442 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9443 {%
```

The combined L^AT_EX & HTML label is printed in a `\text` field:

```
9444 \text{%
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
9445 \ifbool{LWR@amsmultiline}{}{\hspace*{\totwidth@}}%
```

Temporarily end the HTML comment, insert the L^AT_EX & HTML label, then resume the HTML comment. \@firstofone is required to remove extra braces introduced by the amsmath package.)

```
9446 \LWR@htmlclosecomment%
9447 \LWR@origltx@label{\#1}%
9448 \LWR@htmlopencomment%
9449 }% text
```

```

9450 }% mathjax
9451 {%
9452     \LWR@origltx@label{#1}%
9453 }%
9454 }

```

\LWR@beginhideamsmath Starts hiding L^AT_EX math inside an HTML comment.

```

9455 \newcommand*{\LWR@beginhideamsmath}{%
9456 \LWR@stoppars
9457 \LWR@origtilde\LWR@orignewline
9458 \LWR@htmlopencomment
9459
9460 \begingroup
9461 \LWR@restoreorigformatting
9462 }

```

\LWR@endhideamsmath Ends hiding L^AT_EX math inside an HTML comment.

```

9463 \newcommand*{\LWR@endhideamsmath}{%
9464 \endgroup
9465
9466 \LWR@htmclosecomment
9467 \LWR@orignewline
9468 \LWR@startpars
9469 }

```

78.7.2 Environment patches

The amsmath environments already collect their contents in \envbody for further processing. eqnarray is not an *AMS* package, and thus requires special handling.

For SVG math: Each environment is encapsulated inside a *lateximage* environment, along with a special optional argument of \LWR@amsmathbody or \LWR@amsmathbodynumbered telling *lateximage* to use as the HTML <alt> tag the environment's contents which were automatically captured by the *AMS* environment.

For MATHJAX: Each environment is syched with L^AT_EX's equation numbers, typeset with L^AT_EX inside an HTML comment, then printed to HTML output for MATHJAX to process.

Env eqnarray This environment is not an *AMS* environment and thus its body is not automatically captured, so the environ package is used to capture the environment into \BODY.

```

9470 \let\LWR@origeqnarray\eqnarray
9471 \let\LWR@origendeqnarray\endeqnarray

```

To remember whether the starred environment was used, and thus whether to number the equations:

```
9472 \newbool{LWR@numbereqnarray}
9473 \booltrue{LWR@numbereqnarray}
```

Common code used by `eqnarray` and `Beqnarray` (from `fancybox`):

```
9474 \newcommand{\LWR@eqnarrayfactor}{%
```

If `mathjax` or `FormatWP`, print the L^AT_EX expression:

```
9475 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9476 {%
```

If MATHJAX, the environment contents (the `\BODY`) are executed in a HTML comment to trigger the correct equation number increment (if not starred), then are included verbatim in the output for MATHJAX to interpret:

```
9477 \LWR@syncmathjax
9478 \boolfalse{LWR@amsmultiline}
9479 \ifbool{LWR@numbereqnarray}
9480 {
```

If numbering the equations, execute a copy inside an HTML comment block:

```
9481 \LWR@beginhideamsmath
9482 \LWR@origeqnarray
9483 \BODY
9484 \LWR@origendeqnarray
9485 \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MATHJAX to interpret:

```
9486 \LWR@addmathjax{eqnarray}{\BODY}
9487 }%
9488 {%
not LWR@numbereqnarray
```

If not numbering equations, just create the contents for MATHJAX:

```
9489 \LWR@addmathjax{eqnarray*}{\BODY}
9490 }%
LWR@numbereqnarray
9491 }%
mathjax
9492 {%
not mathjax
9493 \ifbool{LWR@numbereqnarray}
9494 {
```

For numbered SVG equations, first create a `lateximage` with an `alt` attribute containing sanitized copy of the source code:

```

9495      \begin{BlockClass}{displaymathnumbered}%
9496      \LWR@newauto{danchor}%
9497      \booltrue{\LWR@indisplaymathimage}%
9498      \begin{lateximage}[(\LWR@startingequationtag--\LWR@equationtag)
9499          \LWR@addmathjax{eqnarray}{\BODY}]
```

Then create the image contents using an actual `eqnarray`:

```

9500      \LWR@origeqnarray
9501      \BODY
9502      \LWR@origendeqnarray
9503      \end{lateximage}
9504      \end{BlockClass}
9505  }%
9506  {%
9507    not \LWR@numbereqnarray
```

If not numbered, do the same, but an extra `\nonumber` seems to be required:

```

9507      \begin{BlockClass}{displaymath}
9508      \LWR@newauto{danchor}%
9509      \booltrue{\LWR@indisplaymathimage}%
9510      \begin{lateximage}[(\LWR@addmathjax{eqnarray*}{\BODY})]
9511          \LWR@origeqnarray
9512          \BODY
9513          \nonumber
9514          \LWR@origendeqnarray
9515          \end{lateximage}
9516          \end{BlockClass}
9517  }%
9518  {%
9519    not \LWR@numbereqnarray
9520 }%
```

Default to number equations in the future:

```

9519 \booltrue{\LWR@numbereqnarray}
9520 }
```

`eqnarray` itself is made with a blank line before and after to force it to be on its own line:

```

9521 \RenewEnviron{eqnarray}
9522 {%
9523
9524 \LWR@eqnarrayfactor
9525
9526 }
```

The starred version is patched to turn off the numbering:

```
9527 \csgpreto{eqnarray*}{\boolfalse{LWR@numbereqnarray}}
9528 \end{warpHTML}
```

79 Lateximages

79.1 Description

Env lateximage A `lateximage` is a piece of the document which is typeset in L^AT_EX then included in the HTML output as an image. This is used for math if `svg` math is chosen, and also for the `picture`, `tikzpicture`, and other environments.

Before typesetting the `lateximage` a large number of formatting, graphics, and symbols-related macros are temporarily restored to their print-mode meaning by `\LWR@restoreorigformatting`. (See section 77.)

A `lateximage` is typeset on its own PDF page inside an HTML comment which starts on the preceding page and ends on following page, and instructions are written to `lateximage.txt` for `lwarpmk` to extract the `lateximage` from the page of the PDF file then generate an accompanying `.svg` file image file. Meanwhile, instructions to show this image are placed into the HTML file after the comment.

An HTML `` is created to hold both the HTML comment, which will have the `pdftotext` conversion, and also the link to the final `.svg` image.

A L^AT_EX label is used to remember which PDF page has the image. A label is used because footnotes, endnotes, and pagenotes may cause the image to appear at a later time. The label is declared along with the image, and so it correctly remembers where the image finally ended up.

HTML alt tag The HTML `alt` tag is set to the L^AT_EX source for `svg` math, some chemistry expressions, and perhaps some other expressions which make sense for text copy/paste. In some other cases, the `alt` tag is set according to the package name.

When creating an `svg` math image, its `alt` tag may be set to the math expression, which may be hashed for image reuse. In the case of `\ensuremath` or after `\inlinemathother`, where the contents require a unique image for each instance of the same expression, the `alt` tag is set to `\mathimagnename`, and the image is not reused.

This expression is visible in the browser if images are not loaded, and appears when the text is copied and pasted. The default is “math image”, and it may be changed

according to the document's language. This may be set in the preamble, or changed as necessary inside the document, where it will affect the following `svg` math images.

For many packages, the output is placed inside a `\textrimage` with an `HTML alt` tag set to the package name followed by `\packagediagramname`. For example:

```
(-xy- diagram)
```

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

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

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

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

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

79.2 Support counters and macros

for HTML output: 9529 `\begin{warpHTML}`

Ctr `LWR@lateximagenumber` Sequence the images.

```
9530 \newcounter{LWR@lateximagenumber}
9531 \setcounter{LWR@lateximagenumber}{0}
```

Ctr `LWR@lateximagedepth` Do not create `\textrimage` inside of `\textrimage`.

```
9532 \newcounter{LWR@lateximagedepth}
9533 \setcounter{LWR@lateximagedepth}{0}
```

A few utility macros to write special characters:

```
9534 \edef\LWR@hashmark{\string#} % for use in \write
9535 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctr `LWR@LIPage` Used to reference the PDF page number of a `\textrimage` to be written into `latexitimages.txt`.

```
9536 \newcounter{LWR@LIPage}
```

```
9537 \end{warpHTML}
```

79.3 Font size

for HTML & PRINT: 9538 \begin{warpall}

\LateximageFontSizeName Declares how large to write text in \lateximages. The .svg file text size should blend well with the surrounding HTML text size.

 **no backslash** *Do not include the leading backslash in the name.*

```
9539 \newcommand*\{\LateximageFontSizeName\}{normalsize}
```

\LateximageFontScale Declares how large to scale inline SVG math images. The .svg file text size should blend well with the surrounding HTML text size. The default is .75, but it may be redefined as needed depending on the HTML font.

```
9540 \newcommand*\{\LateximageFontScale\}{.75}
```

```
9541 \end{warpall}
```

79.4 Sanitizing math expressions for HTML

for HTML output: 9542 \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...

```
9543 \newrobustcmd{\LWR@HTMLsanitize}[1]{%
```

Cancel French babel character handling, and fully expand the strings:

```
9544 \begingroup%
9545 \LWR@FBcancel%
9546 \fullexpandarg%
```

The &, <, and > may be interpreted by the browser:

```
9547 \protect\StrSubstitute{\detokenize{#1}}%
9548 {\detokenize{&}}{\detokenize{&lt;}}[\LWR@strresult]%
9549 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
9550 {\detokenize{<}}{\detokenize{&lt;}}[\LWR@strresult]%
9551 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
9552 {\detokenize{>}}{\detokenize{&gt;}}[\LWR@strresult]%
```

The double quote occasionally causes problems.

```
9553 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
9554 {\detokenize{"}}{\detokenize{"}}[\LWR@strresult]%
```

MathJax allows expressions to be defined with \newcommand. These expressions would appear with ## for each argument, and each must be changed to a single #. This must be done after all the above changes. Attempting another conversion after this causes an error upon further expansion.

```
9555 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
9556 {\detokenize{##}}{\detokenize{\origpound}}[\LWR@strresult]%
9557 \LWR@strresult%
9558 \endgroup%
9559 }
```

```
\LWR@HTMLsanitizeexpand {<text>}
```

This version expands the argument before sanitizing it.

```
9560 \newrobustcmd{\LWR@HTMLsanitizeexpand}[1]{%
```

Cancel French babel character handling, and fully expand the strings:

```
9561 \begingroup%
9562 \LWR@FBcancel%
9563 \fullexpandarg%
```

The difference between this and \LWR@HTMLsanitize (without “expand”) is the following \expandafter:

```
9564 \protect\StrSubstitute{\detokenize\expandafter{\#1}}%
9565 {\detokenize{&}}{\detokenize{&lt;}}[\LWR@strresult]%
```

```

9566 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
9567 {\detokenize{<}}{\detokenize{<}}[\LWR@strresult]%

9568 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
9569 {\detokenize{>}}{\detokenize{>}}[\LWR@strresult]%

9570 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
9571 {\detokenize{"}}{\detokenize{"}}[\LWR@strresult]%

\LWR@HTMLsanitizeexpand is not used for defining new MathJax macros, so the ## conversion is not needed here.

9572 \LWR@strresult%
9573 \endgroup%
9574 }

```

79.5 Equation numbers

Ctr LWR@startingequation For use with `lateximage` and multi-line numbered equations. Remembers the next equation number so that it may be printed in the alt tag.

```

9575 \newcounter{LWR@startingequation}
9576
9577 \@ifundefined{chapter}
9578 {
9579 \renewcommand{\theLWR@startingequation}{%
9580 \arabic{LWR@startingequation}%
9581 }
9582 }
9583 {%
9584 \chapter{%
9585 \renewcommand{\theLWR@startingequation}{%
9586 \ifnumcomp{\value{chapter}}{>}{0}{\arabic{chapter}.}{}%
9587 \arabic{LWR@startingequation}%
9588 }

```

Bool LWR@isstartingequation True for the first equation tag, false for later tags in the same environment.

```
9589 \newbool{LWR@isstartingequation}
```

\LWR@startingequationtag Prints the starting equation number or tag.

```
9590 \let\LWR@startingequationtag\theLWR@startingequation
```

\LWR@equationtag Prints the ending equation number or tag.

This is reset by `lateximage`, may be temporarily overwritten by `\tag` calling `\LWR@remembertag`.

```
9591 \newcommand*{\LWR@equationtag}{}%
```

Only if `svg` math, patch `\tag` after packages have loaded, in case someone else modified `\tag`.

```
9592 \AtBeginDocument{%
9593   \ifbool{mathjax}{}{%
9594     \ifbool{mathjax}{}{%
9595       \NewDocumentCommand{\LWR@remembertag}{m}{%
9596         \ifbool{LWR@isstartingequation}{%
9597           \global\boolfalse{LWR@isstartingequation}%
9598           \xdef\LWR@startingequationtag{\#1}%
9599         }%
9600       }%
9601       \xdef\LWR@equationtag{\#1}%
9602     }%
9603   }%
9604 }% AtBeginDocument
```

\LWR@remembertag {*tag*}

For use inside the math environments while using `svg` math. Sets `\theLWR@startingequation` and `\theequation` to the given tag.

```
9595 \NewDocumentCommand{\LWR@remembertag}{m}{%
9596   \ifbool{LWR@isstartingequation}{%
9597     \global\boolfalse{LWR@isstartingequation}%
9598     \xdef\LWR@startingequationtag{\#1}%
9599   }%
9600 }%
9601 \xdef\LWR@equationtag{\#1}%
9602 }%
9603 }%
9604 }% AtBeginDocument
```

79.6 HTML alt tags

\LWR@amsmathbody {*envname*} For use inside the optional argument to a `lateximage` to add the contents of a AMS math environment to the `<alt>` tag.

```
9605 \newcommand*{\LWR@amsmathbody}[1]{%
9606   \textbackslash\begin{#1} % extra space
9607   \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}%
9608   \textbackslash\end{#1}%
9609 }
```

\LWR@amsmathbodynumbered {*envname*} For use inside the optional argument to a `lateximage` to add the contents of a AMS math environment to the alt tag, prefixed by the equation numbers.

```
9611 \newcommand*\LWR@amsmathbodynumbered[1]
9612 {%
9613 \ifnumcomp{\value{\LWR@startingequation}}{=}{\value{equation}}%
9614 {(\LWR@equationtag)}%
9615 {(\LWR@startingequationtag--\LWR@equationtag)} % extra space
9616 \LWR@amsmathbody{#1} % extra space
9617 }
```

79.7 `lateximage` environment

Env `lateximage` * [*2: <alt> tag*] [*3: add'l hashing*] [*4: css style*]

Typesets the contents and then renders the result as an SVG file. Star causes the image to be hashed for reuse.

The optional `<alt>` tag is included in the HTML code for use with copy/paste.

image filename hashing If starred, a hashed filename is used. If so, the hash is based on the `alt` tag and also the additional hashing argument.

This may be used to provide an expression with a simple `alt` tag but also enough additional information to provide a unique hash.

An example is when the expression is a complicated TeX expression, which would not copy/paste well. A simplified tag may be used, while the complicated expression is duplicated in the additional hashing argument.

Another example is when the expression is simple, but the image depends on options. These options may be decoded into text form and included in the additional hashing argument in order to make the hash unique according to the set of options, even if the simple `alt` tag is still the same.

```
9618 \catcode`\$=\active%
9619
9620 \NewDocumentEnvironment{lateximage}{s O{(image)} O{} O{}}
9621 {%
9622 \LWR@traceinfo{lateximage: starting on \jobname.pdf page \arabic{page}}%
9623 \LWR@traceinfo{lateximage: entering depth is \arabic{\LWR@lateximagedepth}}%
```

Nested `lateximages` remain one large `lateximage`:

```
9624 \ifnumcomp{\value{\LWR@lateximagedepth}}{>}{0}%

```

If nesting inside an already-existing `\teximage`, simply record one more level. *AMS* packages redefine `\addtocounter` to do nothing if inside a `\text`, so lower-level TEX macros are used for tracking nested `\teximages`.

```
9625 {%
9626 %     \addtocounter{LWR@teximagedepth}{1}%
9627     \global\advance\c@LWR@teximagedepth 1\relax% Due to AmS \text macro.
9628 }%
```

Otherwise, this is the outer-most `\teximage`:

```
9629 {% start of outer-most \teximage
```

Remember the next equation number to be allocated, in case it must be printed in a multi-equation environment:

```
9630 \LWR@traceinfo{\teximage: starting outer-most \teximage}%
9631     \setcounter{LWR@startingequation}{\value{equation}}%
9632     \addtocounter{LWR@startingequation}{1}%
9633     \booltrue{LWR@isstartingequation}%
9634     \let\LWR@startingequationtag\theLWR@startingequation%
```

The default equation tag, unless overwritten by `\tag`:

```
9635     \let\LWR@equationtag\theequation%
```

Starting a new `\teximage`:

```
9636     \addtocounter{LWR@teximagenumber}{1}%
9637     \LWR@traceinfo{\teximage: LWR@teximagenumber is \arabic{LWR@teximagenumber}}%
```

While inside a `\teximage`, locally do not use `mathjax`:

```
9638     \boolfalse{mathjax}%

```

Be sure that are doing a paragraph:

```
9639     \LWR@ensuredoingapar%
```

Next file:

```
9640     \addtocounter{LWR@externalfilecnt}{1}%
9641     \LWR@traceinfo{\teximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%
```

Figure out what the next page number will be. `\setcounterpageref` assigns `LWR@LIpage` to the page number for the reference `LWR\teximageXXX`:

```
9642     \setcounter{page}{\arabic{LWR@lateximagearabic{LWR@lateximagenumber}}}%  
9643     \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:

```
9644     \LWR@htmlltag{span id="lateximage\arabic{LWR@lateximagenumber}" % extra space  
9645         class="lateximagesource"}%
```

Write instructions to the lateximages.txt file:

```
9646     \LWR@traceinfo{lateximage: about to write to lateximages.txt}%">  
9647     \IfBooleanTF{#1}{starred}{  
9648     \LWR@traceinfo{lateximage: hash true, adding %  
9649         !\detokenize\expandafter{#2}!\detokenize\expandafter{#3}!}%"}
```

Compute and save the hashed file name for later use:

```
9651     \edef\LWR@hashedname{  
9652         \LWR@mdfive{\detokenize\expandafter{#2}-!-#3}}%  
9653     \LWR@mdfive{\detokenize\expandafter{#2}-!-\detokenize\expandafter{#3}}%  
9654 }%  
9655     \LWR@traceinfo{lateximage: hash is \LWR@hashedname}%"
```

Write the page, hashing, and hashed name:

```
9656     \immediate\write\LWR@lateximagesfile{  
9657         |\arabic{LWR@LIpage}|true|\LWR@hashedname|%"  
9658     }%"  
9659     }% hash  
9660     }% no hash
```

No hash, so write the page, no hashing, and the image number:

```
9661     \LWR@traceinfo{lateximage: hash false}%"  
9662     \immediate\write\LWR@lateximagesfile{  
9663         |\arabic{LWR@LIpage}|false|lateximage-\arabic{LWR@externalfilecnt}|%"  
9664     }%"  
9665     }% no hash
```

Place an open comment tag. This will hide any traces of the lateximage PDF page which were picked up by *pdftotext*.

```
9666     \LWR@traceinfo{lateximage: about to create open comment}%"  
9667     \LWR@htmlopencomment%"
```

One level deeper. At this outer-most `lateximage`, it is known that this is not being used inside an \mathcal{AM} S `\text`, since the outer-most level will never be in math mode.

```
9668 \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
9669 \LWR@traceinfo{lateximage: about to create a new page}%
9670 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
9671 \LWR@traceinfo{lateximage: about to create minipage}%
9672 \LWR@print@minipage{6in}%
9673 \nameuse{\LWR@print@\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for `\hspace`, etc. inside a `lateximage`.

```
9674 \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
9675 \LWR@restoreorigformatting%
```

Use full-page footnotes instead of `minipage` footnotes. These become HTML footnotes.

```
9676 \def\@mpfn{footnote}%
9677 \def\thempfn{\thefootnote}%
9678 \LetLtxMacro\@footnotetext{\LWR@footnotetext}%
```

Create the `LWR@lateximage<number>` label:

```
9679 \LWR@traceinfo{lateximage: about to create label}%
9680 \LWR@orig@label{\LWR@lateximage\arabic{LWR@lateximagenumber}}%
9681 \LWR@traceinfo{lateximage: finished creating the label}%
```

Enable print-mode math functions:

```
9682 \LetLtxMacro$\LWR@origdollar%
9683 \catcode`\$=3% math shift
9684 \LetLtxMacro\(\LWR@origopenparen%
9685 \LetLtxMacro\)\LWR@origcloseparen%
```

Only enable print-mode display math if are not already inside display math:

```
9686 \ifbool{LWR@indisplaymathimage}{}{\% not in display math
9687 \LetLtxMacro\[ \LWR@origopenbracket%
9688 \LetLtxMacro\] \LWR@origclosebracket%
```

```

9689      \let\equation\LWR@origequation%
9690      \let\endequation\LWR@origendequation%
9691      \csletcs{equation*}{\LWR@origequation*}%
9692      \csletcs{endequation*}{\LWR@origendequation*}%
9693  }% not in display math

```

For chemformula:

```

9694  \LetLtxMacro{\LWR@newsingledollar$}{%
9695  \LetLtxMacro{\LWR@newsingledollar$}{syntax highlighting}

9696 }% end of outer-most lateximage
9697 \LWR@traceinfo{lateximage: finished start of environment}%
9698 }% end of \begin{lateximage}

```

\endlateximage When the environment closes:

```

9699 {%
  start of \end{lateximage}
9700 \LWR@traceinfo{lateximage: starting end of lateximage}%

```

Nested more than one deep?

```

9701 \LWR@traceinfo{lateximage: internal depth was \arabic{LWR@lateximagedepth}}%
9702 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{1}%

```

If nesting inside an already existing lateximage, simply record one less level. Uses a lower-level TeX macro due to *AMS* \text change of \addtocounter.

```

9703 {%
9704  \LWR@traceinfo{lateximage: unnesting}%
9705  \global\advance\c@LWR@lateximagedepth -1\relax%
9706 }%

```

If this is the outer-most lateximage:

```
9707 {%
  end of outer-most lateximage
}
```

Finish the lateximage minipage and start a new PDF page:

```

9708 \LWR@traceinfo{lateximage: ending outer-most lateximage}%
9709  \endLWR@print@minipage%
9710  \LWR@orignewpage%
9711  \LWR@print@scriptsize%

```

Close the HTML comment which encapsulated any traces of the lateximage picked up by *pdftotext*:

```

9712     \LWR@print@vspace*{.5\baselineskip}%
9713     \LWR@htmlclosecomment%
9714     \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}%

```

Create a link to the lateximage, allowing its natural height:

```

9715     \IfBooleanTF{#1}{ starred
9716     {%
9717         \LWR@subinlineimage[#2]{lateximage}%
9718         {%
9719             lateximages\OSPathSymbol%
9720             \LWR@print@mbox{\LWR@hashedname}%
9721             }{svg}{#4}%
9722     }%
9723     {%
9724         \LWR@subinlineimage[#2]{lateximage}%
9725     }%
9726         lateximages\OSPathSymbol%
9727         \LWR@print@mbox{lateximage-\theLWR@externalfilecnt}%
9728         }{svg}{#4}%
9729     }%
9729     }% no hash

```

Be sure that are doing a paragraph:

```
9730     \LWR@ensuredoingapar%
```

Close the HTML span which has the *pdftotext* comment and also the link to the .svg image:

```

9731     \LWR@htmlltag{/span}%
9732     \ifbool{HTMLDebugComments}{%
9733         \LWR@htmlcomment{End of lateximage}%
9734     }{}%

```

Undo one lateximage level. This is not inside an *AMS* \text, so regular \addtocounter may be used here.

```

9735     \addtocounter{\LWR@lateximagedepth}{-1}%
9736 }% end of outer-most lateximage
9737 \LWR@traceinfo{lateximage: exiting depth is \arabic{\LWR@lateximagedepth}}%
9738 \LWR@traceinfo{lateximage: done}%
9739 }%
9740 \catcode`\$=3% math shift
9741 \end{warpHTML}

```

for PRINT output: 9742 \begin{warpprint}

Env lateximage [<alt> tag] [<css style>]

`varwidth` is used to create a box of the natural width of its contents.

```
9743 \NewDocumentEnvironment{lateximage}{s o o}
9744   {\begin{varwidth}[b]{\linewidth}}
9745   {\end{varwidth}}
9746 \end{warpprint}
```

80 center, flushleft, flushright

for HTML output: 9747 `\begin{warpHTML}`

Env `center` Replace `center` functionality with css tags:

```
9748 \renewenvironment*{center}
9749 {
9750 \LWR@forcenewpage
9751 \ifbool{FormatWP}
9752 {\BlockClass[\LWR@print@mbox{text-align:center}]{center}}
9753 {\BlockClass{center}}
9754 }
9755 {\endBlockClass}
```

Env `flushright`

```
9756 \renewenvironment*{flushright}
9757 {
9758 \LWR@forcenewpage
9759 \ifbool{FormatWP}
9760 {\BlockClass[\LWR@print@mbox{text-align:right}]{flushright}}
9761 {\BlockClass{flushright}}
9762 }
9763 {\endBlockClass}
```

Env `flushleft`

```
9764 \renewenvironment*{flushleft}
9765 {
9766 \LWR@forcenewpage
9767 \ifbool{FormatWP}
9768 {\BlockClass[\LWR@print@mbox{text-align:left}]{flushleft}}
9769 {\BlockClass{flushleft}}
9770 }
9771 {\endBlockClass}
```

\centering, \raggedleft, and \raggedright usually have no effect on the HTML output, but they may be used to compare with the next token to identify their use at the start of a float. See \LWR@floatalignment.

\centering

```
9772 \newcommand*\{LWR@HTML@centering}\{%
9773 \ifbool{HTMLDebugComments}\{%
9774     \LWR@htmlcomment{centering}\}%
9775 }\}%
9776 }%
9777 \LWR@formatted{centering}
```

\raggedleft

```
9778 \newcommand*\{LWR@HTML@raggedleft}\{%
9779 \ifbool{HTMLDebugComments}\{%
9780     \LWR@htmlcomment{raggedleft}\}%
9781 }\}%
9782 }%
9783 \LWR@formatted{raggedleft}
```

\raggedright

```
9784 \newcommand*\{LWR@HTML@raggedright}\{%
9785 \ifbool{HTMLDebugComments}\{%
9786     \LWR@htmlcomment{raggedright}\}%
9787 }\}%
9788 }%
9789 \LWR@formatted{raggedright}
```

\leftline {\langle text\rangle}

```
9790 \renewcommand{\leftline}[1]{\begin{flushleft}\#1\end{flushleft}}
```

\centerline {\langle text\rangle}

```
9791 \renewcommand{\centerline}[1]{\begin{center}\#1\end{center}}
```

\rightline {\langle text\rangle}

```
9792 \renewcommand{\rightline}[1]{\begin{flushright}\#1\end{flushright}}
```

```
9793 \end{warpHTML}
```

81 Preloaded packages

for HTML output: 9794 \begin{warpHTML}

If the given package was loaded before or by lwarp, load the lwarp version as well.

\LWR@PreloadedPackage {<packagename>}

```

9795 \newcommand*{\LWR@PreloadedPackage}[1]{%
9796     \@ifpackage{#1}{%
9797         {%
9798             \AtBeginDocument{%
9799                 \LWR@origRequirePackage{lwarp-#1}{%
9800             }{%
9801             }{%
9802             }{%
9803         }{%

```

If textcomp was loaded before lwarp, perhaps as part of the font-related packages, explicitly load the lwarp patches now:

9804 \LWR@PreloadedPackage{textcomp}

If xunicode was loaded before lwarp, perhaps as part of the font-related packages, explicitly load the lwarp patches now:

9805 \LWR@PreloadedPackage{xunicode}

If graphics or graphicx were loaded before lwarp, perhaps by xunicode, explicitly load the lwarp patches now:

```

9806 \LWR@PreloadedPackage{graphics}
9807 \LWR@PreloadedPackage{graphicx}
```

ulem may be preloaded by ctex, ctexart, and related classes.

9808 \LWR@PreloadedPackage{ulem}

xetexko-vertical may be preloaded by xetexko.

9809 \LWR@PreloadedPackage{xetexko-vertical}

geometry is preloaded by lwarp, and perhaps by various classes.

9810 \LWR@PreloadedPackage{geometry}

plex is preloaded by some CJK classes.

```
9811 \LWR@PreloadedPackage{plex}
```

stfloats is preloaded by \textit{ltj}^* classes.

```
9812 \LWR@PreloadedPackage{stfloats}
```

\textit{ltx} is preloaded by \textit{ltj}^* classes.

```
9813 \LWR@PreloadedPackage{ltx}
```

```
9814 \end{warpHTML}
```

82 siunitx

Pkg **siunitx** The lwarf core passes a few options to siunitx.

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

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

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

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

for HTML output: 9815 \begin{warpHTML}

Options for siunitx:

```
9816 \newrobustcmd{\LWR@siunitx@textcelsius}{\HTMLentity{deg}C}
9817 \newrobustcmd{\LWR@siunitx@textdegree}{\HTMLentity{deg}}
9818 \newrobustcmd{\LWR@siunitx@textprime}{\HTMLunicode{2032}}
9819 \newrobustcmd{\LWR@siunitx@textdblprime}{\HTMLunicode{2033}}
9820 \newrobustcmd{\LWR@siunitx@textplanckbar}{\text{\textit{\HTMLunicode{0127}}}}
9821
9822 \appto{\LWR@restoreorigformatting}{%
9823 \renewrobustcmd{\LWR@siunitx@textcelsius}{\text{\ensuremath{\textit{\circ}C}}}\%
9824 \renewrobustcmd{\LWR@siunitx@textdegree}{\text{\ensuremath{\textit{\circ}}} }\%
9825 \renewrobustcmd{\LWR@siunitx@textprime}{\text{\ensuremath{\textit{\prime}}}}\%
```

```

9826 \renewrobustcmd{\LWR@siunitx@textdblprime}{\text{\ensuremath{^{\prime\prime}}}}%
9827 \renewrobustcmd{\LWR@siunitx@textplanckbar}{\text{\ensuremath{\hbar}}}}%
9828 }
9829
9830 \PassOptionsToPackage{
9831   detect-mode=true,
9832   per-mode=symbol,% fraction is not seen by pdftotext
9833   text-celsius = {\LWR@siunitx@textcelsius},
9834   text-degree = {\LWR@siunitx@textdegree},
9835   text-arcminute = {\LWR@siunitx@textprime} ,
9836   text-arcsecond = {\LWR@siunitx@textdblprime} ,
9837 }{siunitx}

9838 \end{warpHTML}

```

83 Graphics print-mode modifications

83.1 General limitations

\includegraphics file formats For `\includegraphics` with .pdf or .eps files, the user must provide a .pdf or .eps image file for use in print mode, and also a .svg, .png, or .jpg version of the same image for use in HTML.

```
\includegraphics{filename} % print:.pdf/.eps HTML:.svg, etc.
```

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

If a .pdf or .eps image is referred to with its file extension, the extension will be changed to .svg for HTML:

```
\includegraphics{filename.pdf} % uses .svg in HTML
\includegraphics{filename.eps} % uses .svg in HTML
```

Prog `pdftocairo` To convert a PDF image to SVG, use the utility *pdftocairo*:

PDF to SVG

Enter ⇒ `pdftocairo -svg filename.pdf`

Prog `lwarpmk pdftosvg` For a large number of images, use *lwarpmk*:

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

Prog `lwarpmk epstopdf` For EPS images converted to PDF using the package *epstopdf*, use

Prog `epstopdf` Enter ⇒ `lwarpmk pdftosvg *.PDF`
epstopdf package

to convert to SVG images.

DVI latex When using DVI *latex*, it is necessary to convert EPS to PDF and then to SVG:

```
Enter ⇒ l warpmk epstopdf *.eps      (or a list of filenames)  
Enter ⇒ l warpmk pdftosvg *.pdf       (or a list of filenames)
```

PNG and JPG For PNG or JPG while using *pdflatex*, *lualatex*, or *xelatex*, the same file may be used in both print or HTML versions, and may be used with a file extension, but will also be used without the file extension if it is the only file of its base name.

GIF GIF files may be used for HTML, but another format must also be provided for print output.

file extension priorities If a file extension is not used, for HTML the file extension priorities are: SVG, GIF, PNG, then JPG.

⚠ 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 *l warp* — the entire image will be shown.

⚠ viewport

⚠ viewport units For `\includegraphics`, avoid px and % units for width and height, or enclose them inside *warpHTML* environments. For font-proportional image sizes, use ex or em. For fixed-sized images, use cm, mm, in, pt, or pc. Use the keys `width=.5\linewidth`, or similar for `\textwidth` or `\textheight` to give fixed-sized images proportional to a 6 by 9 inch text area. Do not use the `scale` option, since it is not well supported by HTML browsers.

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

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.

HTML alt tags Likewise, the `\includegraphics alt` key adds an HTML alt tag to an image, and is ignored for print output. If not assigned, each image is given an alt tag of “(image)”.

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

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

83.2 Print-mode modifications

for PRINT output: For print output, accept and then discard the new class key:

```
9839 \begin{warpprint}
9840 \define@key{Gin}{class}{}{}
9841 \define@key{Gin}{alt}{}{}
```

Print-mode additions for the overpic package. See section 315 for the HTML version.

```
9842 \AtBeginDocument{
9843 \@ifpackageloaded{overpic}{
9844 \newcommand*\overpicfontsize{12}
9845 \newcommand*\overpicfontskip{14}
9846 }{}}
9847 }
9848 \end{warpprint}
```

84 xcolor boxes

Pkg xcolor A few new definitions are provided for enhanced HTML colored boxes, and \fcolorbox is slightly modified. Print-mode version are also provided.

Print-mode versions of new xcolor defintions. These are defined inside warpall because they are also used for HTML while inside a `\teximage`. They are defined `\AtBeginDocument` so that the xcolor originals may first be loaded and saved for reuse.

The framed versions are modified to allow a background color of `none`, in which case only the frame is drawn, allowing the background page color to show.

for HTML & PRINT: 9849 \begin{warpall}

After `xparse` may have been loaded ...

```
9850 \AtBeginDocument{
```

... and *only* if xcolor was loaded:

```
9851 \@ifpackageloaded{xcolor}{}{
9852 \LWR@traceinfo{patching xcolor}}
```

The print version:

\colorboxBlock \colorboxBlock is the same as \colorbox:

```
9853 \LetLtxMacro\colorboxBlock\colorbox
```

The original definition is reused by the new versions:

```
9854 \LetLtxMacro\LWR@orig@print@fcolorbox\fcolorbox
```

```
\fcolorbox [⟨framemode⟩] {⟨framecolor⟩} [⟨boxmodel⟩] {⟨boxcolor⟩} {⟨text⟩}
```

In print mode, \fcolorbox is modified to accept a background color of none.

(\fcolorbox is particular about its optional arguments, thus the elaborate combinations of \ifthenelse.)

```
9855 \newsavebox{\LWR@colorminipagebox}
9856
9857 \NewDocumentCommand{\LWR@print@fcolorbox}{o m o m +m}{%
9858 \LWR@traceinfo{\LWR@print@fcolorbox #2 #4}}%
```

Pre-load the contents into an LR box so that they can be used inside a \fcolorbox:

```
9859 \begin{lrbox}{\LWR@colorminipagebox}%
9860 #5%
9861 \end{lrbox}%
```

Sort out the various optional arguments and the background color of none. In each case, the LRbox is placed inside a \fcolorbox.

The current color is remembered, then set to the frame, then the current color is used for the contents.

```
9862 \ifstreq{\#4}{none}%
9863 {%
9864     \LWR@traceinfo{background is none}%
9865     {%
9866         \colorlet{\LWR@currentcolor}{.}%
9867         \color{\#2}%
9868         \fbox{%
9869             \color{\LWR@currentcolor}%
9870             \usebox{\LWR@colorminipagebox}%
9871         }% fbox
9872     }% colorlet
9873 }% #4 none
9874 {%
9875 \LWR@traceinfo{background not none}%
9876 \IfValueTF{\#1}%
9877 {%
```

```

9878     \IfValueTF{#3}%
9879     {\LWR@orig@print@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}{%
9880     {\LWR@orig@print@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}}%
9881 }%
9882 {%
9883     \IfValueTF{#3}%
9884     {\LWR@orig@print@fcolorbox[#2]{#3}{#4}{\usebox{\LWR@colorminipagebox}}}{%
9885     {\LWR@orig@print@fcolorbox[#2]{#4}{\usebox{\LWR@colorminipagebox}}}}%
9886 }%
9887 }%
9888 \LWR@traceinfo{LWR@print@fcolorbox done}%
9889 }%
9890
9891 \renewcommand*{\fcolorbox}{\LWR@print@fcolorbox}

```

\fcolorboxBlock [⟨framemode⟩] {⟨framecolor⟩} [⟨boxmodel⟩] {⟨boxcolor⟩} {⟨text⟩}

In print mode, \fcolorboxBlock is the same as \fcolorbox.

```

9892 \newcommand*{\LWR@print@fcolorboxBlock}{\LWR@print@fcolorbox}%
9893 \newcommand*{\fcolorboxBlock}{\LWR@print@fcolorboxBlock}

```

Env fcolorminipage [⟨1:framemode⟩] {⟨2:framecolor⟩} [⟨3:boxmodel⟩] {⟨4:boxcolor⟩} [⟨5:align⟩] [⟨6:height⟩] [⟨7:inner-align⟩] {⟨8:width⟩}

In print mode, becomes a \fcolorbox containing a minipage:

```

9894 \NewDocumentEnvironment{LWR@print@fcolorminipage}{o m o m O{} o m}%
9895 {%
9896 \LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%

```

Pre-load the contents into an LR box so that they can be used inside a \fcolorbox:

```
9897 \begin{lrbox}{\LWR@colorminipagebox}%
```

If inner alignment is not given, use the outer alignment instead:

```

9898 \IfValueTF{#7}%
9899 {\begin{minipage}[#5][#6][#7]{#8}}%
9900 {\begin{minipage}[#5][#6][#5]{#8}}%
9901 }%
9902 {%
9903 \end{minipage}%
9904 \end{lrbox}%
9905 \LWR@traceinfo{*** starting end fcolorminipage #1 #2 #3 #4 #8}%

```

Sort out the various optional arguments and the background color of none. In each case, the LRbox is placed inside a \fcolorbox.

The current color is remembered, then set to the frame, then the current color is used for the contents.

```

9906 \ifstreq{#4}{none}%
9907 {%
9908     {%
9909         \colorlet{LWR@currentcolor}{.}%
9910         \color{#2}%
9911         \fbox{%
9912             \color{LWR@currentcolor}%
9913             \usebox{\LWR@colorminipagebox}%
9914         }%
9915     }%
9916 }%
9917 {%
9918     \IfValueTF{#1}%
9919     {%
9920         \IfValueTF{#3}%
9921         {\LWR@orig@print@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
9922         {\LWR@orig@print@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
9923     }%
9924     {%
9925         \IfValueTF{#1}%
9926         {\LWR@orig@print@fcolorbox[#2]{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
9927         {\LWR@orig@print@fcolorbox[#2]{#4}{\usebox{\LWR@colorminipagebox}}}%
9928     }%
9929 }%
9930 \LWR@traceinfo{*** finished end fcclorminipage}%
9931 }%
9932 %
9933 \newenvironment*{fcclorminipage}%
9934     {\LWR@print@fcclorminipage}%
9935     {\endLWR@print@fcclorminipage}%
9936 \LWR@traceinfo{xcolor patches done}%
9937 }{}%
9938 }%
9939 \end{warpall}

```

85 chemmacros environments

\makepolymerdelims and redox reactions must be enclosed in a `latexitimage` during HTML output. These environments are provided here in print mode, and in the chemmacros code in HTML mode, as a high-level semantic syntax which automatically embeds the contents in a `latexitimage` with an appropriate alt tag.

```

for PRINT output: 9940 \begin{warpprint}

9941 \AtBeginDocument{
9942 \@ifpackageloaded{chemmacros}{

Env  polymerdelims
9943 \DeclareDocumentEnvironment{polymerdelims}{}{%
9944     {}{}}

Env  redoxreaction  {\langle space above\rangle} {\langle space below\rangle}
For print output, extra space is include above and below the image, and a \textrimage
is not necessary. This extra space must be enforced, even inside a float, so zero-width
rules are used.
For the HTML version, see section 159.4.
9945 \DeclareDocumentEnvironment{redoxreaction}{m m}{%
9946     {\rule{0pt}{#1}}{\rule[-#2]{0pt}{#2}}{%
9947 }{\}}% chemmacros
9948 }% AtBeginDocument

9949 \end{warpprint}

```

86 cleveref

Pkg `cleveref` cleveref package is used as-is with minor patches.

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

Ex:

```

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

```

See `\cpagerefFor` at page [576](#) to redefine the message which is printed for page number references.

loading order cleveref and the following associated macro patches are automatically preloaded at the end of the preamble via `\AtEndPreamble` and `\AfterEndPreamble`. This is done because the HTML conversion requires cleveref. The user’s document may not require

cleveref, thus the user may never explicitly load it, so during HTML output lwarp loads it last. If the user's document preamble uses cleveref options, or functions such as \crefname, then cleveref may be loaded in the user's preamble near the end, and lwarp's additional loading of cleveref will have no effect.

Table 12 on page 491 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: 9950 \begin{warpHTML}

\AtEndPreamble forces cleveref to be loaded last:

```
9951 \AtEndPreamble{
9952 \RequirePackage{cleveref}
9953 }
```

The following patches are applied after cleveref has loaded, and after \AtBeginDocument. Print-mode versions are not required since they all come down to \ref eventually, and \ref has a print-mode version.

```
9954 \AfterEndPreamble{
9955 \LWR@traceinfo{Patching cleveref.}
```

\@setceref {<kindofref>} {<label>}

\templabel becomes the section number.

```
9956 \def\LWR@orig@@@setceref#1#2{\cref@getlabel{#2}{\templabel}#1{\templabel}{}%  
9957  
9958 \ifdefequal{\@setceref}{\LWR@orig@@@setceref}{% before v0.21  
9959     \renewcommand*{\@setceref}[2]{#1{\ref{#2}}{}{}}  
9960 }{  
9961     \ifdefequal{\@setceref}{\LWR@orig@@@setceref}{% as of v0.21  
9962         \renewcommand*{\@setceref}[2]{#1{\ref{#2}}{}{}}  
9963     }{  
9964         \PackageWarning{lwarp-cleveref}{  
9965             Unknown version of cleveref.  
9966             \protect\cref\space will fail.  
9967         }%  
9968     }  
9969 }
```

\@setcerefrange {<text>} {<label>} {<label>}

```

9970 \def\LWR@orig@@@setcrefrange#1#2#3{%
9971   \cref@getlabel{#2}{\@labela}%
9972   \cref@getlabel{#3}{\@labelb}%
9973   #1{\@labela}{\@labelb}{\{}{\}{}{\}}%
9974
9975 \ifdefequal{\@@@setcrefrange}{\LWR@orig@@@setcrefrange}{%
9976   \renewcommand{\@@@setcrefrange}[3]{%
9977     #1{\ref{#2}}{\ref{#3}}{\{}{\}{}{\}}%
9978   }%
9979 }{%
9980   \ifdefequal{\@@@setcrefrange}{\LWR@orig@@@setcrefrange}{%
9981     \renewcommand{\@@@setcrefrange}[3]{%
9982       #1{\ref{#2}}{\ref{#3}}{\{}{\}{}{\}}%
9983     }%
9984   }{%
9985     \PackageWarning{lwarp-cleveref}{%
9986       Unknown version of cleveref.%
9987       \protect\crefrange\space will fail.%
9988     }%
9989   }%
9990 }%
9991

```

\cpagerefFor Redefinable word between “page(s)” and the page numbers.

```
9992 \newcommand*{\cpagerefFor}{for}
```

\@@@setcpageref {\langle typeofref \rangle} {\langle label \rangle}, where typeofref is “page” or “pages”

```

9993 \def\LWR@orig@@@setcpageref#1#2{%
9994   \cref@getpageref{#2}{\@temppage}#1{\@temppage}{\{}{\}}%
9995
9996 \def\LWR@orig@@@setcpageref#1#2{%
9997   \cpageref@getlabel{#2}{\@temppage}#1{\@temppage}{\{}{\}}%
9998
9999 \ifdefequal{\@@@setcpageref}{\LWR@orig@@@setcpageref}{%
10000   \renewcommand*{\@@@setcpageref}[2]{%
10001     #1{\cpagerefFor\ \cref{#2}}{\{}{\}}%
10002   }%
10003 }{%
10004   \ifdefequal{\@@@setcpageref}{\LWR@orig@@@setcpageref}{%
10005     \renewcommand*{\@@@setcpageref}[2]{%
10006       #1{\cpagerefFor\ \cref{#2}}{\{}{\}}%
10007     }%
10008   }%
10009   {%
10010     \PackageWarning{lwarp-cleveref}{%

```

```

10011           Unknown version of cleveref.
10012           \protect\cpageref\space will fail.
10013       }
10014   }
10015 }

10016 \def\LWR@orig@@setcpagerefrange#1#2#3{%
10017   \cref@getpageref{#2}{\@pagea}%
10018   \cref@getpageref{#3}{\@pageb}%
10019   #1{\@pagea}{\@pageb}{\{}{\}{}{\}}%
10020
10021 \def\LWR@orig@@@setcpagerefrange#1#2#3{%
10022   \cpageref@getlabel{#2}{\@pagea}%
10023   \cpageref@getlabel{#3}{\@pageb}%
10024   #1{\@pagea}{\@pageb}{\{}{\}{}{\}}%
10025
10026 \ifdefined{\@setcpagerefrange}{\LWR@orig@@setcpagerefrange}{%
10027   \renewcommand*{\@setcpagerefrange}[3]{%
10028     #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{\{}{\}{}{\}}%
10029   }
10030 }{%
10031   \ifdefined{\@@setcpagerefrange}{\LWR@orig@@@setcpagerefrange}{%
10032     \renewcommand*{\@@setcpagerefrange}[3]{%
10033       #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{\{}{\}{}{\}}%
10034     }
10035   }
10036   {
10037     \PackageWarning{lwarp-cleveref}{%
10038       Unknown version of cleveref.
10039       \protect\cpagerefrange\space will fail.
10040     }
10041   }
10042 }
10043
10044 }% AfterEndPreamble

```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

\label and \pageref do NOT change their behavior according to print or HTML output, and thus do not use the \LWR@formatted system.

```

10045 \LetLtxMacro{\LWR@orig@label}{\label}
10046 \RenewDocumentCommand{\label}{\{}{\LWR@new@label\}}
10047
10048 \LetLtxMacro{\LWR@orig@pageref}{\pageref}
10049 \RenewDocumentCommand{\pageref}{\{}{\LWR@new@pageref\}}
10050 \end{warpHTML}

```

87 picture environment

`Env picture` The picture environment is enclosed inside a `\latextimage`.

for HTML output: 10051 `\begin{warpHTML}`

```
Env picture
10052 \BeforeBeginEnvironment{picture}{\begin{latextimage}[(picture)]}
10053
10054 \AfterEndEnvironment{picture}{\end{latextimage}}
10055 \end{warpHTML}
```

88 Boxes and Minipages

A css flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

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

 **text alignment** Nested minipages adopt their parent's text alignment in HTML, whereas in regular L^AT_EX PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

for HTML output: 10056 `\begin{warpHTML}`

88.1 Counters and lengths

Ctrl `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
10057 \newcounter{LWR@minipagedepth}
10058 \setcounter{LWR@minipagedepth}{0}
```

Len `\LWR@minipagewidth` Used to convert the width into printable units.

```
10059 \newlength{\LWR@minipagewidth}
```

Len `\LWR@minipageheight` Used to convert the height into printable units.

```
10060 \newlength{\LWR@minipageheight}
```

88.2 Footnote handling

Also see section 58 for other forms of footnotes. Minipage footnotes are gathered in section 58.5, and then placed into the document in section 88.3.

88.3 Minipage handling

Bool `LWR@minipagefullwidth` Should the next minipage have no HTML width?

```
10061 \newbool{LWR@minipagefullwidth}
10062 \boolfalse{LWR@minipagefullwidth}
```

`\minipagefullwidth` Requests that the next minipage have no width tag in HTML:

for HTML output: 10063 `\newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}`
 10064 `\end{warpHTML}`

for PRINT output: 10065 `\begin{warpprint}`
 10066 `\newcommand*{\minipagefullwidth}{}{}`
 10067 `\end{warpprint}`

for HTML output: 10068 \begin{warpHTML}

Bool LWR@minipagethispar Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

10069 \newbool{LWR@minipagethispar}
10070 \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 6x9 inch text area.

10071 \NewDocumentEnvironment{LWR@HTML@sub@minipage}{m m m m}
10072 {
10073 \LWR@traceinfo{minipage}%

Temporarily open a group, in which width and height is computed based on a virtual page size instead of the extra-large PDF page used during HTML tag generation.

The following used to be an actual L^AT_EX minipage.

10074 \begingroup

Compute width, adjusted for frames:

10075 \setlength{\LWR@minipagewidth}{#4}%
10076 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%

Only create a new page if not yet nested:

10077 \LWR@orignewpage%

Adjust virtual page size:

10078 \addtolength{\LWR@minipagewidth}{3em}%
 room for frames
 10079 \setlength{\linewidth}{6in}%
 10080 \setlength{\textwidth}{6in}%
 10081 \setlength{\textheight}{9in}%
 10082 }{}%
 10083 \LWR@traceinfo{computed width is \LWR@printlength{\LWR@minipagewidth}}%

Compute height:

10084 \setlength{\LWR@minipageheight}{\textheight}%
 default unless specified
 10085 \ifblank{#2}{}{\setlength{\LWR@minipageheight}{#2}}%

Track nesting depth:

```
10086 \addtocounter{LWR@minipagedepth}{1}%
```

L^AT_EX wants to start a paragraph for the virtual minipage, then start a paragraph again for the contents of the minipage, so cancel the paragraph tag handling until the minipage has begun.

```
10087 \ifbool{FormatWP}{\newline}{}
10088 \LWR@stopars%
```

If FormatWP, add a text frame:

```
10089 \ifbool{FormatWP}{%
10090
10091 \addtocounter{LWR@thisautoidWP}{1}%
10092 \LWR@htmlltag{%
10093     div id="\LWR@print@mbox{autoidWP-\arabic{LWR@thisautoidWP}}" %
10094     class="wpminipage"%
10095 }%
10096
10097 }{}}
```

Create the <div> tag with optional alignment style:

```
10098 \LWR@traceinfo{minipage: creating div class}%
10099 \LWR@htmlltag{div class="minipage" style="%"
10100 \ifthenelse{\equal{#1}{t}}{\LWR@print@mbox{vertical-align:bottom} ; }{%
10101 \ifthenelse{\equal{#1}{c}}{\LWR@print@mbox{vertical-align:middle} ; }{%
10102 \ifthenelse{\equal{#1}{b}}{\LWR@print@mbox{vertical-align:top} ; }{%
10103 \ifthenelse{\equal{#3}{t}}{\LWR@print@mbox{justify-content:flex-start} ; }{%
10104 \ifthenelse{\equal{#3}{c}}{\LWR@print@mbox{justify-content:center} ; }{%
10105 \ifthenelse{\equal{#3}{b}}{\LWR@print@mbox{justify-content:flex-end} ; }{%
10106 \ifthenelse{\equal{#3}{s}}{\LWR@print@mbox{justify-content:space-between} ; }{}}
```

Print the width and optional height styles:

```
10107 \LWR@traceinfo{minipage: about to print the width of \LWR@printlength{\LWR@minipagewidth}}%
10108 \ifbool{LWR@minipagefullwidth}{%
10109 {\boolefalse{LWR@minipagefullwidth}}%
10110 {%
10111     \ifthenelse{\lengthtest{#4}=\linewidth}{%
10112         {}%
10113         {width:\LWR@printlength{\LWR@minipagewidth} ; }%
10114 }%
10115 \LWR@traceinfo{minipage: about to print the height}%
10116 \ifblank{#2}{%
10117     {height:\LWR@printlength{\LWR@minipageheight} ; }%
10118 }}%
```

Finish with an empty line to start the contents on a new line.

```
10118  
10119 % The preceding empty line is required.
```

Set the user-accessible line and text width and height values inside the virtual minipage. These do not affect the actual size of the PDF output, but are used by any reference to \linewidth, etc. inside the virtual minipage being created here.

```
10120 \setlength{\linewidth}{#4}%
10121 \setlength{\textwidth}{6in}%
10122 \setlength{\textheight}{9in}%
```

\raggedright cancels hyphenation, which will be done by HTML instead.

```
10123 \LWR@print@raggedright%
```

Set minipage footnotes:

```
10124 \def\@mpfn{mpfootnote}%
10125 \def\thempfn{\thempfootnote}\c@mpfootnote\z@%
10126 \let\@footnotetext\mpfootnotetext%
```

Resume paragraph tag handling for the contents of the minipage:

```
10127 \LWR@startpars%
10128 \ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%
10129
10130 === begin minipage ===
10131
10132 }{}%
10133 \LWR@traceinfo{minipage: finished starting the minipage}%
10134 }% finished \minipage
10135 {%
10136 \endminipage
```

Print pending minipage footnotes:

```
10136 \LWR@printpendingmpfootnotes%
```

End the environment with closing tag:

```
10137 \ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%
10138
10139 === end minipage ===
10140
10141 }{}%
10142 \LWR@stopars%
```

The following used to be an actual L^AT_EX minipage.

```
10143 \endgroup%
10144
10145 \ifbool{FormatWP}{%
10146
10147 \LWR@htmlelementend{div}%
10148
10149 }{}%
10150 \LWR@htmldivclassend{minipage}%
10151
10152 \addtocounter{LWR@minipagedepth}{-1}%
10153 \LWR@startpars%
10154 \ifbool{FormatWP}{\newline}{}
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
10155 \global\booltrue{LWR@minipagethispar}%
10156 \LWR@traceinfo{LWR@minipage: done}%
10157 }
10158
10159 \NewDocumentEnvironment{LWR@HTML@minipage}{O{t} O{} O{t} m}
10160     {\LWR@HTML@sub@minipage{#1}{#2}{#3}{#4}}
10161     {\endLWR@HTML@sub@minipage}
10162
10163 \LWR@formattedenv{minipage}
```

88.4 \parbox, \mbox, \makebox, \framebox, \fbox, \raisebox

for HTML output:

```
\parbox [⟨pos⟩] [⟨height⟩] [⟨inner-pos⟩] {⟨width⟩} {⟨text⟩}
```

A parbox uses the minipage code:

```
10164 \NewDocumentCommand{\LWR@HTML@parbox}{O{t} O{} O{t} m +m}
10165 {
10166 \LWR@traceinfo{parbox of width #4}%
10167 \begin{minipage}[#1][#2][#3]{#4}%
10168 #5
10169 \end{minipage}%
10170 }
10171
10172 \LWR@formattedenv{parbox}
```

```
\mbox {<text>} Nullified for HTML.

10173 \newcommand*{\LWR@HTML@mbox}[1]{#1}
10174
10175 \LWR@formatted{mbox}

\makebox (<pos>) [<width>] [<pos>] {<text>}
10176 \NewDocumentCommand{\LWR@HTML@makebox}{d() o o m}{%
```

Check for the optional width:

```
10177 \IfValueTF{#2}%
10178 {%
```

Check for the horizontal text alignment. For stretched, the best HTML can do is justified alignment.

```
10179      {%
10180          \def\LWR@align{center}%
10181          \ifstreq{\#3}{l}{\def\LWR@align{left}}{}%
10182          \ifstreq{\#3}{r}{\def\LWR@align{right}}{}%
10183          \ifstreq{\#3}{s}{\def\LWR@align{justify}}{}%
```

To print the width argument:

```
10184     \setlength{\LWR@tempwidth}{#2}%

```

inline-block allows width and text-align to be used in a .

```
10185     \InlineClass[%]
10186         \LWR@print@mbox{display:inline-block} ; %
10187         \LWR@print@mbox{text-align}:\LWR@align\ ; %
10188         width:\LWR@printlength{\LWR@tempwidth}%
10189     ]%
10190     {makebox}%
10191     {#4}%
10192 }% scope
10193 }%
```

Without a width argument, the text is simply used inline:

```
10194 {#4}%
10195 no width
10196 }
10197 \LWR@formatted{makebox}
```

```
\framebox  {[<width>]} [<pos>] {<text>}

10198 \LetLtxMacro{\LWR@origframebox}\framebox
10199
10200 \NewDocumentCommand{\LWR@HTML@framebox}{o o m}{%
10201 \fbox{\makebox[#1][#2]{#3}}%
10202 }
10203
10204 \LWR@formatted{framebox}
```

\LWR@forceminwidth {[<legth>]}

Sets \LWR@atleastonept to be at least 1pt.

```
10205 \newlength{\LWR@atleastonept}
10206
10207 \newcommand*{\LWR@forceminwidth}[1]{%
10208 \setlength{\LWR@atleastonept}{#1}%
10209 \ifthenelse{%
10210     \lengthtest{\LWR@atleastonept>0pt}\AND%
10211     \lengthtest{\LWR@atleastonept<1pt}%
10212 }%
10213 {\setlength{\LWR@atleastonept}{1pt}}%
10214 {}%
10215 }
```

\LWR@blackborderpadding Prints the HTML attributes for a black border and padding.

\LWR@forceminwidth must be used first in order to set the border width.

```
10216 \newcommand*{\LWR@blackborderpadding}{%
10217 border:\LWR@printlength{\LWR@atleastonept} solid black ; %
10218 padding:\LWR@printlength{\fboxsep}%
10219 }
```

\fbox {<text>}

Creates a framed inline span enclosing the text.

Create a new HTML version, but don't use it until after xcolor may have loaded:

```
10220 \newcommand{\LWR@HTML@fbox}[1]{%
10221 \LWR@traceinfo{HTML fbox}%
10222 \LWR@forceminwidth{\fboxrule}%
10223 \InlineClass[%]
10224 \LWR@blackborderpadding%
10225 ]{\fbox}{#1}
10226 }
```

xcolor \lets things to \fbox when it is loaded, and this must remain even for HTML output while in a `lateximage`, so \fbox is not modified until \AtBeginDocument:

```
10227 \AtBeginDocument{\LWR@formatted{fbox}}
```

`\fboxBlock {<text>}` Creates a framed HTML <div> of the text.

First, a print-mode version. This is newly defined for print mode, so it is defined inside `warpall`.

for HTML & PRINT:

```
10228 \end{warpHTML}
10229 \begin{warpall}
10230 \let\fboxBlock\fbox
10231 \end{warpall}
10232 \begin{warpHTML}
```

for HTML output: Next, an HTML version:

```
10233 \newcommand{\LWR@HTML@fboxBlock}[1]{%
10234 \LWR@forceminwidth{\fboxrule}%
10235 \begin{BlockClass}[%
10236 \LWR@blackborderpadding%
10237 ]\fboxBlock}
10238 #1
10239 \end{BlockClass}
10240 }
10241
10242 \LWR@formatted{fboxBlock}
10243
10244 \end{warpHTML}
```

`Env fminipage [<align>] [<height>] [<align>] {<width>}`

Creates a framed HTML <div> around its contents.

for HTML & PRINT: Print version:

```
10245 \begin{warpall}
10246
10247 \newsavebox{\LWR@fminipagebox}
10248
10249 \NewDocumentEnvironment{\LWR@print@fminipage}{O{t} o O{t} m}
10250 {%
```

An outer minipage will be used for vertical alignment. An inner minipage will be framed with \fbox.

If the optional inner alignment is not given, use the outer instead:

```

10251 \IfValueTF{#3}%
10252 {\def\LWR@thisalign{#3}}
10253 {\def\LWR@thisalign{#1}}%

```

Form the outer minipage depending on whether a height was given. Make the outer minipage larger to compensate for the frame.

```

10254 \IfValueTF{#2}%
10255 {\minipage[#1][#2+2\fboxsep+2\fboxrule][\LWR@thisalign]{#4+2\fboxsep+2\fboxrule}}%
10256 {\minipage[#1]{#4+2\fboxsep+2\fboxrule}}%

```

Capture the contents of the environment:

```
10257 \begin{lrbox}{\LWR@fminipagebox}%
```

Nest the contents inside an inner minipage of the desired size:

```

10258 \IfValueTF{#2}%
10259 {\minipage[#1][#2][\LWR@thisalign]{#4}}%
10260 {\minipage[#1]{#4}}%
10261 }%
10262 {%

```

Close the inner minipage and the LR box with the contents:

```

10263 \endminipage%
10264 \end{lrbox}%

```

Create a frame around the contents of the environment:

```
10265 \fbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```

10266 \endminipage%
10267 }%
10268
10269 \LetLtxMacro\fminipage\LWR@print\fminipage
10270 \LetLtxMacro\endfminipage\endLWR@print\fminipage
10271 % \newenvironment{fminipage}{\LWR@print\fminipage}{\endLWR@print\fminipage}
10272
10273 \end{warpall}%

```

for HTML output: 10274 \begin{fminipageHTML}

```

10275
10276 \NewDocumentEnvironment{\LWR@HTML\fminipage}{O{t} o O{t} m}
10277 {%
10278 \LWR@traceinfo{fminipage #1 #2 #3 #4}%

```

```

10279 \LWR@forceminwidth{\fboxrule}%
10280 \setlength{\LWR@tempwidth}{#4}%
10281 \IfValueT{#2}{\setlength{\LWR@tempheight}{#2}}%

10282 \LWR@stopars%

10283 \begin{BlockClass}[%  

10284 \LWR@blackborderpadding ; %  

10285 \IfValueT{#2}{height:\LWR@printlength{\LWR@tempheight} ; }%  

10286 width:\LWR@printlength{\LWR@tempwidth}%
10287 ]{fminipage}%
10288 }
10289 {%
10290 \end{BlockClass}%

```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```

10291 \global\booltrue{\LWR@minipagethispar}%
10292 \LWR@traceinfo{fminipage done}%
10293 }
10294
10295 \LWR@formattedenv{fminipage}

10296 \end{warpHTML}

\raisebox {<raiselen>} [<height>] [<depth>] {<text>}

10297 \begin{warpHTML}
10298
10299 \NewDocumentCommand{\LWR@HTML@raisebox}{m o o m}{%
10300 #4%
10301 }
10302
10303 \LWR@formatted{raisebox}

10304 \end{warpHTML}

```

89 Direct formatting

- ⚠ **\bfseries, etc.** \textbf{, etc. are supported, but \bfseries, etc. work only in some situations.}
- ⚠ **HTML special chars** &, <, and > have special meanings in HTML. If \&, \textless, and \textgreater are

used, proper HTML entities will be used, but there may be HTML parsing problems if these special characters occur unescaped in program listings or other verbatim text.

program listings For program listings, the listings package is supported, and its literate option is used to convert &, <, and > to proper HTML entities.

verbatim The various verbatim-related environments do not convert &, <, and >, so care must be taken to avoid accidentally including valid HTML code inside these environments. Adding a space on either side may be sufficient.

For high-level block and inline custom css classes, see section 52.8.

for HTML output: 10305 \begin{warpHTML}

```
\emph  {\langle text\rangle}

10306 \DeclareRobustCommand{\LWR@HTML@emph}[1]{%
10307   {%
10308     \itshape%
10309     \LWR@htmlspan{em}{\#1}%
10310   }%
10311 }
10312 \LWR@formatted{emph}
10313
10314 \DeclareRobustCommand{\LWR@null@emph}[1]{\#1}

\textmd  {\langle text\rangle}

10315 \DeclareRobustCommand{\LWR@HTML@textmd}[1]{%
10316   {%
10317     \mdseries%
10318     \InlineClass(font-weight:normal){textmd}{\#1}%
10319   }%
10320 }
10321 \LWR@formatted{textmd}
10322
10323 \DeclareRobustCommand{\LWR@null@textmd}[1]{\#1}

\textbf  {\langle text\rangle}

10324 \DeclareRobustCommand{\LWR@HTML@textbf}[1]{%
10325   {%
10326     \bfseries%
10327     \LWR@htmlspan{b}{\#1}%
10328   }%
10329 }
10330 \LWR@formatted{textbf}
```

```
10331
10332 \DeclareRobustCommand{\LWR@null@textbf}[1]{#1}

\textrm {⟨text⟩}

10333 \DeclareRobustCommand{\LWR@HTML@textrm}[1]{%
10334   {%
10335     \rmfamily%
10336     \InlineClass{font-family:serif}{textrm}{#1}%
10337   }%
10338 }
10339 \LWR@formatted{textrm}
10340
10341 \DeclareRobustCommand{\LWR@null@textrm}[1]{#1}

\textsf {⟨text⟩}

10342 \DeclareRobustCommand{\LWR@HTML@textsf}[1]{%
10343   {%
10344     \sffamily%
10345     \InlineClass{font-family:sans}{textsf}{#1}%
10346   }%
10347 }
10348 \LWR@formatted{textsf}
10349
10350 \DeclareRobustCommand{\LWR@null@textsf}[1]{#1}

\tt {⟨text⟩}

10351 \DeclareRobustCommand{\LWR@HTML@texttt}[1]{%
10352   {%
10353     \ttfamily%
10354     \LWR@htmlspan{kbd}{#1}%
10355   }%
10356 }
10357 \LWR@formatted{texttt}
10358
10359 \DeclareRobustCommand{\LWR@null@texttt}[1]{#1}

\textup {⟨text⟩}

10360 \DeclareRobustCommand{\LWR@HTML@textup}[1]{%
10361   {%
10362     \upshape%
10363     \InlineClass{font-style:normal}{textup}{#1}%
10364   }%
10365 }
```

```
10366 \LWR@formatted{textup}
10367
10368 \DeclareRobustCommand{\LWR@null@textup}[1]{#1}

\textit {⟨text⟩}

10369 \DeclareRobustCommand{\LWR@HTML@textit}[1]{%
10370   {%
10371     \itshape%
10372     \LWR@htmlspan{i}{#1}%
10373   }%
10374 }
10375 \LWR@formatted{textit}
10376
10377 \DeclareRobustCommand{\LWR@null@textit}[1]{#1}

\textsc {⟨text⟩}

10378 \DeclareRobustCommand{\LWR@HTML@textsc}[1]{%
10379   {%
10380     \scshape%
10381     \InlineClass(
10382       font-variant: small-caps ;
10383       font-variant-numeric: oldstyle-nums ;
10384     ){textsc}{#1}%
10385   }%
10386 }
10387 \LWR@formatted{textsc}
10388
10389 \DeclareRobustCommand{\LWR@null@textsc}[1]{#1}

\textsi {⟨text⟩}

10390 @ifundefined{textsi}{
10391   \LetLtxMacro{\LWR@print@textsi}{\LWR@print@textsc}
10392 }{%
10393
10394 \DeclareRobustCommand{\LWR@HTML@textsi}[1]{%
10395   {%
10396     \sishape%
10397     \InlineClass(
10398       font-style: italic;
10399       font-variant: small-caps ;
10400       font-variant-numeric: oldstyle-nums ;
10401     ){textsi}{#1}%
10402   }%
10403 }
10404 \LWR@formatted{textsi}
```

```

10405
10406 \DeclareRobustCommand{\LWR@null@textsi}[1]{#1}

\textsl  {\langle text\rangle}

10407 \DeclareRobustCommand{\LWR@HTML@textsl}[1]{%
10408   {%
10409     \slshape%
10410     \InlineClass(font-style:oblique){textsl}{#1}%
10411   }%
10412 }
10413 \LWR@formatted{textsl}
10414
10415 \DeclareRobustCommand{\LWR@null@textsl}[1]{#1}

\textnormal {\langle text\rangle}

10416 \DeclareRobustCommand{\LWR@HTML@textnormal}[1]{\textmd{\textrm{\textup{#1}}}}
10417 \LWR@formatted{textnormal}
10418
10419 \DeclareRobustCommand{\LWR@null@textnormal}[1]{#1}

10420 \DeclareRobustCommand{\LWR@null@rmfamily}{}
10421 \DeclareRobustCommand{\LWR@null@sffamily}{}
10422 \DeclareRobustCommand{\LWR@null@ttfamily}{}
10423 \DeclareRobustCommand{\LWR@null@bfseries}{}
10424 \DeclareRobustCommand{\LWR@null@mdseries}{}
10425 \DeclareRobustCommand{\LWR@null@upshape}{}
10426 \DeclareRobustCommand{\LWR@null@slshape}{}
10427 \DeclareRobustCommand{\LWR@null@scshape}{}
10428 \DeclareRobustCommand{\LWR@null@itshape}{}
10429 \DeclareRobustCommand{\LWR@null@normalfont}{}

10430 \DeclareRobustCommand{\LWR@null@em}{}

```

\LWR@nullfonts Removes formatting during filename operations.



Use only inside a group.

The following are *not* made robust, since they must be expanded to their nullified versions.

```

10431 \newcommand*{\LWR@nullfonts}{%
10432 \LetLtxMacro{\emph}{\LWR@null@emph}%
10433 \LetLtxMacro{\textmd}{\LWR@null@textmd}%
10434 \LetLtxMacro{\textbf}{\LWR@null@textbf}%

```

```
10435 \LetLtxMacro\textrm\LWR@null@textrm%
10436 \LetLtxMacro\textsf\LWR@null@textsf%
10437 \LetLtxMacro\texttt\LWR@null@texttt%
10438 \LetLtxMacro\textup\LWR@null@textup%
10439 \LetLtxMacro\textit\LWR@null@textit%
10440 \LetLtxMacro\textsc\LWR@null@textsc%
10441 \LetLtxMacro\textsi\LWR@null@textsi%
10442 \LetLtxMacro\textsl\LWR@null@textsl%
10443 \LetLtxMacro\textnormal\LWR@null@textnormal%
10444 \LetLtxMacro\rmfamily\LWR@null@rmfamily%
10445 \LetLtxMacro\sffamily\LWR@null@sffamily%
10446 \LetLtxMacro\ttfamily\LWR@null@ttfamily%
10447 \LetLtxMacro\bfseries\LWR@null@bfseries%
10448 \LetLtxMacro\mdseries\LWR@null@mdseries%
10449 \LetLtxMacro\upshape\LWR@null@upshape%
10450 \LetLtxMacro\slshape\LWR@null@slshape%
10451 \LetLtxMacro\scshape\LWR@null@scshape%

10452 \LetLtxMacro\sishape\LWR@null@sishape%

10453 \LetLtxMacro\itshape\LWR@null@itshape%
10454 \LetLtxMacro\normalfont\LWR@null@normalfont%
10455 \LetLtxMacro\em\LWR@null@em%

10456 \renewcommand*{\,}{-}%
10457 \renewcommand*{-}{-}%
10458 \renewcommand*{\newline}{ }%
10459 \renewcommand*{\textellipsis}{-}%

10460 \renewcommand*{\HTMLUnicode}[1]{}%
10461 \renewcommand*{\HTMLEntity}[1]{}%
```

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

```
10462 \renewcommand*{\&}{and}%

10463 \renewcommand{\textsuperscript}[1]{##1}%
10464 \renewcommand{\textsubscript}[1]{##1}%

10465 \renewcommand{\underline}[1]{##1}%

10466 \RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
10467 \DeclareExpandableDocumentCommand{\InlineClass}{D{(){}{}} o m +m}{##4}%

```

Nullify math macros.

```
10468 \def\(##1\){}%
10469 \def\[##1\]{}%
10470 \RenewDocumentCommand{\LWR@subsingleollar}{s m m m}{}%
```

Use the simpler form with `\texorpdfstring`:

```
10471 \let\texorpdfstring\relax%
10472 \newcommand{\texorpdfstring}[2]{##2}%
10473 }
```

Remembers the current font family, series, and shape.

```
10474 \newcommand*{\LWR@f@family}{rm}%
10475 \newcommand*{\LWR@f@series}{md}%
10476 \newcommand*{\LWR@f@shape}{up}
```

`\LWR@textcurrentfont {<text>}`

Prints the text with the current font choices. Avoids nesting repeated font selections.

```
10477 \newcounter{\LWR@textcurrentfontdepth}
10478 \setcounter{\LWR@textcurrentfontdepth}{0}
10479
10480 \newcommand*{\LWR@textcurrentfont}[1]{%
10481     \ifnumcomp{\value{\LWR@textcurrentfontdepth}}{>}{0}{%
10482         \%%
10483         \addtocounter{\LWR@textcurrentfontdepth}{1}%
10484         #1%
10485         \addtocounter{\LWR@textcurrentfontdepth}{-1}%
10486     }%
10487     \%%
10488     \addtocounter{\LWR@textcurrentfontdepth}{1}%
10489     \InlineClass{%
10490         text\LWR@f@family\LWR@origtilde{}%
10491         text\LWR@f@series\LWR@origtilde{}%
10492         text\LWR@f@shape%%
10493     }%
10494     {\#1}%
10495     \addtocounter{\LWR@textcurrentfontdepth}{-1}%
10496     \%%
10497 }
```

Env `\LWR@blocktextcurrentfont` Prints the contents with the current font choices.

```
10498 \newenvironment*{\LWR@blocktextcurrentfont}{%
```

```
10499 \BlockClass{%
10500     text\LWR@f@family\LWR@origtilde{}%
10501     text\LWR@f@series\LWR@origtilde{}%
10502     text\LWR@f@shape%
10503 }%
10504 }{\endBlockClass}

\mdseries
10505 \renewrobustcmd*{\mdseries}{\renewcommand*{\LWR@f@series}{md} }

\bfseries
10506 \renewrobustcmd*{\bfseries}{\renewcommand*{\LWR@f@series}{bf} }

\rmfamily
10507 \renewrobustcmd*{\rmfamily}{\renewcommand*{\LWR@f@family}{rm} }

\sffamily
10508 \renewrobustcmd*{\sffamily}{\renewcommand*{\LWR@f@family}{sf} }

\ttfamily
10509 \renewrobustcmd*{\ttfamily}{\renewcommand*{\LWR@f@family}{tt} }

\upshape
10510 \renewrobustcmd*{\upshape}{\renewcommand*{\LWR@f@shape}{up} }

\itshape
10511 \renewrobustcmd*{\itshape}{\renewcommand*{\LWR@f@shape}{it} }

\scshape
10512 \renewrobustcmd*{\scshape}{\renewcommand*{\LWR@f@shape}{sc} }

\sishape
10513 \@ifundefined{sishape}{
```

```
10514     \newrobustcmd{\sishape}{\renewcommand{\LWR@f@shape}{si}}
10515 }{
10516     \renewrobustcmd{\sishape}{\renewcommand{\LWR@f@shape}{si}}
10517 }

\sishape

10518 \renewrobustcmd{\slshape}{\renewcommand{\LWR@f@shape}{sl}}

\normalfont

10519 \renewrobustcmd{\normalfont}{\rmfamily\mdseries\upshape}

\sp {text}
For siunitx. Must work in math mode.

10520 \renewcommand{\sp}[1]{\text{\sup{#1}}}

\sb {text}
For siunitx. Must work in math mode.

10521 \renewcommand{\sb}[1]{\text{\sub{#1}}}

\textsuperscript {text}
10522 \renewrobustcmd{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}#1}

@\textsuperscript {text}
10523 \renewcommand{\@textsuperscript}[1]{\LWR@htmlspan{sup}{#1}#1}

\textsubscript {text}
10524 \AtBeginDocument{
10525 \renewrobustcmd{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}#1}
10526 }

@\textsubscript {text}
10527 \AtBeginDocument{
10528 \renewcommand{\@textsubscript}[1]{\LWR@htmlspan{sub}{#1}#1}
10529 }
```

\up {*text*} Prints superscript.

This is \let at the beginning of the document in case some other package has changed the definition.

```
10530 \AtBeginDocument{\let\up\textsuperscript}
```

\fup {*text*} Prints superscript.

Supports fmtcount package.

This is \let at the beginning of the document in case some other package has changed the definition.

```
10531 \AtBeginDocument{\let\fup\textsuperscript}
```

\underline {*text*}

```
10532 \renewcommand{\underline}[1]{%
10533   \InlineClass{%
10534     (text-decoration:underline; text-decoration-skip: auto)%
10535     {underline}{#1}%
10536 }
```

\LWR@overline {*text*}

```
10537 \newcommand{\LWR@overline}[1]{%
10538   \InlineClass{%
10539     (text-decoration:overline; text-decoration-skip: auto)%
10540     {overline}{#1}%
10541 }
```

\LWR@currenttextcolor The color to use for text and \rule, defaulting to black:

```
10542 \newcommand*{\LWR@currenttextcolor}{black}
```

\LWR@tempcolor The color converted to HTML colorspace.

```
\LWR@tempcolortwo
10543 \newcommand*{\LWR@tempcolor}{}%
10544 \newcommand*{\LWR@tempcolortwo}{}%
```

\LWR@findcurrenttextcolor Sets \LWR@tempcolor to the current color.

```
10545 \newcommand*{\LWR@findcurrenttextcolor}{}%
10546 \renewcommand{\LWR@tempcolor}{black}%
10547 }
```

\LWR@textcurrentcolor {<text>} Like \textcolor but uses the current \color instead.

```
10548 \NewDocumentCommand{\LWR@textcurrentcolor}{m}{%
1049     \renewcommand*{\LWR@currenttextcolor}{black}%
1050     #1%
1051 }

10552 \end{warpHTML}
```

for PRINT output: 10553 \begin{warpprint}

\LWR@textcurrentfont {<text>}

Prints the text with the current font choices.

```
10554 \newcommand*{\LWR@textcurrentfont}[1]{#1}
```

Env LWR@blocktextcurrentfont Prints the contents with the current font choices.

```
10555 \newenvironment*{LWR@blocktextcurrentfont}{}{}

10556 \end{warpprint}
```

90 Skips, spaces, font sizes

for HTML output: 10557 \begin{warpHTML}

\, and \thinspace may be redefined by other packages, so are redefined \AtBeginDocument here.

Direct-formatting space commands become HTML entities:

```
10558 \AtBeginDocument{
10559 \renewrobustcmd*{\,}{\HTMLunicode{202f}} % HTML thin non-breakable space
10560 \renewrobustcmd*{\thinspace}{\HTMLunicode{202f}} % HTML thin non-breakable space
10561 \renewrobustcmd*{\negthinspace}{\HTMLunicode{202f}} % HTML thin non-breakable space
10562 \renewrobustcmd*{\~}{\HTMLentity{nbsp}}
10563 \renewrobustcmd*{\textellipsis}{\HTMLunicode{2026}}
10564 }
```

Direct-formatting font sizes are ignored:

```
10565 \newrobustcmd*{\LWR@HTML@normalsize}{}{}
```

```
10566 \LWR@formatted{normalsize}
10567
10568 \newrobustcmd*{\LWR@HTML@small}{}
10569 \LWR@formatted{small}
10570
10571 \newrobustcmd*{\LWR@HTML@footnotesize}{}
10572 \LWR@formatted{footnotesize}
10573
10574 \newrobustcmd*{\LWR@HTML@scriptsize}{}
10575 \LWR@formatted{scriptsize}
10576
10577 \newrobustcmd*{\LWR@HTML@tiny}{}
10578 \LWR@formatted{tiny}
10579
10580 \newrobustcmd*{\LWR@HTML@large}{}
10581 \LWR@formatted{large}
10582
10583 \newrobustcmd*{\LWR@HTML@Large}{}
10584 \LWR@formatted{Large}
10585
10586 \newrobustcmd*{\LWR@HTML@LARGE}{}
10587 \LWR@formatted{LARGE}
10588
10589 \newrobustcmd*{\LWR@HTML@huge}{}
10590 \LWR@formatted{huge}
10591
10592 \newrobustcmd*{\LWR@HTML@Huge}{}
10593 \LWR@formatted{Huge}

10594 \DeclareDocumentCommand{\onecolumn}{}
10595
10596 \DeclareDocumentCommand{\twocolumn}{O{}}
10597
10598 #1
10599
10600 }

\hfill

10601 \newcommand*{\LWR@HTML@hfill}{\qquad}
10602 \LWR@formatted{hfill}

\hrulefill

10603 \newcommand*{\LWR@HTML@hrulefill}{\rule{1in}{1pt}}
10604 \LWR@formatted{hrulefill}
```

```
\dotfill
```

```
10605 \newcommand*{\LWR@HTML@dotfill}{\dots}
10606 \LWR@formatted{dotfill}
```

```
\newpage
```

```
10607 \renewcommand*{\newpage}{
10608
10609 }
```

`\newline` Uses the HTML `
` element.

```
10610 \newrobustcmd*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
10611 \LetLtxMacro{\newline}{\LWR@newlinebr}
```

`\\" Redefined to \LWR@endofline or \LWR@tabularendofline.`

```
\LWR@endofline * [<len>]
```

`\\" is assigned to \LWR@endofline at \LWR@LwarpStart.`

`Inside tabular, \\" is temporarily changed to \LWR@tabularendofline.`

```
10612 \LetLtxMacro{\LWR@origendofline}\
10613 \NewDocumentCommand{\LWR@endofline}{s O{0pt}}
10614 {%
10615 \newline%
```

```
10616 \setlength{\LWR@templengthone}{#2}%
10617 \ifdimgreater{\LWR@templengthone}{0pt}{\newline}{}%
10618 }
```

`\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 \quad's HTML output.

```
10619 \newcommand*{\LWR@minipagestartpars}{%
10620 \ifbool{LWR@minipagethispar}{\LWR@startpars}{()}%
10621 }
```

\LWR@minipagestopars Placed just after \hspace, \quad, or \quad's HTML output.

```
10622 \newcommand*{\LWR@minipagestopars}{%
10623 \ifbool{LWR@minipagethispar}{\LWR@stopars}{()}%
10624 }
```

\quad Handles special minipage & horizontal space interactions. Uses 2003 EM SPACE to pass validation.

```
10625 \renewrobustcmd*{\quad}{%
10626 \LWR@minipagestopars%
10627 \HTMLunicode{2003}%
10628 \LWR@minipagestartpars%
10629 }
```

\quad Handles special minipage & horizontal space interactions.

```
10630 \renewrobustcmd*{\quad}{\quad\quad}
```

\enskip Handles special minipage & horizontal space interactions.

```
10631 \renewrobustcmd*{\enskip}{%
10632 \LWR@minipagestopars%
10633 \HTMLunicode{2002}%
10634 \LWR@minipagestartpars%
10635 }
```

Len \LWR@tempwidth Used to compute span width, height, raise for \hspace and \rule:

```
Len \LWR@tempheight
Len \LWR@tempraise 10636 \newlength{\LWR@tempwidth}
10637 \newlength{\LWR@tempheight}
10638 \newlength{\LWR@tempraise}
```

```
\LWR@select@html@hspace * {<length>} * {<length>}
\hspace
```

Handles special minipage & horizontal space interactions.

Prints a span of a given width. Ignores the optional star.

\hspace{\fill} is converted to \hspace{2em}, equal to \qquad.

```
10639 \newcommand{\LWR@select@html@hspace}{%
10640 \RenewDocumentCommand{\hspace}{s m}{%
10641 \setlength{\LWR@tempwidth}{##2}}%
```

If \fill, change to \qquad:

```
10642 \ifnum\gluestretchorder\LWR@tempwidth>0%
10643 \setlength{\LWR@tempwidth}{2em}%
10644 \fi%
```

Only if the width is not zero:

```
10645 \ifdim\comp{\LWR@tempwidth}{=}{0pt}{}{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
10646 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
10647 \ifdim\comp{\LWR@tempwidth}{=}{.16667em}%
10648 {%
10649   \HTMLunicode{2009}% thin breakable space
10650 }%
```

Print the span with the converted width. Not rounded.

```
10651 {%
10652   \LWR@htmntagc{%
10653     span style="width:\LWR@printlength{\LWR@tempwidth}; % extra space
10654       display:inline-block"%
10655 }%
```

If formatting for a word processor, approximate with a number of \quads, in case a span of a given width is not supported:

```
10656 \ifbool{FormatWP}{%
10657   \setlength{\LWR@templengthone}{\LWR@tempwidth}%
10658   \whiledo{\lengthtest{\LWR@templengthone}>1em}{%
10659     \quad%
10660     \addtolength{\LWR@templengthone}{-1em}}%
```

```
10661      }%
10662  }{ }%
```

Close the span:

```
10663      \LWR@htmltagc{/span}%
10664  }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
10665      \LWR@minipagestartpars%
10666 }% width not 0
10667 }%
10668 }
```

```
\LWR@select@html@nohspase * {<length>}
\hspase Used to disable \hspase while creating description \items.
```

```
10669 \newcommand{\LWR@select@html@nohspase}{%
10670     \RenewDocumentCommand{\hspase}{s m}{}
10671 }
```

```
\LWR@select@print@hspase
10672 \newcommand*{\LWR@select@print@hspase}{%
10673     \renewrobustcmd\hspase{\@ifstar\@hspacer\@hspase}%
10674 }
```

```
\hspase * {<length>}
Handles special minipage & horizontal space interactions.
```

```
10675 \LWR@select@html@hspase
```

```
\LWR@vspace * {<length>} Nullified vspace.
```

```
10676 \NewDocumentCommand{\LWR@HTML@vspace}{s m}{}
10677
10678 \LWR@formatted{vspace}
```

```
\linebreak [<num>] Inserts an HTML br tag.
```

```
10679 \renewcommand*{\linebreak}[1][]{\newline}
```

```
\nolinelbreak  [<num>]
10680 \renewcommand*{\nolinelbreak}[1][]{}

\pagebreak  [<num>]      Starts a new paragraph.
10681 \renewcommand*{\pagebreak}[1][]{%
10682
10683 }

\nopagebreak  [<num>]
10684 \renewcommand*{\nopagebreak}[1][]{}

\enlargethispage * {<len>}
10685 \RenewDocumentCommand{\enlargethispage}{s m}{}{}

\clearpage
\cleardoublepage
10686 \renewcommand*{\clearpage}{}%
10687 \renewcommand*{\cleardoublepage}{}%

\rule  [<raise>] {<width>} {<height>}
Handles special minipage & horizontal space interactions.
Creates a span of a given width and height. Ignores the optional star.
\fill is zero-width, so \hspace{\fill} is ignored.

10688 \newcommand*{\LWR@HTML@rule}[3][]{\%
The width is copied into a temporary LATEX length, from which comparisons and
conversions may be made:

10689 \setlength{\LWR@tempwidth}{#2}\%
If it's zero-width then skip the entire rule:
10690 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}{%
10691 {}%\ zero- width
10692 {}%\ non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
10693     \ifthenelse{%
10694         \lengthtest{\LWR@tempwidth>0pt}\AND%
10695         \lengthtest{\LWR@tempwidth<1pt}%
10696     }%
10697     {\setlength{\LWR@tempwidth}{1pt}}%
```

Likewise with height:

```
10698     \setlength{\LWR@tempheight}{#3}%
10699     \ifthenelse{%
10700         \lengthtest{\LWR@tempheight>0pt}\AND%
10701         \lengthtest{\LWR@tempheight<1pt}%
10702     }%
10703     {\setlength{\LWR@tempheight}{1pt}}%
```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
10704     \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.

```
10705     \LWR@findcurrenttextcolor%
10706     \LWR@htmntagc{%
10707     span
10708     style=""}
```

The background color is used to draw the filled rule. The color may be changed by \textcolor.

```
10709     \ifbool{FormatWP}{}{\background:\LWR@currenttextcolor ; }%
```

The width and height are printed, converted to PT:

```
10710     width:\LWR@printlength{\LWR@tempwidth} ; %
10711     height:\LWR@printlength{\LWR@tempheight} ; %
```

The raise height is converted to a css transform. The *2 raise multiplier is to approximately match HTML output's X height. Conversion to a L^AT_EX length allows a typical L^AT_EX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a L^AT_EX length limits the allowable syntax. To do: A superior method would compute a ratio of L^AT_EX ex height, then print that to HTML with an ex unit.

```

10712     \ifblank{#1}%
10713     {}%
10714     {%
10715         \setlength{\LWR@tempraise}{\Opt-#1}%
10716         \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
10717         \LWR@orignewline%
10718         -ms-transform: translate(\Opt,\LWR@printlength{\LWR@tempraise}); %
10719         \LWR@orignewline%
10720         -webkit-transform: translate(\Opt,\LWR@printlength{\LWR@tempraise}); %
10721         \LWR@orignewline%
10722         transform: translate(\Opt,\LWR@printlength{\LWR@tempraise}); %
10723         \LWR@orignewline%
10724     }%

```

Display inline-block to place the span inline with the text:

```

10725     display:inline-block;"%
10726     }%

```

If formatting for a word processor, approximate with a number of underscores, in case a span of a given width is not supported:

```

10727     \ifbool{FormatWP}{%
10728         \setlength{\LWR@templengthone}{\LWR@tempwidth}%
10729         \whiledo{\lengthtest{\LWR@templengthone}>1em}{%
10730             \_{}%
10731             \addtolength{\LWR@templengthone}{-1em}%
10732         }%
10733     }{%

```

Close the span:

```

10734     \LWR@htmlltagc{/span}%

```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

10735     \LWR@minipagestartpars%
10736 }% non-zero width
10737 }
10738
10739 \LWR@formatted{rule}

10740 \end{warpHTML}

```

91 \phantomsection

for HTML output: 10741 \begin{warpHTML}

\phantomsection Emulate the hyperref \phantomsection command, often used to insert the bibliography into the table of contents. Ignores \ForceHTMLTOC.

```
10742 \DeclareDocumentCommand{\phantomsection}{}
10743 {%
10744 \begingroup%
10745 \boolfalse{LWR@forcinghtmltoc}%
10746 \section*{}%
10747 \endgroup%
10748 }
```

10748 \end{warpHTML}

92 \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>

and the spacing described in the metafont package documentation.

92.1 HTML logos

for HTML output: 10749 \begin{warpHTML}

\TeX\ TeX

latexlogo is a css class used to properly typeset the E and A in L^AT_EX and friends.

latexlogofont is a css class used to select the font for the rest of the logo in L^AT_EX, Lua^AT_EX, Con^ATEXt, etc.

10750 \LetLtxMacro{\LWR@origTeX}{\TeX}

```
10751
10752 \newrobustcmd*{\LWR@TeX}
10753 {%
10754     \InlineClass{latexlogofont}%
10755     {%
10756         \InlineClass{latexlogo}%
10757         {%
10758             T%
10759             \InlineClass{latexlogosub}{e}%
10760             X%
10761         }%
10762     }%
10763 }

\LaTeX \LATEX, \LATEX2\varepsilon
\LaTeXe
10764 \LetLtxMacro{\LWR@origLaTeX}{\LaTeX}
10765
10766 \newrobustcmd*{\LWR@LaTeX}
10767 {%
10768     \InlineClass{latexlogofont}%
10769     {%
10770         \InlineClass{latexlogo}%
10771         {%
10772             L%
10773             \InlineClass{latexlogosup}{a}%
10774             T%
10775             \InlineClass{latexlogosub}{e}%
10776             X%
10777         }%
10778     }%
10779 }
10780
10781 \LetLtxMacro{\LWR@origLaTeXe}{\LaTeXe}
10782
10783 \renewrobustcmd*{\LaTeXe}
10784 {%
10785     \LaTeX%
10786     \InlineClass{latexlogofont}{%
10787         \InlineClass{latexlogotwoe}{%
10788             2%
10789             \InlineClass{latexlogotwoesub}{\HTMLunicode{03B5}}%
10790         }%
10791     }%
10792 }

\LuaTeX \LuaTeX, \LuaLATEX
\LuaLaTeX
```

```
10793 \newrobustcmd*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
10794 \newrobustcmd*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

\XeTeX X_ETEX, X_LATEX
 \XeLaTeX xetexlogo is a css class which aligns the backwards E in X_ETEX and spaces TEX appropriately.

xelatexlogo is a css class which aligns the backwards E in X_ETEX and spaces L_ETEX appropriately.

```
10795 \newrobustcmd*{\Xe}
10796   {%
10797     X%
10798     \InlineClass{xelatexlogosub}{\HTMLunicode{18e}}%
10799   }
10800
10801 \newrobustcmd*{\LWR@XeTeX}{\InlineClass{xelatexlogo}{\Xe}\TeX}
10802
10803 \newrobustcmd*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}
```

\ConTeXt ConTEXt

```
10804 \newrobustcmd*{\LWR@ConTeXt}
10805 {\InlineClass{latexlogofont}{Con}\TeX{}}
10806 \InlineClass{latexlogofont}{t}
```

\BibTeX BIB_TEX, *MakeIndex*
 \MakeIndex


```
10807 \providerobustcmd*{\BibTeX}
10808 {\InlineClass{latexlogofont}{B\textsc{ib}}}\TeX}
10809
10810 \newrobustcmd*{\MakeIndex}
10811 {\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.

```
10812 \AtBeginDocument{%
10813 \DeclareDocumentCommand{\AmS}{}%
10814 {%
10815   \InlineClass{amslogo}{%
10816     \textit{%
10817       A%
10818       \InlineClass{latexlogosub}{M}%
10819       S%
}}
```

```

10820      }%
10821      }%
10822 }%
10823 }

\MiKTeX MiKTEX

10824 \newrobustcmd*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}

\LyX LyX

lyxlogo is a css class used for the LyX logo.

10825 \newrobustcmd*{\LyX}{\InlineClass{lyxlogo}{LyX}}


10826 \end{warpHTML}

```

92.2 Print logos

```

for PRINT output: 10827 \begin{warpprint}
10828
10829 \newrobustcmd*{\XeTeXrevE}
10830   {\hspace{-.1667em}\raisebox{-.5ex}{E}\hspace{-.125em}}
10831
10832 \AtBeginDocument{
10833 \@ifpackageloaded{graphics}{
10834   \renewrobustcmd*{\XeTeXrevE}
10835     {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
10836 }{
10837   \PackageWarning{l warp}{Load graphicx or graphics
10838     for improved XeTeX logo}
10839 }
10840 }
10841
10842 \providerobustcmd*{\XeTeX}{\mbox{X}\XeTeXrevE\TeX}
10843 \providerobustcmd*{\XeLaTeX}{\mbox{X}\XeTeXrevE\LaTeX}
10844 \providerobustcmd*{\AmS}{%
10845   \leavevmode\hbox{$\mathcal{A}$}\kern-.2em\lower.376ex%
10846   \hbox{$\mathcal{M}$}\kern-.2em\mathcal{S}%
10847 }
10848 \newrobustcmd*{\LyX}{\textsf{LyX}}
10849 \providerobustcmd*{\LuaTeX}{\mbox{Lua}\TeX}
10850 \providerobustcmd*{\LuaLaTeX}{\mbox{Lua}\LaTeX}
10851 \providerobustcmd*{\BibTeX}{\mbox{B}\textit{texsc}{ib}\TeX}
10852 \providerobustcmd*{\MakeIndex}{\mbox{\textit{texit}{MakeIndex}}}
10853 \providerobustcmd*{\ConTeXt}{\mbox{Con}\TeX{}t}

```

```
10854 \providerobustcmd*\{MiKTeX}\{\\mbox{MiK\\TeX}}\\
10855 \\end{warpprint}
```

93 \AtBeginDocument, \AtEndDocument

for HTML output: 10856 \\begin{warpHTML}

```
\\LWR@LwarpStart Automatically sets up the HTML-related actions for the start and end of the document.
\\LWR@LwarpEnd
10857 \\AfterEndPreamble{\\LWR@LwarpStart}
10858 \\AtEndDocument{\\LWR@LwarpEnd}

10859 \\end{warpHTML}
```

94 KOMA-SCRIPT classes

Load patches to koma-script.

for HTML output: 10860 \\begin{warpHTML}

```
10861 \\@ifclassloaded{scrbook}{\\RequirePackage{lwarp-patch-komascript}}{}
10862 \\@ifclassloaded{scrartcl}{\\RequirePackage{lwarp-patch-komascript}}{}
10863 \\@ifclassloaded{scrreprt}{\\RequirePackage{lwarp-patch-komascript}}{}

10864 \\end{warpHTML}
```

95 MEMOIR class

Load patches to memoir.

for HTML output: 10865 \\begin{warpHTML}

```
10866 \\@ifclassloaded{memoir}{\\RequirePackage{lwarp-patch-memoir}}{}

10867 \\end{warpHTML}
```

96 ut^{*} classes

Load patches to uj^{*} and ut^{*} classes, as well as ltj^{*} classes.

for HTML output: 10868 \begin{warpHTML}

```

10869 \newcommand*\LWR@patchujtclasses{%
  uj/t does not use \partname
  10870   \def\@partnameformat{}%
  10871   \def\@partcntformat##1{%
    10872     \prepartname%
    10873     \csname the##1\endcsname%
    10874     \postpartname%
    10875     \quad%
  10876   }%
  10877   \@ifundefined{chapter}{}{%
    10878     \def\@chapcntformat##1{%
      10879       \prechaptername%
      10880       \csname the##1\endcsname%
      10881       \postchaptername%
      10882       \quad%
    10883   }%
  10884 }

```

Use decimal points instead of centered dots:

```

10885   \renewcommand{\thepart}{\@Roman\c@part}
10886   \@ifundefined{chapter}{%
10887     \renewcommand{\thesection}{\@arabic\c@section}%
10888   }{%
10889     \renewcommand{\thechapter}{\@arabic\c@chapter}%
10890     \renewcommand{\thesection}{\thechapter.\@arabic\c@section}%
10891   }%
10892   \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}%
10893   \renewcommand{\thesubsubsection}{%
10894     \thesubsection.\@arabic\c@subsubsection}%
10895   \renewcommand{\theparagraph}{%
10896     \thesubsubsection.\@arabic\c@paragraph}%
10897   \renewcommand{\thesubparagraph}{%
10898     \theparagraph.\@arabic\c@subparagraph}%
10899   \@ifundefined{chapter}{%
10900     \renewcommand{\thefigure}{\@arabic\c@figure}%
10901     \renewcommand{\thetable}{\@arabic\c@table}%
10902   }%

```

```

10903     \renewcommand{\thefigure}{%
10904         \ifnum\c@chapter>\z@\thechapter.\fi\arabic{c@figure}%
10905     \renewcommand{\thetable}{%
10906         \ifnum\c@chapter>\z@\thechapter.\fi\arabic{c@table}%
10907     }%
10908 }%
10909
10910 \@ifclassloaded{ujarticle}{\LWR@patchujtclasses}{}%
10911 \@ifclassloaded{ujbook}{\LWR@patchujtclasses}{}%
10912 \@ifclassloaded{ujreport}{\LWR@patchujtclasses}{}%
10913 \@ifclassloaded{utarticle}{\LWR@patchujtclasses}{}%
10914 \@ifclassloaded{utbook}{\LWR@patchujtclasses}{}%
10915 \@ifclassloaded{utreport}{\LWR@patchujtclasses}{}%
10916 \@ifclassloaded{ltjarticle}{\LWR@patchujtclasses}{}%
10917 \@ifclassloaded{ltjbook}{\LWR@patchujtclasses}{}%
10918 \@ifclassloaded{ltjreport}{\LWR@patchujtclasses}{}%
10919 \@ifclassloaded{ltjsarticle}{\LWR@patchujtclasses}{}%
10920 \@ifclassloaded{ltjsbook}{\LWR@patchujtclasses}{}%
10921 \@ifclassloaded{ltjsreport}{\LWR@patchujtclasses}{}%
10922 \@ifclassloaded{ltjskiyou}{\LWR@patchujtclasses}{}%
10923 \@ifclassloaded{ltjspf}{\LWR@patchujtclasses}{}%
10924 \@ifclassloaded{ltjarticle}{\LWR@patchujtclasses}{}%
10925 \@ifclassloaded{ltjtbook}{\LWR@patchujtclasses}{}%
10926 \@ifclassloaded{ltjtreport}{\LWR@patchujtclasses}{}%
10927 \end{warpHTML}

```

97 CTEXpatch

Patches for ctex and related classes, which are loaded before lwarp.

All CTEX classes and the ctex package seem to load ctexpatch, so its presence is used to decide whether to have lwarp patch CTEX.

for HTML output: 10928 \begin{warpHTML}

\AtBeginDocument in case the user set FileSectionNames in the preamble.

```

10929 \AtBeginDocument{%
10930     \ifpackageloaded{ctexpatch}{%
10931         \def\partcntformat#1{%
10932             \LWR@isolate{\CTEX@partname}~%
10933             \CTEX@part@aftername}%
10934     }%
10935     \def\partnameformat{}%

```

```
10937      \def\@chapcntformat#1{%
10938          \LWR@isolate{\CTEX@chaptername}~%
10939          \CTEX@chapter@aftername%
10940          }%
10941      }{}%
10942  }%
10943 }%
```

10944 \end{warpHTML}

File 2 **l warp-2in1.sty**

§ 98 Package **2in1**

Pkg 2in1 2in1 is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{2in1}

File 3 **l warp-2up.sty**

§ 99 Package **2up**

Pkg 2up 2up is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{2up}[2010/05/15]

```
2 \def\source#1#2#3{}  
3 \def\target#1#2#3{}  
4 \def\targetlayout#1{}  
5 \newdimen\pageseplength  
6 \newdimen\pagesepwidth  
7 \newdimen\pagesepoffset  
8 \def\twoupeemptypage{}  
9 \def\twoupclearpage{}  
10 \def\twoupeject{}  
11 \def\twouparticle{}  
12 \def\twoupplain{}  
13 \def\twouplegaltarget{}  
14 \def\twouplandscape{}  
15 \def\TwoupWrites{}
```

File 4 **l warp-a4.sty**

§ 100 Package **a4**

Pkg a4 a4 is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{a4}

```
2 \newcommand*{\WideMargins}{}
```

File 5 l warp-a4wide.sty

§ 101 Package **a4wide**

Pkg a4wide a4wide is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{a4wide}

File 6 l warp-a5comb.sty

§ 102 Package **a5comb**

Pkg a5comb a5comb is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{a5comb}

File 7 l warp-abstract.sty

§ 103 Package **abstract**

(Emulates or patches code by PETER WILSON.)

Pkg abstract abstract is supported and patched by l warp.

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

for HTML output: memoir provides an abstract environment even though it is not an article or report class. Meanwhile, l warp loads book to emulate memoir, but book does not have an abstract environment, so when the abstract package is loaded for emulation there is no pre-existing abstract to redefine, which would cause an error. Thus, a null abstract is provided here:

```
1 \ProvideDocumentEnvironment{abstract}{}{}{}
```

Accept all options for l warp-abstract:

```
2 \LWR@ProvidesPackagePass{abstract}

3 \AtBeginDocument{
4 \BeforeBeginEnvironment{abstract}{

5 \LWR@forcenewpage
6 \BlockClass{abstract}
7 }
8 \AfterEndEnvironment{abstract}{\endBlockClass}
9 }

10 \renewcommand{\@bsrunintitle}{%
11 \hspace*{\abstitleskip}%
12 {\abstractnamefont{%
13 \InLineClass{abstractrunintitle}{\abstractname}%
14 \@bslabeldelim}%
15 }%
16 }

17
18 \@ifclassloaded{memoir}
19 {
20   \renewenvironment{abstract}{%
21 %     \titlepage
22     \null\vfil
23     \begin{par penalty}\lowpenalty
24     \if@bsrunin
25     \else
26       \if@bsstyle
27         \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}%
28       \else
29         \ifnumber@bs
30           \num@bs
31         \else
32           \begin{\absnamepos}%
33             \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}%
34             \end{par penalty}\OM
35             \end{\absnamepos}%
36 %%             \vspace{\abstitleskip}%
37           \fi
38           \fi
39           \vspace{\abstitleskip}%
40         \fi
41         \put@bsintoc%
42         \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
43         {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
44       }
45 }% not memoir
46 \if@titlepage
47   \renewenvironment{abstract}{%
48 %     \titlepage
49     \null\vfil
```

```
50     \begin{parpenalty}\lowpenalty
51     \if@bsrunin
52     \else
53         \if@bsstyle
54             \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
55         \else
56             \ifnumber@bs
57                 \num@bs
58             \else
59                 \begin{\absnamepos}%
60                 \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}%
61                 \end{parpenalty}\M
62                 \end{\absnamepos}%
63 %%                 \vspace{\abstitleskip}%
64             \fi
65             \fi
66             \vspace{\abstitleskip}%
67         \fi
68         \put@bsintoc%
69         \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
70         {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
71     }
72 \else
73     \renewenvironment{abstract}{%
74     \if@bsrunin
75     \else
76         \if@bsstyle
77             \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
78         \else
79             \ifnumber@bs
80                 \num@bs
81             \else
82                 \begin{\absnamepos}%
83                 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
84                 \end{\absnamepos}%
85 %%                 \vspace{\abstitleskip}%
86             \fi
87             \fi
88             \vspace{\abstitleskip}%
89         \fi
90         \put@bsintoc%
91         \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
92         {\par\end{@bstr@ctlist}}
93 \fi
94 }% not memoir
```

File 8 **l warp-accsupp.sty**

§ 104 Package **accsupp**

Pkg accsupp accsupp is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{accsupp}

2 \newcommand*{\BeginAccSupp}[1]{}
3 \newcommand*{\EndAccSupp}[1]{}
```

File 9 **l warp-acro.sty**

§ 105 Package **acro**

(Emulates or patches code by CLEMENS NIEDERBERGER.)

Pkg acro acro is patched for use by l warp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{acro}
```

\DeclareAcronym is used in the preamble, where l warp has not yet made the dollar active, so temporarily enable l warp math catcode just for this definition:

```

2 \ExplSyntaxOn
3 \NewDocumentCommand \LWR@DeclareAcronym {mm}
4 {
5     \acro_declare_acronym:nn {#1} {#2}
6     \catcode`\$=3% l warp
7 }
8 \ExplSyntaxOff
9
10 \RenewDocumentCommand{\DeclareAcronym}{}{
11     \catcode`\$=\active% l warp
12     \LWR@DeclareAcronym
13 }
```

Modified to activate the current font:

```

14 \ExplSyntaxOn
15 \cs_gset_protected:Npn \acro_write_short:nn #1#2
16 {
```

```
17      \mode_if_horizontal:F { \leavevmode }
18      \group_begin:
19          \bool_if:NTF \l__acro_custom_format_bool
20              { \l__acro_custom_format_tl }
21              { \l__acro_short_format_tl }
22              {\LWR@textcurrentfont{#2}}% lwarp
23      \group_end:
24  }
25
26 \cs_gset_protected:Npn \acro_write_alt:nn #1#2
27  {
28      \mode_if_horizontal:F { \leavevmode }
29      \group_begin:
30          \bool_if:NTF \l__acro_custom_format_bool
31              { \l__acro_custom_format_tl }
32              { \l__acro_alt_format_tl }
33              {\LWR@textcurrentfont{#2}}% lwarp
34      \group_end:
35  }
36
37 \cs_gset_protected:Npn \acro_write_long:nn #1#2
38  {
39      \mode_if_horizontal:F { \leavevmode }
40      \group_begin:
41          \bool_if:NTF \l__acro_custom_long_format_bool
42              { \l__acro_custom_long_format_tl }
43              { \use:n }
44      {
45          \use:x
46          {
47              \exp_not:n {#1}
48          {
49              \bool_if:NTF \l__acro_first_upper_bool
50                  { \exp_not:N \__acro_first_upper_case:n { \exp_not:n {
51                      \LWR@textcurrentfont{#2}}% lwarp
52                  } } }
53                  { \exp_not:n { \LWR@textcurrentfont{#2} } }% lwarp
54          }
55      }
56  }
57  \group_end:
58 }
59 \ExplSyntaxOff
```

File 10 **l warp-acronym.sty**

§ 106 Package **acronym**

(Emulates or patches code by TOBIAS OETIKER.)

Pkg acronym acronym is patched for use by l warp.

⚠ multiply-defined labels \acresetall does not work with cleveref, causing multiply-defined labels. l warp patches acronym for HTML, but not for print mode.

for HTML output: 1 \LWR@ProvidesPackagePass{acronym}

Uses \textit instead of \itshape:

```
2 \renewcommand{\acfia}[1]{%
3   {\textit{\AC@acl{#1}}}} (\ifAC@starred\acs*{#1}\else\acs{#1}\fi)}
```

Removes the mbox to allow math inside:

```
4 \renewcommand*\AC@acs[1]{%
5   \mbox{%
6     \expandafter\AC@get\csname fn@\#1\endcsname\@firstoftwo{#1}%
7   }}
```

Modified for cleveref and zref:

```
8 \renewcommand*\AC@und@newl@bel[3]{%
9   \@ifundefined{#1@#3}{%
10    {%
11      \global\expandafter\let\csname#2@#3\endcsname\@nnil
12      \global\expandafter\let\csname#2@#3@cref\endcsname\@nnil% l warp
13    }%
14    {%
15      \global\expandafter\let\csname#1@#3\endcsname\relax
16      \global\expandafter\let\csname#1@#3@cref\endcsname\relax% l warp
17      \global\expandafter\let\csname Z@R@#3\endcsname\relax% l warp
18    }%
19 }}
```

Modified for cleveref and zref:

```
20 \renewcommand*\AC@testdef[3]{%
21 \ifstreq{\#1}{Z@R}{\{}{\% l warp
```

```

22  \@ifundefined{s@#2}\@secondoftwo\@firstofone
23  {%
24    \expandafter\ifx\csname s@#2\endcsname\empty
25      \expandafter\@firstofone
26    \else
27      \expandafter\xdef\csname s@#2\endcsname{%
28        \expandafter\expandafter
29        \expandafter\gobble
30        \csname s@#2\endcsname
31      }%
32      \expandafter\@gobble
33    \fi
34  }%
35  {%
36    \@testdef{#1}{#2}{#3}%
37  }%
38 }% l warp
39 }%

```

File 11 **l warp-adjmulticol.sty**

§ 107 Package **adjmulticol**

(Emulates or patches code by BORIS VEYTSMAN.)

Pkg adjmulticol adjmulticol is emulated.

Emulation similar to multicols is used, with adjusted margins. If the number of columns is specified as 1, it is set so, but if two or greater are used, l warp allows a variable number of columns up to three.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{adjmulticol}

2 \RequirePackage{multicol}

adjmulticols * {\langle numcols \rangle} {\langle left margin \rangle} {\langle right margin \rangle}

3 \NewDocumentEnvironment{adjmulticols}{s m m m}
4 {%

```

Compute the margins, and limit to positive only:

```

5 \setlength{\LWR@templengthone}{#3}%
6 \ifdimcomp{\LWR@templengthone}{<}{0pt}{\setlength{\LWR@templengthone}{0pt}}{%
7 \setlength{\LWR@templengthtwo}{#4}%
8 \ifdimcomp{\LWR@templengthtwo}{<}{0pt}{\setlength{\LWR@templengthtwo}{0pt}}{%

```

If one column is specified, use a <div> of class `singlecolumn`, else use `multicols`:

```
9 \newcommand*{\LWR@mcolstype}{multicols}%
10 \ifnumcomp{#2}{=}{1}{\renewcommand*{\LWR@mcolstype}{singlecolumn}}{}%
```

Help avoid page overflow:

```
11 \LWR@forcenewpage%
```

Create the <div> with the given margin and class:

```
12 \BlockClass[%
13 \LWR@print@mbox{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
14 \LWR@print@mbox{margin-right:\LWR@printlength{\LWR@templengthtwo}}%
15 ]{\LWR@mcolstype}%
16 }
17 {\endBlockClass}
```

File 12 lwarf-addlines.sty

§ 108 Package **addlines**

(Emulates or patches code by WILL ROBERTSON.)

Pkg `addlines` `addlines` is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{addlines}[2018/12/05]

```
2 \newcommand\addlines{\@ifstar\addlines@a\addlines@a}
3 \newcommand\addlines@a[1][1]{}
4 \let\addline\addlines
5 \newcommand\removelines{\@ifstar\removelines@a\removelines@a}
6 \newcommand\removelines@a[1][1]{}
7 \let\removeline\removelines
8 \newcommand\squeeze[1][0]{}
```

File 13 lwarf-afterpage.sty

§ 109 Package **afterpage**

(Emulates or patches code by DAVID CARLISLE.)

Pkg `afterpage` Emulated.

for HTML output: Discard all options for l warp-afterpage:

```
1 \LWR@ProvidesPackageDrop{afterpage}
2 \newcommand{\afterpage}[1]{#1}
```

File 14 **l warp-algorithm2e.sty**

§ 110 Package **algorithm2e**

(Emulates or patches code by CHRISTOPHE FIORIO.)

Pkg algorithm2e algorithm2e is patched for use by l warp.

For print output, captions are placed according to package options, but for HTML output captions are placed where used. Therefore, to have captions appear at the top of the algorithms for both print and HTML, place each captions at the top of each algorithm.

for HTML output: 1 \LWR@ProvidesPackagePass{algorithm2e}

For the list-of entries:

```
2 \renewcommand{\l@algocf}[2]{\hypertocfloat{1}{algocf}{loa}{#1}{#2}}
```

Select the l warp float style according to the algorithm2e style:

```
3 \newcommand*{\LWR@floatstyle@algocf}{ruled}
4
5 \ifdefstring{\algocf@style}{boxed}{%
6 \renewcommand*{\LWR@floatstyle@algocf}{boxed}
7 }{%
8
9 \ifdefstring{\algocf@style}{boxruled}{%
10 \renewcommand*{\LWR@floatstyle@algocf}{boxruled}
11 }{%
12
13 \ifdefstring{\algocf@style}{plain}{%
14 \renewcommand*{\LWR@floatstyle@algocf}{plain}
15 }{}}
```

Paragraph handling to allow line numbers under certain conditions:

```
16 \newbool{\LWR@algocf@dopars}
17 \booltrue{\LWR@algocf@dopars}
18
```

```

19 \renewcommand{\algocf@everypar}{%
20 \ifbool{LWR@algocf@dopars}{%
21   \ifbool{LWR@doingstartpars}{%
22     \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}{%
23       {}%
24     }{%
25       \algocf@everyparnl\algocf@everyparhanging%
26     }%
27   }{}%
28 }{}%
29 }

```

lwarp caption handling:

```

30 \renewcommand{\algocf@makedescription}[2]{%
31 \LWR@HTML@caption@begin{\algocf}%
32 \LWR@isolate{\algocf@captiontext{\#1}{\#2}}%
33 \LWR@HTML@caption@end%
34 }

```

Print any caption where it is declared:

```

35 \renewcommand{\algocf@makedescription@plain}[2]{%
36   \LWR@HTML@caption@begin{\algocf}%
37   \LWR@isolate{\algocf@captiontext{\#1}{\#2}}%
38   \LWR@HTML@caption@end%
39 }
40
41 \renewcommand{\algocf@makedescription@boxed}[2]{%
42   \LWR@HTML@caption@begin{\algocf}%
43   \LWR@isolate{\algocf@captiontext{\#1}{\#2}}%
44   \LWR@HTML@caption@end%
45 }
46
47 \renewcommand{\algocf@makedescription@ruled}[2]{%
48   \LWR@HTML@caption@begin{\algocf}%
49   \LWR@isolate{\algocf@captiontext{\#1}{\#2}}%
50   \LWR@HTML@caption@end%
51 }

```

Turn off line numbering while making the caption:

```

52 \long\def\algocf@latexcaption#1[#2]#3{%
53   \boolfalse{LWR@algocf@dopars}%
54   \l warp
55   \par%
56   \addcontentsline{\csname ext@#1\endcsname}{#1}%
57   {\protect\numberline{\csname the#1\endcsname}\ignorespaces \LWR@isolate{\#2}}%
58   \begingroup%
59   \parboxrestore%
60 }

```

```

59  \if@minipage%
60    \@setminipage%
61  \fi%
62  \normalsize%
63  \makecaption{\csname fnum@\#1\endcsname}{\ignorespaces #3}\par%
64  \endgroup%
65 \booltrue{LWR@algocf@dopars}%
66 } lwarf

```

Line numbers are printed in a `` of class `alg2elinenumber`:

```

67 \renewcommand{\algocf@printnl}[1]{%
68   \InLineClass{alg2elinenumber}{\NlSty{#1}}{%
69 }%

```

While initializing an algorithm environment, locally declare the style of a regular figure to be the same as the algorithm style, in case the `figure` option was used.

```

70 \preto{\algocf@init}{%
71   \edef\LWR@floatstyle@figure{\LWR@floatstyle@algocf}%
72 }%

```

For `lwarf`, the algorithm is not assembled inside a box, since `lateximages` would not work, so the captions are printed where declared.

```

73 \renewcommand{\@algocf@start}{%
74   \let\@mathsemicolon=\; \def\;{\ifmmode\@mathsemicolon\else\@endalgoln\fi}%
75 %   \raggedright%
76   \AlFnt{}%
77   \booltrue{LWR@algocf@dopars}%
78 } lwarf
79
80 \renewcommand{\@algocf@finish}{%
81   \boolfalse{LWR@algocf@dopars}%
82   \lineskip\normallineskip\setlength{\skiptotal}{\@defaultskiptotal}%
83   \let\;=\@mathsemicolon%
84   \let\;=\@emathdisplay%
85 }%

```

Use an HTML break:

```

86 \renewcommand{\BlankLine}{%
87 \LWR@stoppars%
88 \LWR@htmlltagc{br /}%
89 \LWR@startpars%
90 }%

```

Simplified for HTML. The paragraph handling must be preserved.

```

91 \renewcommand{\SetKwInOut}[2]{%
92   \algocf@newcommand{#1}[1]{%
93     \ifthenelse{\boolean{\algocf@hanginginout}}{%
94       {\relax}%
95       {\algocf@seteveryparhanging{\relax}}%
96     \ifthenelse{\boolean{\algocf@inoutnumbered}}{%
97       {\relax}%
98       {\algocf@seteveryparnl{\relax}}%
99     }%
100       \KwSty{#2\algocf@typo:}%
101     ~##1\par%
102   }%
103   \algocf@linesnumbered% reset the numbering of the lines
104   \ifthenelse{\boolean{\algocf@hanginginout}}{%
105     {\relax}%
106     {\algocf@reseteveryparhanging}%
107   }%
108 }%
109
110 \renewcommand{\ResetInOut}[1]{%

```

Each of the following creates a <div> of a given class, and turns off line numbering while creating the <div> tags:

```

111 \renewcommand{\algocf@Vline}[1]{%
112 \boolfalse{LWR@algocf@dopars}%
113 \begin{BlockClass}{alg2evline}%
114 \booltrue{LWR@algocf@dopars}%
115 #1
116 \boolfalse{LWR@algocf@dopars}%
117 \end{BlockClass}%
118 \booltrue{LWR@algocf@dopars}%
119 }

120 \renewcommand{\algocf@Vsline}[1]{%
121 \boolfalse{LWR@algocf@dopars}%
122 \begin{BlockClass}{alg2evsline}%
123 \booltrue{LWR@algocf@dopars}%
124 #1
125 \boolfalse{LWR@algocf@dopars}%
126 \end{BlockClass}%
127 \booltrue{LWR@algocf@dopars}%
128 }

129 \renewcommand{\algocf@Noline}[1]{%
130 \boolfalse{LWR@algocf@dopars}%
131 \begin{BlockClass}{alg2enoline}%
132 \booltrue{LWR@algocf@dopars}%
133 #1

```

```

134 \boolfalse{LWR@algocf@dopars}%
135 \end{BlockClass}
136 \booltrue{LWR@algocf@dopars}%
137 }

```

The [H] environment is converted to a regular float, which in HTML is placed where declared. Reusing the regular float allows the [H] version to reuse the ruled and boxed options.

```

138 \LetLtxMacro\algocf@Here\algocf
139 \LetLtxMacro\endalgocf@Here\endalgocf

```

File 15 l warp-algorithmicx.sty

§ 111 Package **algorithmicx**

(Emulates or patches code by SZÁSZ JÁNOS.)

Pkg algorithmicx algorithmicx is supported with minor adjustments.

for HTML output: 1 \LWR@ProvidesPackagePass{algorithmicx}

Inside the algorithmic environment, level indenting is converted to a of the required length, and comments are placed inside a which is floated right.

⚠ package conflicts If using \newfloat, trivfloat, and/or algorithmicx together, see section 429.1.

for HTML output: 2 \begin{warpHTML}

```

3 \AtBeginEnvironment{algorithmic}{%
4 %
5 \let\origALG@doentity\ALG@doentity%
6 %
7 \renewcommand*\ALG@doentity{%
8 \origALG@doentity%
9 \LWR@htmltagc{%
10 span style="width:\LWR@printlength{\ALG@thistlm}; display:inline-block;"%
11 }%
12 \ifbool{FormatWP}{%
13 \setlength{\LWR@templengthone}{\the\ALG@thistlm}%
14 \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
15 \quad%
16 \addtolength{\LWR@templengthone}{-1em}%
17 }%
18 }{}%
19 \LWR@htmltagc{/span}%

```

```
20 }%
21
22 \let\LWR@origComment\Comment%
23
24 \renewcommand{\Comment}[1]{%
25     \InlineClass{floatright}{\LWR@origComment{#1}}%
26 }%
27 }
28
29 \renewcommand\algorithmiccomment[1]{%
30 \hfill\HTMLunicode{25B7} #1% white right triangle
31 }%

32 \end{warpHTML}
```

File 16 l warp-alltt.sty**§ 112 Package alltt**

(Emulates or patches code by JOHANNES BRAAMS.)

Pkg alltt alltt is patched for use by l warp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{alltt}

2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching alltt.}
4 \AtBeginEnvironment{alltt}{%
5 \LWR@forcenewpage
6 \LWR@atbeginverbatim{3}{alltt}%
7 }
8 \AfterEndEnvironment{alltt}{%
9     \LWR@afterendverbatim{2}%
10 }
11 }
```

File 17 l warp-amsmath.sty**§ 113 Package amsmath**

(Emulates or patches code by AMERICAN MATHEMATICAL SOCIETY, LATEX3 PROJECT.)

Pkg amsmath amsmath is patched for use by l warp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{amsmath}
```

Patches to allow \eqref inside a caption:

```
2 \def\maketag@@@#1{\text{#1}}
3 \def\tagform@#1{\maketag@@@{(\ignorespaces#1\unskip)}}
```

Patches for *AMS* math \tag macro to remember the first tag:

```
4 \ifbool{mathjax}{}{%
5
6 \LetLtxMacro{\LWR@origmake@df@tag@@}{\make@df@tag@@}
7 \LetLtxMacro{\LWR@origmake@df@tag@@@}{\make@df@tag@@@}
8
9 \renewcommand*{\make@df@tag@@}[1]{%
10 \LWR@remembertag{#1}%
11 \LWR@origmake@df@tag@@{#1}%
12 }
13
14 \renewcommand*{\make@df@tag@@@}[1]{%
15 \LWR@remembertag{#1}%
16 \LWR@origmake@df@tag@@@{#1}%
17 }
18
19 }% not mathjax
```

The following *AMS* environments are more easily patched in-place:

Env **multline**

```
20 \BeforeBeginEnvironment{multline}{
21
22 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{%
23 {
24     \LWR@syncmathjax
25     \booltrue{\LWR@amsmultiline}
26     \LWR@beginhideamsmath
27 }
28 {
29     \begin{BlockClass}{displaymathnumbered}
30     \LWR@newautoidanchor%
31     \booltrue{\LWR@indisplaymathimage}%
32     \begin{lateXimage}[\LWR@amsmathbodynumbered{multline}]
33 }
34 }
35
36 \AfterEndEnvironment{multline}{%
37 }
```

```
38 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
39 {
40     \LWR@endhideamsmath
41     \boolfalse{LWR@amsmultiline}
42     \LWR@addmathjax{multiline}{\the\@envbody}
43 }
44 {\end{lateximage}\end{BlockClass}}
45
46 }

Env  multiline*
47 \BeforeBeginEnvironment{multiline*}{

48
49 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
50 {
51     \LWR@syncmathjax
52     \booltrue{LWR@amsmultiline}
53     \LWR@beginhideamsmath
54 }
55 {
56     \begin{BlockClass}{displaymath}
57     \LWR@newautoidanchor
58     \booltrue{LWR@indisplaymathimage}%
59     \begin{lateximage}[\LWR@amsmathbody{multiline*}]
60 }
61 }
62
63 \AfterEndEnvironment{multiline*}{

64
65 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
66 {
67     \LWR@endhideamsmath
68     \boolfalse{LWR@amsmultiline}
69     \LWR@addmathjax{multiline*}{\the\@envbody}
70 }
71 {\end{lateximage}\end{BlockClass}}
72
73 }
74

Env  gather
75 \BeforeBeginEnvironment{gather}{

76
77 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
78 {
79     \LWR@syncmathjax
```

```
80      \boolfalse{LWR@amsmultiline}
81      \LWR@beginhideamsmath
82 }
83 {
84     \begin{BlockClass}{displaymathnumbered}
85     \LWR@newauto{danchor}%
86     \booltrue{LWR@indisplaymathimage}%
87     \begin{lateximage}[\LWR@amsmathbody{numbered}{gather}]
88 }
89 }
90
91 \AfterEndEnvironment{gather}{
92
93 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{
94 {
95     \LWR@endhideamsmath
96     \LWR@addmathjax{gather}{\the\@envbody}
97 }
98 \end{lateximage}\end{BlockClass}
99
100 }
```

Env **gather***

```
101 \BeforeBeginEnvironment{gather*}{
102
103 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{
104 {
105     \LWR@syncmathjax
106     \boolfalse{LWR@amsmultiline}
107     \LWR@beginhideamsmath
108 }
109 {
110     \begin{BlockClass}{displaymath}
111     \LWR@newauto{danchor}%
112     \booltrue{LWR@indisplaymathimage}%
113     \begin{lateximage}[\LWR@amsmathbody{gather*}]
114 }
115 }
116
117 \AfterEndEnvironment{gather*}{
118
119 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{
120 {
121     \LWR@endhideamsmath
122     \LWR@addmathjax{gather*}{\the\@envbody}
123 }
124 \end{lateximage}\end{BlockClass}
125 }
```

```
126 }

Env  align

127 \BeforeBeginEnvironment{align}{

128
129 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
130 {
131     \LWR@syncmathjax
132     \boolefalse{\LWR@amsmultiline}
133     \LWR@beginhideamsmath
134 }
135 {
136     \begin{BlockClass}{displaymathnumbered}
137     \LWR@newautoidanchor%
138     \booletrue{\LWR@indisplaymathimage}%
139     \begin{lateximage}[\LWR@amsmathbodynumbered{align}]
140 }
141 }
142
143 \AfterEndEnvironment{align}{

144 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
145 {
146     \LWR@endhideamsmath
147     \LWR@addmathjax{align}{\the\@envbody}
148 }
149
150 {\end{lateximage}\end{BlockClass}}
151
152 }

Env  align*

153 \BeforeBeginEnvironment{align*}{

154
155 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
156 {
157     \LWR@syncmathjax
158     \boolefalse{\LWR@amsmultiline}
159     \LWR@beginhideamsmath
160 }
161 {
162     \begin{BlockClass}{displaymath}
163     \LWR@newautoidanchor%
164     \booletrue{\LWR@indisplaymathimage}%
165     \begin{lateximage}[\LWR@amsmathbody{align*}]
166 }
167 }
```

```
168
169 \AfterEndEnvironment{align*}{
170
171 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
172 {
173     \LWR@endhideamsmath
174     \LWR@addmathjax{align*}{\the\@envbody}
175 }
176 {\end{lateximage}\end{BlockClass}}
177
178 }
```

Env flalign

```
179 \BeforeBeginEnvironment{flalign}{
180
181 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
182 {
183     \LWR@syncmathjax
184     \boolfalse{LWR@amsmultiline}
185     \LWR@beginhideamsmath
186 }
187 {
188     \begin{BlockClass}{displaymathnumbered}
189     \LWR@newautoidanchor%
190     \booltrue{LWR@indisplaymathimage}%
191     \begin{lateximage}[\LWR@amsmathbodynumbered{flalign}]
192 }
193 }
194
195 \AfterEndEnvironment{flalign}{
196
197 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
198 {
199     \LWR@endhideamsmath
200     \LWR@addmathjax{flalign}{\the\@envbody}
201 }
202 {\end{lateximage}\end{BlockClass}}
203
204 }
```

Env flalign*

```
205 \BeforeBeginEnvironment{flalign*}{
206
207 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
208 {
209     \LWR@syncmathjax
```

```
210      \boolearnfalse{LWR@amsmultiline}
211      \LWR@beginhideamsmath
212 }
213 {
214     \begin{BlockClass}{displaymath}
215     \LWR@newautooidanchor%
216     \boolearntrue{LWR@indisplaymathimage}%
217     \begin{lateximage}[\LWR@amsmathbody{flalign*}]
218 }
219 }
220
221 \AfterEndEnvironment{flalign*}{
222
223 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{
224 {
225     \LWR@endhideamsmath
226     \LWR@addmathjax{flalign*}{\the\envbody}
227 }
228 {\end{lateximage}\end{BlockClass}}
229
230 }
```

Env alignat

```
231 \BeforeBeginEnvironment{alignat}{

232
233 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{
234 {
235     \LWR@syncmathjax
236     \boolearnfalse{LWR@amsmultiline}
237     \LWR@beginhideamsmath
238 }
239 {
240     \begin{BlockClass}{displaymathnumbered}
241     \LWR@newautooidanchor%
242     \boolearntrue{LWR@indisplaymathimage}%
243     \begin{lateximage}[\LWR@amsmathbodynumbered{alignat}]
244 }
245 }
246
247 \AfterEndEnvironment{alignat}{

248
249 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{
250 {
251     \LWR@endhideamsmath
252     \LWR@addmathjax{alignat}{\the\envbody}
253 }
254 {\end{lateximage}\end{BlockClass}}
255 }
```

```

256 }

Env alignat*
257 \BeforeBeginEnvironment{alignat*}{

258
259 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
260 {
261     \LWR@syncmathjax
262     \boolfalse{LWR@amsmultiline}
263     \LWR@beginhideamsmath
264 }
265 {
266     \begin{BlockClass}{displaymath}
267     \LWR@newautoidanchor%
268     \booltrue{LWR@indisplaymathimage}%
269     \begin{lateximage}[\LWR@amsmathbody{alignat*}]
270 }
271 }
272
273 \AfterEndEnvironment{alignat*}{

274 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
275 {
276     \LWR@endhideamsmath
277     \LWR@addmathjax{alignat*}{\the\@envbody}
278 }
279
280 {\end{lateximage}\end{BlockClass}}
281
282 }

```

File 18 **l warp-amsthm.sty**

§ 114 Package **amsthm**

(Emulates or patches code by PUBLICATIONS TECHNICAL GROUP—AMERICAN MATHEMATICAL SOCIETY.)

The original source code is located in `amsclass.dtx`, and printed in `amsclass.pdf`.

Pkg amsthm amsthm is patched for use by l warp.

for HTML output: amsthm must be loaded before mdframed:

```

1 \@ifpackageloaded{mdframed}{
2     \PackageError{l warp}{%
3         {Package mdframed must be loaded after package amsthm.}}

```

Table 14: amsthm package—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.

```

4      {%
5      Move \detokenize{\usepackage}{amsthm} before \detokenize{\usepackage}{mdframed}.
6      Package amsthm may be loaded by something else, which must also be moved
7      before mdframed.%}
8  }
9 }
10 {}}

11 \LWR@ProvidesPackagePass{amsthm}
```

Storage for the style being used for new theorems:

```
12 \newcommand{\LWR@newtheoremstyle}[1]{plain}
```

Patched to remember the style being used for new theorems:

```

13 \renewcommand{\theoremstyle}[1]{%
14   \@ifundefined{th@#1}{%
15     \PackageWarning{amsthm}{Unknown theoremstyle '#1'}%
16     \thm@style{plain}%
17     \renewcommand{\LWR@newtheoremstyle}[1]{plain}\% l warp
18   }{%
19     \thm@style{#1}%
20     \renewcommand{\LWR@newtheoremstyle}[1]{#1}\% l warp
21   }%
22 }
```

Patched to remember the style for this theorem type:

```

23 \def\@xnthm#1#2{%
24   \csedef{\LWR@thmstyle#2}{\LWR@newtheoremstyle}\% l warp
```

```

25  \let\@tempa\relax
26  \@xp\@ifdefinable\csname #2\endcsname{%
27    \global\@xp\let\csname end#2\endcsname\@endtheorem
28    \ifx **#1% unnumbered, need to get one more mandatory arg
29      \edef\@tempa##1{%
30        \gdef\@xp\@nx\csname#2\endcsname{%
31          \@nx\@thm{\@xp\@nx\csname th@\the\thm@style\endcsname}{%
32            {}{##1}}}
33      \else % numbered theorem, need to check for optional arg
34        \def\@tempa{\oparg{\cynthm{#2}}[]}{%
35      \fi
36      \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\@nameuse{LWR@thmstyle#2}}}\% l warp
37  }%
38  \@tempa
39 }

```

Patched to enclose with css:

```

40 \newcommand{\LWR@haveamsthmname}{%
41 \renewcommand{\thmname}[1]{\InlineClass{amsthmname}\LWR@thisthmstyle}{##1}}
42 }
43
44 \newcommand{\LWR@haveamsthmnumber}{%
45 \renewcommand{\thmnumber}[1]{\InlineClass{amsthmnumber}\LWR@thisthmstyle}{##1}}
46 }
47
48 \newcommand{\LWR@haveamsthmnote}{%
49 \renewcommand{\thmnote}[1]{\InlineClass{amsthmnote}\LWR@thisthmstyle}{##1}}
50 }
51
52 \LWR@haveamsthmname
53 \LWR@haveamsthmnumber
54 \LWR@haveamsthmnote

```

Patches for css:

```

55 \def\@begintheorem#1#2[#3]{%
56   \item[
57 %   \deferred@thm@head{
58 %     \the\thm@headfont \thm@indent
59   \@ifempty{#1}{\let\thmname\@gobble}{\LWR@haveamsthmname}\% l warp
60   \@ifempty{#2}{\let\thmnumber\@gobble}{\LWR@haveamsthmnumber}\% l warp
61   \@ifempty{#3}{\let\thmnote\@gobble}{\LWR@haveamsthmnote}\% l warp
62   \thm@swap\swappedhead\thmhead{#1}{#2}{#3}\%
63   \the\thm@headpunct-
64   \thmheadnl % possibly a newline.
65   \hskip\thm@headsep
66 %   }%
67 ]

```

```
68 \ignorespaces}
```

Patched for css:

```
69 \def\@thm#1#2#3{%
70   \ifhmode\unskip\unskip\par\fi
71   \normalfont
72   \LWR@forcenewpage% lwarp
73   \BlockClass{amsthmbody}\LWR@thisthmstyle}% lwarp
74   \trivlist
75   \let\thmheadnl\relax
76   \let\thm@swap\@gobble
77   \thm@notefont{\fontseries\mddefault\upshape}%
78   \thm@headpunct{.}% add period after heading
79   \thm@headsep 5\p@ plus\p@ minus\p@\relax
80   \thm@space@setup
81   #1% style overrides
82   \topsep \thm@preskip % used by thm head
83   \topsepadd \thm@postskip % used by \endparenv
84   \def\@tempa{#2}\ifx\@empty\@tempa
85     \def\@tempa{\oparg{\begin{theorem}{#3}{}[]}}%
86   \else
87     \refstepcounter{#2}%
88     \def\@tempa{\oparg{\begin{theorem}{#3}{\csname the#2\endcsname}[]}}%
89   \fi
90   \@tempa
91 }
```

cleveref patches \atBeginDocument to do \cref@thmoptarg if an optional argument is given. lwarp then patches \cref@thmoptarg \AtBeginDocument.

```
92 \AtBeginDocument{
93 \def\cref@thmoptarg[#1]#2#3#4{%
94   \ifhmode\unskip\unskip\par\fi%
95   \normalfont%
96   \LWR@forcenewpage% lwarp
97   \BlockClass{amsthmbody}\LWR@thisthmstyle}% lwarp
98   \trivlist%
99   \let\thmheadnl\relax%
100  \let\thm@swap\@gobble%
101  \thm@notefont{\fontseries\mddefault\upshape}%
102  \thm@headpunct{.}% add period after heading
103  \thm@headsep 5\p@ plus\p@ minus\p@\relax%
104  \thm@space@setup%
105  #2% style overrides
106  \topsep \thm@preskip % used by thm head
107  \topsepadd \thm@postskip % used by \endparenv
108  \def\@tempa{#3}\ifx\@empty\@tempa
109    \def\@tempa{\oparg{\begin{theorem}{#4}{}[]}}%
```

```

110      \else%
111          \refstepcounter[#1]{#3}\% <<< cleveref modification
112          \def\@tempa{\oparg{\begin{theorem}{#4}{\csname the#3\endcsname}[]}}%
113      \fi%
114      \@tempa
115 }%
116 }% AtBeginDocument
117
118 \def\endtheorem{\endtrivlist\endBlockClass\endpefalse }

```

Proof QED symbol:

```

119 \AtBeginDocument{
120 \ifundefined{LWR@orig@openbox}{
121 \LetLtxMacro{LWR@orig@openbox}{\openbox}
122 \LetLtxMacro{LWR@orig@blacksquare}{\blacksquare}
123 \LetLtxMacro{LWR@orig@Box}{\Box}
124
125 \def\openbox{\text{\HTMLunicode{25A1}}}\% UTF-8 white box
126 \def\blacksquare{\text{\HTMLunicode{220E}}}\% UTF-8 end-of-proof
127 \def\Box{\text{\HTMLunicode{25A1}}}\% UTF-8 white box
128
129 \appto{LWR@restoreorigformatting}{%
130 \LetLtxMacro{\openbox}{LWR@orig@openbox}%
131 \LetLtxMacro{\blacksquare}{LWR@orig@blacksquare}%
132 \LetLtxMacro{\Box}{LWR@orig@Box}%
133 }% appto
134 }{}% @ifundefined
135 }% AtBeginDocument

```

Patched for css:

```

136 \renewenvironment{proof}[1][\proofname]{\par
137 \LWR@forcenewpage% lwarp
138     \BlockClass{amsthmproof}\% lwarp
139     \pushQED{\qed}\%
140     \normalfont \topsep6\p@\plus6\p@\relax
141     \trivlist
142     \item[
143         \InlineClass{amsthmproofname}{\#1\@addpunct{.}}]\ignorespaces\% changes
144     }%
145     \InlineClass{theoremendmark}{\popQED}\endtrivlist%
146     \endBlockClass% lwarp
147     \endpefalse
148 }

```

File 19 l warp-anonchap.sty**§ 115 Package anonchap**

(Emulates or patches code by PETER WILSON.)

Pkg anonchap anonchap is emulated.

Pkg tocloft If using tocloft with tocbibind, anonchap, fncychap, or other packages which change chapter title formatting, load tocloft with its titles option, which tells tocloft to use standard L^AT_EX commands to create the titles, allowing other packages to work with it.

The code is shared by tocbibind.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{anonchap}

2 \newcommand{\simplechapter}[1][]{\@empty} {%
3   \def\@chapcntformat##1{%
4     #1~\csname the##1\endcsname\simplechapterdelim\quad%
5   }%
6 }
7
8 \newcommand{\restorechapter}{%
9 \let\@chapcntformat\@secntformat%
10 }
```

File 20 l warp-any size.sty**§ 116 Package anysize**

(Emulates or patches code by MICHAEL SALZENBERG, THOMAS ESSER.)

Pkg anysize anysize is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{any size}

2 \def\papersize#1#2{}%
3 \def\marginsize#1#2#3#4{}
```

File 21 **l warp-appendix.sty**

§ 117 Package **appendix**

(Emulates or patches code by PETER WILSON.)

Pkg appendix appendix is patched for use by l warp.

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

for HTML output:

```
1 \LWR@ProvidesPackagePass{appendix}

2 \renewcommand*{\@chap@pppage}{%
3 \part*{\appendixpagename}
4 \if@dotoc@pp
5 \addaptheadtotoc
6 \fi
7 }
8
9 \renewcommand*{\@sec@pppage}{%
10 \part*{\appendixpagename}
11 \if@dotoc@pp
12 \addaptheadtotoc
13 \fi
14 }
```

File 22 **l warp-arabicfront.sty**

§ 118 Package **arabicfront**

Pkg arabicfront arabicfront is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{arabicfront}
```

File 23 l warp-array.sty**§ 119 Package array**

Pkg array array is used as-is for print output, and emulated for HTML.

plarray and plectarray do not affect \firsthline or \lasthline, and so are not affected by the following.

for HTML output: Remove the default nullified macros:

```
1 \let\firsthline\relax
2 \let\lasthline\relax
3
4 \LWR@ProvidesPackagePass{array}

5 \newcommand*{\LWR@HTML@firsthline}{\LWR@HTMLhline}%
6 \LWR@expandableformatted{firsthline}
7
8 \newcommand*{\LWR@HTML@lasthline}{\LWR@HTMLhline}%
9 \LWR@expandableformatted{lasthline}
```

File 24 l warp-arydshln.sty**§ 120 Package arydshln**

(Emulates or patches code by HIROSHI NAKASHIMA.)

Pkg arydshln arydshln heavily patches tabular code, so the actual package is not used. arydshln is emulated for HTML tabular, and reverts to solid rules for SVG math array and tabular in a lateximage.

css is not able to display a double-dashed border, so a single-dashed rule is displayed as a single-dashed border, and a double-dashed rule is displayed as a thicker single-dashed border.

for HTML output: array is required to allow \newcolumn below.

```
1 \RequirePackage{array}
2 \LWR@ProvidesPackageDrop{arydshln}
```

Ignored, but included for source compatibility:

```

3 \newdimen\dashlinedash \dashlinedash4pt %
4 \newdimen\dashlinegap \dashlinegap4pt %
5 \let\hdashlinewidth\dashlinedash
6 \let\hdashlinegap\dashlinegap
7
8 \def\ADLnullwide{}%
9 \def\ADLsomewide{}%
10 \def\ADLnullwidehline{}%
11 \def\ADLsomewidehline{}%
12
13 \def\ADLactivate{}%
14 \def\ADLinactivate{}%
15 \newcommand*{\ADLdrawingmode}[1]{}%
16 \newcommand*{\ADLnoshorthanded}{}%
17 \newcommand*{\dashgapcolor}[2]{}%
18 \newcommand*{\nodashgapcolor}{}%

```

In a `lateximage`, revert to solid vertical rules:

```

19 \appto\LWR@restoreorigformatting{%
20 \newcolumntype{::}{|}%
21 \newcolumntype{;}{[1]{|}}%
22 \LetLtxMacro\hdashline\hline%
23 }

```

Some of these macros are already defined as temporary placeholders in the `lwarp` core, so they must be redefined here.

The emulated defaults also work for an emulated print mode inside a `lateximage`:

```

24 \def\hdashline{
25 %      \adl@hdashline\adl@ihdashline
26      \adl@hdashline\adl@inactivehdl
27 }
28 \def\adl@hdashline#1{\noalign{\ifnum0='}\fi
29 %          \ifadl@zwhrule \vskip-\arrayrulewidth
30 %          \else
31 %              \adl@hline\adl@connect\arrayrulewidth
32 %              \hrule \height \arrayrulewidth% lwarp
33 %          \fi
34          \@ifnextchar[%]
35              {#1}%
36              {#1[%}
37 %              \dashlinedash/\dashlinegap
38              1pt/1pt
39          ]}%
40 % \def\adl@ihdashline[#1/#2]{\ifnum0='{\fi}%

```

```

41 %          \multispan{\adl@columns}\unskip \adl@hcline\z@[#1/#2]%
42 %          \noalign{\ifnum0='}\fi
43 %          \futurelet@\tempa\adl@xhline}
44 \def\adl@inactivehd1[#1/#2]{
45 %          \ifadl@zwhrule \vskip-\arrayrulewidth \fi
46          \hrule\@height\arrayrulewidth
47          \futurelet@\tempa\adl@xhline}
48 \def\adl@xhline{\ifx\@tempa\hline \adl@ixhline\fi
49          \ifx\@tempa\hbox{\adl@ixhline\fi
50          \ifnum0='{\fi}}
51 \def\adl@ixhline{\vskip\doublerulesep \adl@hline\relax\doublerulesep}
52 \def\adl@hline#1#2{%
53 % \tempcpta#2
54 %          \global\advance\adl@totalheight\tempcpta
55 %          \xdef\adl@rowsL{\adl@rowsL
56 %          (#1/\number\tempcpta);}%
57 %          \xdef\adl@rowsR{\adl@rowsR
58 %          (#1/\number\tempcpta);}
59 }
60
61 \def\cdashline#1{\noalign{\ifnum0='}\fi
62          \@ifnextchar[%]
63 %          {\adl@cdline[#1]}%
64 %          {\adl@cdline[#1] [\dashlinedash/\dashlinegap]}
65 %          {\adl@inactivecdl[#1]}%
66 %          {\adl@inactivecdl[#1] [\dashlinedash/\dashlinegap]}}
67 }
68
69 \def\adl@inactivecdl[#1-#2] [#3]{\ifnum0='{\fi}\cline{#1-#2}}

```

File 25 **lwarf-asymptote.sty**

§ 121 Package **asymptote**

(Emulates or patches code by ANDY HAMMERLINDL, JOHN BOWMAN, TOM PRINCE.)

Pkg **asymptote** **asymptote** is patched for use by **lwarf**.

To compile:

```
pdflatex project.tex
asy project-*.asy
pdflatex project.tex
```

```
lwarpmk print
asy project-*.asy
lwarpmk print1
lwarpmk print1
```

```
lwarpmk html
asy project_html-*.asy
lwarpmk html1
lwarpmk html1
lwarpmk limages
```

for HTML output:

```
1 \LWR@ProvidesPackagePass{asympote}

2 \BeforeBeginEnvironment{asy}{\begin{lateximage}[(-asympote-\`{a}packagediagramname)]}
3 \AfterEndEnvironment{asy}{\end{lateximage}}
4
5 \xpatchcmd{\asyinclude}
6   {\begingroup}
7   {\begin{lateximage}[(-asympote-\`{a}packagediagramname)]}
8   {}
9   {\LWR@patcherror{asympote}{asyinclude-begingroup}}
10
11 \xpatchcmd{\asyinclude}
12   {\endgroup}
13   {\end{lateximage}}
14   {}
15   {\LWR@patcherror{asympote}{asyinclude-endgroup}}
```

File 26 **lwarf-atbegshi.sty**

§ 122 Package **atbegshi**

(Emulates or patches code by HEIKO OBERDIEK.)

Pkg **atbegshi** Emulated.

for HTML output: Discard all options for lwarf-atbegshi:

```
1 \LWR@ProvidesPackageDrop{atbegshi}[2011/10/05]
```

```

2 \newcommand*{\AtBeginShipout}[1]{}
3 \newbox\AtBeginShipoutBox
4 \newcommand*{\AtBeginShipoutNext}[1]{}
5 \newcommand*{\AtBeginShipoutFirst}[1]{}
6 \newcommand*{\AtBeginShipoutDiscard}(){}
7 \newcommand*{\AtBeginShipoutInit}){}
8 \newcommand*{\AtBeginShipoutAddToBox}[1]{}
9 \newcommand*{\AtBeginShipoutAddToBoxForeground}[1]{}
10 \newcommand*{\AtBeginShipoutUpperLeft}[1]{}
11 \newcommand*{\AtBeginShipoutUpperLeftForeground}[1]{}
12 \newcommand*{\AtBeginShipoutOriginalShipout}[1]{}
13 \def\AtBeginShipoutBoxWidth{0pt}
14 \def\AtBeginShipoutBoxHeight{0pt}
15 \def\AtBeginShipoutBoxDepth{0pt}
16

```

File 27 **lwarf-attachfile.sty**

§ 123 Package **attachfile**

(Emulates or patches code by SCOTT PAKIN.)

Pkg **attachfile** attachfile is patched for use by lwarf.

 Metadata is ignored for now.

for HTML output: 1 \LWR@ProvidesPackagePass{attachfile}

Encloses each icon:

```

2 \newenvironment*{\LWR@attachfile@icon}
3 {
4     \begin{lateximage} *%
5         [-attachfile-] %
6         [%
7             \detokenize\expandafter{\atfi@icon@icon}-%
8             \detokenize\expandafter{\atfi@color@rgb} %
9         ] %
10 }
11 {
12     \end{lateximage}
13 }

```

Each icon is enclosed inside a `\LWR@attachfile@icon` environment:

```

14 \xpretocmd{\atfi@acroGraph}{\LWR@attachfile@icon}{}{}
15 \xapptocmd{\atfi@acroGraph}{\endLWR@attachfile@icon}{}{}

```

```

16
17 \xpretocmd{\atfi@acroPaperclip}{\LWR@attachfile@icon}{}{}
18 \xapptocmd{\atfi@acroPaperclip}{\endLWR@attachfile@icon}{}{}
19
20 \xpretocmd{\atfi@acroPushPin}{\LWR@attachfile@icon}{}{}
21 \xapptocmd{\atfi@acroPushPin}{\endLWR@attachfile@icon}{}{}
22
23 \xpretocmd{\atfi@acroTag}{\LWR@attachfile@icon}{}{}
24 \xapptocmd{\atfi@acroTag}{\endLWR@attachfile@icon}{}{}

```

Disable PDF file embedding:

```
25 \DeclareRobustCommand{\atfi@embedfile}[1]{}
```

The displayed output for an \attachfile reference:

```

26 \newcommand*{\LWR@attachfile@appearance}(){}
27
28 \DeclareRobustCommand{\atfi@set@appearance}[1]{%
29     \def\LWR@attachfile@appearance{#1}%
30 }

```

A file annotation becomes a reference:

```

31 \DeclareRobustCommand{\atfi@insert@file@annot}[1]{%
32     \href{#1}{\LWR@attachfile@appearance}%
33 }

```

File 28 **l warp-attachfile2.sty**

§ 124 Package **attachfile2**

(Emulates or patches code by HEIKO OBERDIEK.)

Pkg **attachfile2** attachfile2 is patched for use by l warp.



Metadata is ignored for now.

for HTML output: 1 \LWR@ProvidesPackagePass{attachfile2}

Adds memory of the selected color:

```

2 \def\LWR@attachfiletwo@color{}%
3
4 \define@key{AtFi}{color}{%
5     \def\LWR@attachfiletwo@color{#1}%    l warp

```

```

6   \HyColor@AttachfileColor{#1}%
7       \atfi@color@tex\atfi@color@inline\atfi@color@annot
8           {attachfile2}{color}%
9 }

```

Encloses each icon:

```

10 \newenvironment*{LWR@attachfile@icon}
11 {
12     \begin{lateximage} *%
13         [-attachfile-]%
14         [%
15             \detokenize\expandafter{\atfi@icon@icon}-%
16             \detokenize\expandafter{\LWR@attachfiletwo@color}%
17         ]%
18 }
19 {
20     \end{lateximage}
21 }

```

Each icon is enclosed inside a LWR@attachfile@icon environment:

```

22 \xpretocmd{\atfi@acroGraph}{\LWR@attachfile@icon}{}{}
23 \xapptocmd{\atfi@acroGraph}{\endLWR@attachfile@icon}{}{}
24
25 \xpretocmd{\atfi@acroPaperclip}{\LWR@attachfile@icon}{}{}
26 \xapptocmd{\atfi@acroPaperclip}{\endLWR@attachfile@icon}{}{}
27
28 \xpretocmd{\atfi@acroPushPin}{\LWR@attachfile@icon}{}{}
29 \xapptocmd{\atfi@acroPushPin}{\endLWR@attachfile@icon}{}{}
30
31 \xpretocmd{\atfi@acroTag}{\LWR@attachfile@icon}{}{}
32 \xapptocmd{\atfi@acroTag}{\endLWR@attachfile@icon}{}{}

```

Disable PDF file embedding:

```
33 \DeclareRobustCommand{\atfi@embedfile}[1]{}
```

The displayed output for an \attachfile reference:

```

34 \newcommand*{\LWR@attachfile@appearance}{}%
35
36 \def\atfi@set@appearance@icon{%
37     \atfi@set@appearance{\csname atfi@acro\atfi@icon@icon\endcsname}%
38 }
39
40 \DeclareRobustCommand{\atfi@set@appearance}[1]{%
41     \def\LWR@attachfile@appearance{#1}%
42 }

```

A file annotation becomes a reference:

```
43 \DeclareRobustCommand{\atfi@insert@file@annot}[1]{%
44     \href{#1}{\LWR@attachfile@appearance}%
45 }
```

Modified for text color:

```
46 \DeclareRobustCommand{\notextattachfile}[2][]{%
47     \begingroup
48     \atfi@setup{#1}%
49     \ifatfi@print
50         \leavevmode
51         \begingroup
52             \HyColor@UseColor\atfi@color@tex
53             \LWR@textcurrentcolor{#2}%
54             l warp
54 \% \strut
55         \endgroup
56 \% \else
57 \%     \sbox\ltx@zero{#2\strut}%
58 \%     \makebox[\wd0]{}%
59     \fi
60 \endgroup
61 }
```

Modified to draw the icon:

```
62 \DeclareRobustCommand{\noattachfile}[1][]{%
63     \begingroup
64     \atfi@setup{#1}%
65     \atfi@set@appearance@icon
66     \ifatfi@print
67         \LWR@attachfile@appearance%
68         l warp
69 \% \expandafter
69 \% \atfi@refxform\csname atfi@appobj@\atfi@icon@icon\endcsname
70 \% \else
71 \%     \makebox[\atfi@appearancewidth]{}%
72     \fi
73 \endgroup
74 }
```

File 29 **l warp-authblk.sty**

§ 125 Package **authblk**

(Emulates or patches code by PATRICK W. DALY.)

Pkg authblk authblk is patched for HTML.

package support l warp supports the native L^AT_EX titling commands, and also supports the packages authblk and titling. If both are used, authblk should be loaded before titling.

⚠ load order If using the titling package, additional titlepage fields for \published and \subtitle may be added by using \AddSubtitlePublished in the preamble. See section 65.8.

(Emulates or patches code by PATRICK W. DALY.)

for HTML output: Require that authblk be loaded before titling:

```
1 \@ifpackageloaded{titling}{
2 \PackageError{l warp-authblk}
3 {Package authblk must be loaded before titling}
4 {Titling appends authblk's author macro, so authblk must be loaded first.}
5 }
6 {}
```

Load authblk:

```
7 \LWR@ProvidesPackagePass{authblk}
```

Patch to add a class for the affiliation:

```
8 \LetLtxMacro\LWRAB@affil\affil
9
10 \renewcommand{\affil}[2][]{%
11 \LWRAB@affil[#1]{\protect\InlineClass{affiliation}{#2}}%
12 }
```

Create an HTML break for an \authorcr:

```
13 \renewcommand*{\authorcr}{\protect\LWR@newlinebr}
```

File 30 l warp-axessibility.sty

§ 126 Package **axessibility**

Pkg axessibility axessibility is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{axessibility}
2 \newcommand{\wrap}[1]{}{}
```

File 31 **l warp-axodraw2.sty**

§ 127 Package **axodraw2**

(Emulates or patches code by JOHN C. COLLINS, J.A.M. VERMASEREN.)

Pkg **axodraw2** axodraw2 is patched for use by l warp.

for HTML output: 1 \LWR@ProvidesPackagePass{axodraw2}

```
2 \BeforeBeginEnvironment{axopicture}{\begin{lateximage}[(-axopicture-~\packagediagramname)]}
3
4 \AfterEndEnvironment{axopicture}{\end{lateximage}}
```

File 32 **l warp-backref.sty**

§ 128 Package **backref**

(Emulates or patches code by DAVID CARLISLE AND SEBASTIAN RAHTZ.)

Pkg **backref** backref is patched for use by l warp.

⚠ loading Note that backref must be explicitly loaded, and is not automatically loaded by hyperref when generating HTML output.

for HTML output: 1 \LWR@ProvidesPackagePass{backref}

Force the hyperref option:

```
2 \def\backref{} \let\backrefxxx\hyper@section@backref
```

File 33 **l warp-balance.sty**

§ 129 Package **balance**

(Emulates or patches code by PATRICK W. DALY.)

Pkg **balance** Emulated.

for HTML output:

Discard all options for l warp-balance:

```

1 \LWR@ProvidesPackageDrop{balance}
2 \newcommand*{\balance}{}%
3 \newcommand*{\nobalance}{}%

```

File 34 **l warp-biblatex.sty**

§ 130 Package **biblatex**

(Emulates or patches code by PHILIPP LEHMAN.)

Pkg **biblatex** When biblatex is used, modifications from newfloat may have to be undone.

for HTML output:

1. l warp uses newfloat.
2. For classes with chapters which newfloat does not know about, such as CTEX-related classes, newfloat may modify \addtocontents.
3. biblatex, though, wants to patch \addtocontents, which causes an error if \addtocontents has been changed.
4. Therefore, \addtocontents is restored to its original here, since biblatex is about to be loaded.
5. This means that the newfloat's chapterlistsgaps option may no longer work.

```

1 \ifdef{\newfloat@addtocontents@ORI}{%
2   \let\addtocontents\newfloat@addtocontents@ORI
3 }{}%
4 \LWR@ProvidesPackagePass{biblatex}

```

File 35 **l warp-bibunits.sty**

§ 131 Package **bibunits**

(Emulates or patches code by THORSTEN HANSEN.)

Pkg **bibunits** bibunits is patched for use by l warp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{bibunits}
```

```
2 \def\bu@bibdata{\BaseJobname}
```

File 36 **l warp-bigdelim.sty**

§ 132 Package **bigdelim**

(Emulates or patches code by PIET VAN OOSTRUM, ØYSTEIN BACHE, JERRY LEICHTER.)

Pkg **bigdelim** **bigdelim** is used as-is for print or `lateximage`, and patched for HTML.

The delimiters are displayed in HTML by printing the delimiter, the text, and a thick border across the side of the `\multirow` which indicates the actual height of the delimiter. The delimiter character is given a `` class of `l delim` or `r delim`, and the default css sets this to `font-size:200%`

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

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

<>	a	b
left	$\left\{ \begin{array}{cc} c & d \\ e & f \end{array} \right.$	
<>	g	h

for HTML output: First, remove the temporary definitions of `\l delim` and `\r delim`, which were previously defined for tabular scanning in case `bigdelim` was not loaded:

```
1 \let\l delim\relax
2 \let\r delim\relax
```

Next, load the package's new definitions:

```
3 \LWR@ProvidesPackagePass{bigdelim}

\l delim {\{1:delimiter\}} {\{2:#rows\}} {\{3:width\}} [{4:text}]
\r delim
    4 \NewDocumentCommand{\LWR@HTML@l delim}{m m m O{} }{%
```

```

5 \renewcommand{\LWR@multirowborder}{right}%
6 \multirow{#2}{#3}{#4 \InlineClass{ldelim}{#1}}%
7 }%
8 %
9 \LWR@formatted{ldelim}%
10 %
11 \NewDocumentCommand{\LWR@HTML@rdelim}{m m m O{} }{%
12 \renewcommand{\LWR@multirowborder}{left}%
13 \multirow{#2}{#3}{\InlineClass{rdelim}{#1} #4}%
14 }%
15 %
16 \LWR@formatted{rdelim}%

```

File 37 **lwarp-bigstrut.sty**

§ 133 Package **bigstrut**

(Emulates or patches code by PIET VAN OOSTRUM, ØYSTEIN BACHE, JERRY LEICHTER.)

Pkg **bigstrut** **bigstrut** is used as-is for print or `lateximage`, and patched for HTML.

for HTML output: 1 \LWR@ProvidesPackagePass{bigstrut}

```

2 \LetLtxMacro{\LWR@origbigstrut}{\bigstrut}
3 %
4 \renewcommand{\bigstrut}[1][x]{}
5 %
6 \appto{\LWR@restoreorigformatting}{%
7 \LetLtxMacro{\bigstrut}{\LWR@origbigstrut}%
8 }%

```

File 38 **lwarp-bitpattern.sty**

§ 134 Package **bitpattern**

(Emulates or patches code by JEAN-MARC BOURGUET.)

Pkg **bitpattern** **bitpattern** is patched for use by `lwarp`.

for HTML output: 1 \LWR@ProvidesPackagePass{bitpattern}

```

2 \xpatchcmd{\bitpattern}{%
3   {\begingroup}%
4   {\begin{lateximage}[-bitpattern-\~\packagediagramname]}}

```

```

5      {}
6      {\LWR@patcherror{bitpattern}{bitpattern}}
7
8 \xpatchcmd{\bp@Done}
9   {\endgroup}
10  {\end{lateximage}}
11  {}
12  {\LWR@patcherror{bitpattern}{bp@Done}}

```

File 39 **lwarf-blowup.sty**

§ 135 Package **blowup**

Pkg **blowup** **blowup** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{blowup}

2 \newcommand*\blowUp[1]{}

```

File 40 **lwarf-booklet.sty**

§ 136 Package **booklet**

(Emulates or patches code by PETER WILSON.)

Pkg **booklet** **booklet** is nullified.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{booklet}[2009/09/02]

2 \newdimen\pageseplength
3 \newdimen\pagesepwidth
4 \newdimen\pagesepoffset
5 \newif\ifsidebyside    \sidebysidetrue
6 \newif\ifuselandscape \uselandscapefalse
7 \newif\ifprintoption  \printoptionfalse
8 \newcommand*\{\pagespersignature}[1]{}
9 \def\magstepminus#1{}
10 \newcommand*\{\target}[3]{}
11 \newcommand*\{\source}[3]{}
12 \newcommand*\{\setpdftargetpages}(){}
13 \newcommand*\{\setdvipstargetpages}){}
14 \newcommand*\{\targettopbottom}){}
15 \newcommand*\{\twoupemptypage}){}

```

```
16 \newcommand*{\twoupclearpage}{}  
17 \newcommand*{\checkforlandscape}{}  


---


```

File 41 **l warp-bookmark.sty**

§ 137 Package **bookmark**

(Emulates or patches code by HEIKO OBERDIEK.)

Pkg **bookmark** bookmark is emulated.

for HTML output: Discard all options for l warp-bookmark:

```
1 \LWR@ProvidesPackageDrop{bookmark}  
  
2 \newcommand*{\bookmarksetup}[1]{}  
3 \newcommand*{\bookmarksetupnext}[1]{}  
4 \newcommand*{\bookmark}[2][]{  
5 \newcommand*{\bookmarkdefinestyle}[2]{}  
6 \newcommand*{\bookmarkget}[1]{}  
7 \newcommand{\BookmarkAtEnd}[1]{}
```

File 42 **l warp-booktabs.sty**

§ 138 Package **booktabs**

(Emulates or patches code by SIMON FEAR.)

Pkg **booktabs** booktabs is emulated during HTML output, and used as-is during print output and inside an HTML `lateximage`.

for HTML output: First, forget the placeholder macros:

```
1 \LetLtxMacro{\toprule}{\relax}  
2 \LetLtxMacro{\midrule}{\relax}  
3 \LetLtxMacro{\cmidrule}{\cline}  
4 \LetLtxMacro{\bottomrule}{\relax}  
5 \LetLtxMacro{\addlinespace}{\relax}  
6 \LetLtxMacro{\morecmidrules}{\relax}  
7 \LetLtxMacro{\specialrule}{\relax}  
8  
9 \LWR@ProvidesPackagePass{booktabs}
```

```
10 \DeclareDocumentCommand{\LWR@HTML@toprule}{o d()}%
```

```
11   {%
12     \IfValueTF{#1}%
13       {\LWR@docmidrule[#1] (){1-\arabic{LWR@tabletotalLaTeXcols}}}{%
14       {%
15         \ifbool{FormatWP}{%
16           {\LWR@docmidrule[#1] (){1-\arabic{LWR@tabletotalLaTeXcols}}}{%
17             {\global\booltrue{LWR@doingtbrule}}}{%
18           }{%
19             \LWR@getmynexttoken}%
20           }%
21 \LWR@expandableformatted{toprule}%
22
23 \DeclareDocumentCommand{\LWR@HTML@midrule}{o d()}{%
24   {%
25     \IfValueTF{#1}{%
26       {\LWR@docmidrule[#1] (){1-\arabic{LWR@tabletotalLaTeXcols}}}{%
27       {%
28         \ifbool{FormatWP}{%
29           {\LWR@docmidrule[#1] (){1-\arabic{LWR@tabletotalLaTeXcols}}}{%
30             {\addtocounter{LWR@hlines}{1}}}{%
31           }{%
32             \LWR@getmynexttoken}%
33           }%
34 \LWR@expandableformatted{midrule}%
35
36 \DeclareDocumentCommand{\LWR@HTML@cmidrule}{O{\LWR@cmidrulewidth} d() m}{%
37   {\LWR@docmidrule[#1] (#2){#3}}%
38   \LWR@getmynexttoken}%
39 }{%
40
41 \LWR@expandableformatted{cmidrule}%
42
43 \DeclareDocumentCommand{\LWR@HTML@bottomrule}{o d()}{%
44   {%
45     \IfValueTF{#1}{%
46       {\LWR@docmidrule[#1] (){1-\arabic{LWR@tabletotalLaTeXcols}}}{%
47       {%
48         \ifbool{FormatWP}{%
49           {\LWR@docmidrule[#1] (){1-\arabic{LWR@tabletotalLaTeXcols}}}{%
50             {\global\booltrue{LWR@doingtbrule}}}{%
51           }{%
52             \LWR@getmynexttoken}%
53           }{%
54 \LWR@expandableformatted{bottomrule}%
55
56 \DeclareDocumentCommand{\LWR@HTML@addlinespace}{o}{%
57
58 \LWR@expandableformatted{addlinespace}%
59
60 \DeclareDocumentCommand{\LWR@HTML@morecmidrules}{}{}}
```

```
61
62 \LWR@expandableformatted{morecmidrules}
63
64 \DeclareDocumentCommand{\LWR@HTML@specialrule}{m m m d()}{%
65     {\LWR@docmidrule[#1] (){1-\arabic{LWR@tabletotalLaTeXcols}}\LWR@getmynexttoken}%
66
67 \LWR@expandableformatted{specialrule}
```

File 43 **l warp-bophook.sty**

§ 139 Package **bophook**

Pkg bophook bophook is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{bophook}

2 \newcommand*{\AtBeginPage}[1]{}
3 \newcommand*{\PageLayout}[1]{}
```

File 44 **l warp-bounddvi.sty**

§ 140 Package **bounddvi**

Pkg bounddvi bounddvi is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{bounddvi}
```

File 45 **l warp-boxedminipage2e.sty**

§ 141 Package **boxedminipage2e**

(Emulates or patches code by SCOTT PAKIN.)

Pkg boxedminipage2e boxedminipage2e is emulated.

for HTML output: Discard all options for l warp-boxedminipage2e:

```
1 \LWR@ProvidesPackageDrop{boxedminipage2e}

2 \newenvironment{boxedminipage}{%
3 \begin{BlockClass}{framebox}}%
```

```
4 \minipage%
5 }
6 {
7 \endminipage%
8 \end{BlockClass}
9 }
```

File 46 l warp-breakurl.sty**§ 142 Package breakurl**

(Emulates or patches code by VILAR CAMARA NETO.)

Pkg breakurl breakurl is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{breakurl}

```
2 \LetLtxMacro\burl\url
3
4 \NewDocumentCommand{\LWR@burlaltb}{O{} +m m}{%
5 \LWR@ensuredoingapar%
6 \LWR@subhyperref{#2}%
7 \LWR@subhyperreftext{#3}%
8 \endgroup% restore catcodes
9 }
10
11 \newrobustcmd*\burlalt{%
12 \begingroup%
13 \catcode`\#=12%
14 \catcode`\%=12%
15 \catcode`\&=12%
16 \catcode`\~=12%
17 \catcode`\_=12%
18 \LWR@burlaltb%
19 }
20
21 \LetLtxMacro\urlalt\burlalt
```

File 47 l warp-breqn.sty**§ 143 Package breqn**

(Emulates or patches code by MICHAEL J. DOWNES, MORTEN HØGHLØM.)

Pkg breqn breqn is patched for use by l warp.

⚠️ darray darray is not supported.

⚠️ MATHJAX MathJax does not support breqn.

for HTML output:

```
1 \LWR@ProvidesPackagePass{breqn}

2 \setkeys{breqn}{spread={5pt}}
3
4 \def\eqnumside{R}
5 % \def\eqnumplace{T}
6
7 \BeforeBeginEnvironment{dmath}{
8     \begin{BlockClass}{displaymathnumbered}
9         \LWR@newautoidanchor%
10        \booltrue{\LWR@indisplaymathimage}%
11        \begin{lateximage}[-breqn dmath- \mathimagename]
12    }
13
14 \AfterEndEnvironment{dmath}{
15     \end{lateximage}\end{BlockClass}
16 }
17
18 \BeforeBeginEnvironment{dmath*}{
19     \begin{BlockClass}{displaymath}
20         \LWR@newautoidanchor%
21        \booltrue{\LWR@indisplaymathimage}%
22        \begin{lateximage}[-breqn dmath*- \mathimagename]
23    }
24
25 \AfterEndEnvironment{dmath*}{
26     \end{lateximage}\end{BlockClass}
27 }
28
29 \BeforeBeginEnvironment{dseries}{
30     \begin{BlockClass}{displaymathnumbered}
31         \LWR@newautoidanchor%
32        \booltrue{\LWR@indisplaymathimage}%
33        \begin{lateximage}[-breqn dseries- \mathimagename]
34    }
35
36 \AfterEndEnvironment{dseries}{
37     \end{lateximage}\end{BlockClass}
38 }
39
40 \BeforeBeginEnvironment{dseries*}{
41     \begin{BlockClass}{displaymath}
42         \LWR@newautoidanchor%
43        \booltrue{\LWR@indisplaymathimage}%
```

```

44     \begin{lateximage}[-breqn dseries*- \mathimagename]
45 }
46
47 \AfterEndEnvironment{dseries*}{
48     \end{lateximage}\end{BlockClass}
49 }
50
51 \BeforeBeginEnvironment{dgroup}{
52     \begin{BlockClass}{displaymath}
53     \LWR@newauto{danchor}%
54     \booltrue{\LWR@indisplaymathimage}%
55     \begin{lateximage}[-breqn dgroup- \mathimagename]
56 }

57 \AfterEndEnvironment{dgroup}{
58     \end{lateximage}\end{BlockClass}
59 }
60 }

61 \BeforeBeginEnvironment{dgroup*}{
62     \begin{BlockClass}{displaymath}
63     \LWR@newauto{danchor}%
64     \booltrue{\LWR@indisplaymathimage}%
65     \begin{lateximage}[-breqn dgroup*- \mathimagename]
66     \end{lateximage}\end{BlockClass}
67 }

68 \AfterEndEnvironment{dgroup*}{
69     \end{lateximage}\end{BlockClass}
70 }
71 }
```

File 48 l warp-bsheaders.sty

§ 144 Package **bsheaders**

Pkg bsheaders bsheaders is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{bsheaders} [1997/10/06]

File 49 l warp-bx papersize .sty

§ 145 Package **bxpapersize**

Pkg bxpapersize bxpapersize is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{bxpapersize}

```

2 \providecommand*\papersizesetup{\bpxpapersizesetup}
3 \newcommand*\bpxpapersizesetup[1]{}

```

File 50 **l warp-bytefield.sty**

§ 146 Package **bytefield**

(Emulates or patches code by SCOTT PAKIN.)

Pkg bytefield bytefield is patched for use by l warp.

for HTML output:

```

1 \LWR@ProvidesPackagePass{bytefield}

2 \BeforeBeginEnvironment{bytefield}{\begin{lateximage}[(-bytefield-\~\packagediagramname)]}
3
4 \AfterEndEnvironment{bytefield}{\end{lateximage}}

```

File 51 **l warp-cancel.sty**

§ 147 Package **cancel**

Pkg cancel cancel is used as-is for SVG math, and emulated for HTML text output.

for HTML output:

```

1 \LWR@origRequirePackage{l warp-xcolor}%
2 \LWR@ProvidesPackagePass{cancel}

```

\cancelto is math-only, so is used as-is.

```

3 \LetLtxMacro{\LWR@origcancel}{cancel}
4 \LetLtxMacro{\LWR@origbcancel}{bcancel}
5 \LetLtxMacro{\LWR@origxcancel}{xcancel}
6
7 \appto{\LWR@restoreorigformatting}{%
8 \LetLtxMacro{\cancel}{\LWR@origcancel}%
9 \LetLtxMacro{\bcancel}{\LWR@origbcancel}%
10 \LetLtxMacro{\xcancel}{\LWR@origxcancel}%
11 }

```

\LWR@cancelcolor {{text}} {{color}} {{class}} {{colorstyle}} {{FormatWPstyle}}

Add colors if not empty:

```

12 \newcommand{\LWR@cancelcolor}[5]{%

```

```

13 \ifcsempty{#2}%
14 {\InlineClass{#5}{#3}{#1}}%
15 {\LWR@htmlspanclass[#5;#4:\LWR@origpound\LWR@tempcolor]{#3}{#1}}%
16 }

\cancel {<text>}
17 \DeclareRobustCommand{\cancel}[1]{%
18 \begingroup%
19 \CancelColor%
20 \LWR@findcurrenttextcolor%
21 \color{black}%
22 \LWR@cancelcolor{#1}{\LWR@tempcolor}{sout}{text-decoration-color}%
23     {text-decoration:line-through}%
24 \endgroup%
25 }
26
27 \LetLtxMacro\bcancel\cancel
28 \LetLtxMacro\xcancel\cancel

```

File 52 **l warp-canoniclayout.sty**

§ 148 Package **canoniclayout**

Pkg canoniclayout canoniclayout is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{canoniclayout}

2 \newcommand*{\currentfontletters}{}%
3 \newcommand*{\charactersperpage}{}%

```

File 53 **l warp-caption.sty**

§ 149 Package **caption**

(Emulates or patches code by AXEL SOMMERFELDT.)

Pkg caption caption is patched for use by l warp.

for HTML output:

```

1 \LWR@ProvidesPackagePass{caption}

2 \renewcommand\caption@ibox[3]{%
3   \testopt{\caption@ibox{#1}{#2}{#3}}{%
4     \wd\@tempboxa}%

```

```
5      \linewidth% l warp
6  }%
7 % \LWR@traceinfo{caption@ibox: done}%
8 }

9 \long\def\caption@iiibox#1#2#3[#4]{%
10   \testopt{\caption@iiibox{#1}{#2}{#3}{#4}}\captionbox@hj@default
11 }

12 \long\def\caption@iiibox#1#2#3#4[#5]{%
13 %   \setbox\@tempboxa\hbox{#6}%
14   \begingroup
15   #1% set \caption@position
16   \caption@iftop{%
17     \LWR@traceinfo{\caption@iiibox top}%
18     \endgroup
19     \parbox[t]{#4}{%
20       #1\relax
21       \caption@setposition t%
22 %         \vbox{\caption#2{#3}}%
23         {\caption#2{#3}}% l warp
24 %         \captionbox@hrule
25 %         \csname caption@hj@#5\endcsname
26 %         \unhbox\@tempboxa
27         #6% l warp
28       }%
29     }{%
30       \LWR@traceinfo{\caption@iiibox bottom}%
31     \endgroup
32     \parbox[b]{#4}{%
33       #1\relax
34       \caption@setposition b%
35 %         \csname caption@hj@#5\endcsname
36 %         \unhbox\@tempboxa
37         #6% l warp
38 %         \captionbox@hrule
39 %         \vtop{\caption#2{#3}}%
40         {\caption#2{#3}}% l warp
41       }%
42     }%
43 \LWR@traceinfo{caption@iiibox: done}%
44 }

45 \def\caption@caption{%
46   \caption@iftype
47   {%
48     \caption@checkgrouplevel\@empty\caption
49     \caption@star
50     {\caption@refstepcounter\@capttype}%
51   }
```

```
52      {\caption@dblarg{\@caption\@capttype}}}%
53      {\caption@Error{\noexpand\caption outside float}%
54      \caption@gobble}%
55 }
56
57 \long\def\caption@@caption#1[#2]#3{%
58   \ifcaption@star \else
59     \caption@prepareanchor{#1}{#2}%
60     \memcaptioninfo{#1}{\csname the#1\endcsname}{#2}{#3}%
61     \cnameuse{nag@hascaptiontrue}%
62   \fi
63
64   \par
65   \caption@beginex{#1}{#2}{#3}%
66   \caption@setfloatcapt{%
67     \caption@boxrestore
68     \if@minipage
69       \setminipage
70     \fi
71     \caption@normalsize
72     \ifcaption@star
73       \let\caption@makeanchor@\firstofone
74     \fi
75     \makecaption{\csname fnum@\#1\endcsname}%
76     {\ignorespaces\caption@makeanchor{#3}}\par
77   \caption@if@minipage{\minipagetrue}{\minipagefalse}%
78 }%
79
80 \caption@@make {<caption label>} {<caption text>}
81 \renewcommand\caption@@make[2]{%
82 \LWR@startpars% lwarp
83 % \sbox{\tempboxa{#1}%
84 % \ifdim\wd\tempboxa=\z@
85 % \let\caption@lsep\relax
86 % \fi
87 % \caption@ifempty{#2}{%
88 %   \let\caption@lsep\empty
89 %   \let\caption@tfmt\firstofone
90 % }%
91 % \setpar{\LWR@closeparagraph\@@par}% lwarp
92 \caption@applyfont
93 \caption@fmt
94   {\ifcaption@star\else
95     \begingroup
96       \captionlabelfont
97       #1%
```

```

96      \endgroup
97      \fi}%
98 {\ifcaption@star\else
99   \begingroup
100    \caption@iflf\captionlabelfont
101    \relax\caption@lsep
102   \endgroup
103 \fi}%
104 {{\captiontextfont
105   \caption@ifstrut
106   {\vrule\@height\ht\strutbox\@width\z@\%
107   \}}%
108   \nobreak\hskip\z@skip % enable hyphenation
109   \caption@tfmt{\#2}
110 \LWR@ensuredoingapar% l warp
111   \caption@ifstrut
112   {\ifhmode\@finalstrut\strutbox\fi}%
113   \}}%
114   \par}}
115 \LWR@stoppars% l warp
116 }

\caption@@make@ {{()}{()}}
117 \renewcommand{\caption@@make@}[2]{%
118   \caption@stepthecounter
119   \caption@beginhook
120   \caption@@make{\#1}{\#2}%
121   \caption@endhook
122 }

123 % \DeclareCaptionBox{none}{\#2}
124 \DeclareCaptionBox{parbox}{\#2}%
125 \#2%
126 }
127 \DeclareCaptionBox{colorbox}{\#2}%
128 \#2%
129 }

```

File 54 **l warp-cases.sty**

§ 150 Package **cases**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg cases cases is patched for use by l warp.

 **MATHJAX** MathJax does not support cases.

for HTML output:

```

1 \LWR@ProvidesPackagePass{cases}

2 \BeforeBeginEnvironment{numcases}{
3     \begin{BlockClass}{displaymathnumbered}
4         \LWR@newautoidanchor%
5         \booltrue{LWR@indisplaymathimage}%
6         \begin{lateximage}[-cases- \mathimagename]
7     }
8
9 \AfterEndEnvironment{numcases}{
10    \end{lateximage}\end{BlockClass}
11 }

12 \BeforeBeginEnvironment{subnumcases}{
13     \begin{BlockClass}{displaymathnumbered}
14         \LWR@newautoidanchor%
15         \booltrue{LWR@indisplaymathimage}%
16         \begin{lateximage}[-cases- \mathimagename]
17     }
18 }
19
20 \AfterEndEnvironment{subnumcases}{
21     \end{lateximage}\end{BlockClass}
22 }
```

File 55 **l warp-changebar.sty**

§ 151 Package **changebar**

Pkg **changebar** **changebar** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{changebar} [2018/03/09]

2 \newcommand*\{\cbstart\}{}%
3 \newcommand*\{\cbend\}{}%
4 \newenvironment*\{\changebar\}{}{}%
5 \newcommand*\{\cbdelete\}{}%
6 \newcommand*\{\nochnagebars\}{}%
7 \newcommand*\{\cbccolor\}[1]{}%
8 \newlength{\changebarwidth}%
9 \newlength{\deletebarwidth}%
10 \newlength{\changebarsep}%
11 \newcounter{changebargrey}
```

File 56 **l warp-changepage.sty**

§ 152 Package **changepage**

(Emulates or patches code by PETER WILSON.)

Pkg changepage changepage is emulated.

for HTML output: Discard all options for l warp-changepage:

```
1 \LWR@ProvidesPackageDrop{changepage}

2 \newif\ifoddpage
3 \DeclareRobustCommand{\checkoddpage}{\oddpage=true}
4 \DeclareRobustCommand{\chagetext}[5]{}
5 \DeclareRobustCommand{\changepage}[9]{}
6
7 \@ifundefined{adjustwidth}{}
8 \newenvironment{adjustwidth}[2]{}{}
9 \newenvironment{adjustwidth*}[2]{}{}
10 }{
11 \renewenvironment{adjustwidth}[2]{}{}
12 \renewenvironment{adjustwidth*}[2]{}{}
13 }

14 \DeclareDocumentCommand{\strictpagecheck}{}{ }
15 \DeclareDocumentCommand{\easypagecheck}{}{ }
```

File 57 **l warp-chngpage.sty**

§ 153 Package **chngpage**

(Emulates or patches code by PETER WILSON.)

Pkg chngpage chngpage is emulated.

for HTML output: Discard all options for l warp-chngpage:

```
1 \LWR@ProvidesPackageDrop{chngpage}
2 \LWR@origRequirePackage{l warp-changepage}
```

File 58 **lwarf-chappg.sty**

§ 154 Package **chappg**

(Emulates or patches code by ROBIN FAIRBAIRNS.)

Pkg chappg chappg is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{chappg}
2 \renewcommand{\pagenumbering}{[2] [] {}}
3 \providecommand{\chappgsep}{--}
```

File 59 **lwarf-chapterbib.sty**

§ 155 Package **chapterbib**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg chapterbib chapterbib is patched for use by lwarf.

for HTML output:

```
1 \LWR@ProvidesPackagePass{chapterbib}
2 \xdef\@savedjobname{\BaseJobname}
3 \let\@currentipfile\@savedjobname
```

File 60 **lwarf-chemfig.sty**

§ 156 Package **chemfig**

(Emulates or patches code by CHRISTIAN TELLECHEA.)

Pkg chemfig chemfig is patched for use by lwarf.

If using `\polymerdelim` to add delimiters to a `\chemfig`, wrap both inside a single `lateXimage`:

```
\begin{lateXimage}[(-chemfig-\packagediagramname)]
\chemfig{...}
\polymerdelim[...]{...}
\end{lateXimage}
```

The images are not hashed because they depend on external settings which may be changed at any time, and are unlikely to be reused inline anyhow.

for HTML output:

```
1 \LWR@ProvidesPackagePass{chemfig}
2 \LetLtxMacro{\LWR@chemfig}{\origchemfig}\chemfig
3
4 \DeclareDocumentCommand{\chemfig}{s O{} O{} m}{%
5   \begin{ lateximage } [(-chemfig--\packagediagramname)]{%
6     \IfBooleanTF{#1}{%
7       \LWR@chemfig*[#2][#3]{#4}%
8     }{%
9       \LWR@chemfig@origchemfig[#2][#3]{#4}%
10    }%
11   \end{ lateximage }%
12 }
13
14 \LetLtxMacro{\LWR@chemfig}{\origCF@lewis@b}\CF@lewis@b
15
16 \def\CF@lewis@b#1#2{%
17 \begin{ lateximage } [(-chemfig--\packagediagramname)]{%
18 \LWR@chemfig@origCF@lewis@b{#1}{#2}%
19 \end{ lateximage }%
20 }
21
22 \preto{\schemestart}{\begin{ lateximage } [(-chemfig--\packagediagramname)]{%
23 \appto{\CF@schemestop}{\end{ lateximage }}%
24
25 \LetLtxMacro{\LWR@chemfig}{\origchemleft}\chemleft
26
27 \def\chemleft#1#2\chemright#3{%
28 \begin{ lateximage } [(-chemfig--\packagediagramname)]{%
29 \LWR@chemfig@origchemleft#1#2\chemright#3}%
30 \end{ lateximage }%
31 }
32
33 \LetLtxMacro{\LWR@chemfig}{\origchemup}\chemup
34
35 \def\chemup#1#2\chemdown#3{%
36 \begin{ lateximage } [(-chemfig--\packagediagramname)]{%
37 \LWR@chemfig@origchemup#1#2\chemdown#3}%
38 \end{ lateximage }%
39 }
```

File 61 **l warp-chemformula.sty**

§ 157 Package **chemformula**

(Emulates or patches code by CLEMENS NIEDERBERGER.)

Pkg **chemformula** **chemformula** is patched for use by **l warp**.

The SVG images are hashed according to contents and local options. Global options are assumed to be constant document-wide.

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

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

for HTML output:

```

1 \LWR@ProvidesPackagePass{chemformula}[2017/03/23]

2 \ExplSyntaxOn

```

`\ch` Enclose in an inline SVG image or MathJax. The `alt` tag is the contents of the `\ch` expression. The filename is hashed, and also has additional hashing information based on the local options.

```

3 \RenewDocumentCommand \ch { O{}m }
4   {%

```

To work inside `align` with `\displaymathother`, a simple version must be used to work with `chemformula`'s adaptation to `align`.

```

5   \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
6   {
7     \chemformula_ch:nn {#1} {#2}%
8   }%

```

If used as the outer level, must temporarily ensure MATHJAX is disabled:

```

9   {
10   \begingroup%
11   \boolfalse{mathjax}%

```

An inline image is used, adjusted for the baseline:

```

12      \LWR@subsingle$*{%
13          \textbackslash ch\{\LWR@HTMLsanitize{#2}\}%
14      }%
15      \protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}%
16  }%
17  {%
18      \chemformula_ch:nn {#1} {#2}%
19  }%
20  \endgroup%
21 }
22 }
```

\chcpd Similar to \ch.

```

23 \cs_gset_protected:Npn \chemformula_chcpd:nn #1#2
24 {
25     \begingroup%
26     \boolfalse{mathjax}%
27     \LWR@subsingle$*{%
28         \textbackslash chcpd\{\LWR@HTMLsanitize{#2}\}%
29     }%
30     \protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}%
31 }%
32 {%
33     \group_begin:
34     \tl_if_blank:nF {#2}
35     {
36         \keys_set:nn {chemformula} {#1}
37         \__chemformula_save_catcodes:
38         \__chemformula_sanitize:Nn
39             \l__chemformula_chemformula_tmpa_tl
40             {#2}
41         \__chemformula_input_compound_no_check:NV
42             \l__chemformula_compound_tl
43             \l__chemformula_chemformula_tmpa_tl
44             \__chemformula_prepare_output:N \l__chemformula_compound_tl
45             \chemformula_write:V \l__chemformula_compound_tl
46     }
47     \group_end:
48 }
49 }
```

\charrow If standalone, appears in a regular *lateximage*.

```

50 \RenewDocumentCommand \charrow { mO{}O{} } %
51 {
52     \begin{latextimage}[(-chemformula- charrow)]
53     \group_begin:
54         \__chemformula_draw_arrow:nnn {#1} {#2} {#3}
```

```

55     \group_end:
56     \end{lateximage}
57 }
```

\chname If standalone, appears in a regular `lateximage`, hashed according to contents.

```

58 \RenewDocumentCommand \chname { R(){}R(){} }
59 {
60     \begin{lateximage}*[%
61         \textbackslash chname(\LWR@HTMLsanitize{\#1})(\LWR@HTMLsanitize{\#2})
62     ]%
63         \chemformula_chwritebelow:nn {\#1} {\#2}
64     \end{lateximage}
65 }
```

\chlewis Placed inline, hashed according to contents and options.

```

66 \RenewDocumentCommand \chlewis { O{}mm }
67 {
68     \begingroup%
69     \boolfalse{mathjax}%
70     \LWR@singledollar*\textbackslash chlewis\{\#2\}\{\#3\}%
71     {
72         \protect\LWR@HTMLsanitize{\detokenize\expandafter{\#1}}%
73     }%
74         \chemformula_lewis:nnn {\#1} {\#2} {\#3}
75     }
76     \endgroup%
77 }
```

\l warp redefines the \$ character, so special handling is required to escape math expressions inside \ch.

This boolean tracks a new kind of escaped math:

```
78 \bool_new:N      \l__chemformula_first_last_LWRdollar_bool
```

\chemformula_input_escape_math

Adds additional escaping for the new dollar definition:

```

79 \cs_gset_protected:Npn \__chemformula_input_escape_math:n #1
80 {
81     \__chemformula_first_last_math:n {\#1}
82     \bool_if:NT \l__chemformula_first_last_dollar_bool
83     {
84         \bool_set_true:N \l__chemformula_first_last_math_bool
85         \__chemformula_read_escape_dollar:w #1 \q_nil
```

```

86      }
87      \bool_if:NT \l_chemformula_first_last_mathbraces_bool
88      {
89          \bool_set_true:N \l_chemformula_first_last_math_bool
90          \__chemformula_read_escape_mathbraces:w #1 \q_nil
91      }

```

Added by lwarp:

```

92      \bool_if:NT \l_chemformula_first_last_LWRdollar_bool%      lwarp
93      {
94          \bool_set_true:N \l_chemformula_first_last_math_bool%  lwarp
95          \__chemformula_read_escape_LWRdollar:w #1 \q_nil%    lwarp
96      }
97  }

```

\chemformula_read_escape_LWRdollar

The following parses the contents inside the new dollars.

lwarp keeps the dollar as its original math shift until the document starts. While chemmacros is being patched, the dollar must temporarily be set to its new meaning during the following definition.

```

98 \begingroup
99 \catcode`\$=\active
100
101 \cs_new_protected:Npn \__chemformula_read_escape_LWRdollar:w $#1$ \q_nil
102 {
103     \__chemformula_read_escape_math:n {#1}
104 }
105
106 \endgroup

```

\chemformula_bool_set_if_first_last

The following looks at the first and last tokens for delimiters to escape math inside \ch. The original definition is modified to look for the control sequences which are used by the new meaning of \$.

```

107 \cs_new_protected:Npn \__chemformula_bool_cs_set_if_first_last:NnNN #1#2#3#4
108  {
109      \int_zero:N \l_chemformula_tmpa_int
110      \int_zero:N \l_chemformula_tmpb_int
111      \int_set:Nn \l_chemformula_tmpa_int { \tl_count:n {#2} }
112      \tl_map_inline:nn {#2}
113      {
114          \int_incr:N \l_chemformula_tmpb_int
115          \int_compare:nT { \l_chemformula_tmpb_int = 1 }

```

```
116 {
```

At the start, the cs_ version compares control sequences:

```
117 \ifdefstreq{\#1}{#3}%
118 {
119     \bool_set_true:N #1
120 }%
121 {}
122 }
```

At the end, compare more control sequences:

```
123 \int_compare:nT { \l__chemformula_tmpb_int = \l__chemformula_tmpa_int }
124 {
125     \ifdefstreq{\#1}{#4}
126     {}
127     {
128         \bool_set_false:N #1
129     }
130 }
131 }
132 }
```

\chemformula_first_last_math

Modified to check for the new meaning of \$ at first/last:

```
133 \cs_gset_protected:Npn \__chemformula_first_last_math:n #1
134 {
135     \bool_set_false:N \l__chemformula_first_last_math_bool
136     \bool_set_false:N \l__chemformula_first_last_dollar_bool
137     \bool_set_false:N \l__chemformula_first_last_LWRdollar_bool% lwarp
138     \bool_set_false:N \l__chemformula_first_last_mathbraces_bool
139     \__chemformula_bool_set_if_first_last:Nnnn
140         \l__chemformula_first_last_dollar_bool
141         {#1}
142         { $ } { $ }
143     \bool_if:NF \l__chemformula_first_last_dollar_bool
144     {
145         \__chemformula_bool_set_if_first_last:Nnnn
146             \l__chemformula_first_last_mathbraces_bool
147             {#1}
148             { \( } { \) }
```

Added by lwarp:

```
149     \bool_if:NF \l__chemformula_first_last_mathbraces_bool% lwarp
150     {
151         \__chemformula_bool_cs_set_if_first_last:NnNN
152             \l__chemformula_first_last_LWRdollar_bool
153             {#1}
```

```

154           { \LWR@newsingledollar } { \LWR@newsingledollar }
155       }%
156   }
157 }

158 \ExplSyntaxOff

```

File 62 **lwarf-chemgreek.sty**§ 158 Package **chemgreek***(Emulates or patches code by CLEMENS NIEDERBERGER.)*Pkg **chemgreek** chemgreek is patched for use by lwarf.**Greek symbols** To use text-mode symbols, use packages textalpha or textgreek. Using the other**⚠ package selection** packages supported by chemgreek will result in math-mode greek characters, which will result in SVG images being used. These images will be hashed.**⚠ X_ELATEX, L_ATEX** If using X_ELATEX or L_ATEX, select the **fontspec** mapping:

```
\selectchemgreekmapping{fontspec}
```

for HTML output: 1 \LWR@ProvidesPackagePass{chemgreek} [2016/02/10]

```

2 \ExplSyntaxOn
3
4 \cs_gset_protected:Npn \chemgreek_text:n #1
5   { { \text {#1} } }
6
7 \appto\LWR@restoreorigformatting{%
8 \cs_set_protected:Npn \chemgreek_text:n #1%
9   { \ensuremath { \text {#1} } } }%
10 }
11
12 \ExplSyntaxOff

```

File 63 **lwarf-chemmacros.sty**§ 159 Package **chemmacros***(Emulates or patches code by CLEMENS NIEDERBERGER.)*Pkg **chemmacros** chemmacros is patched for use by lwarf.

for HTML output: 1 \LWR@ProvidesPackagePass{chemmacros}

SVG file hashing assumes that the relevant options are constant for the entire document.

§ 159.1 Changes to the user's document

⚠ `\makepolymerdelims` When using `\makepolymerdelims`, enclose the entire expression inside a `polymerdelims` environment, such as (from the chemmacros manual):

```
\begin{polymerdelims}
\chemfig{-[@{op},.75]CH_2-CH(-[6]Cl)-[@{cl},0.25]}
\makepolymerdelims{5pt}[27pt]{op}{cl}
\end{polymerdelims}
```

⚠ **redox reactions** Redox reactions must be enclosed inside a `redoxreaction` environment. For print output, extra space must be included above and/or below the result, so they are declared as arguments to the environment, instead of being manually entered as per the chemmacros manual. For HTML output, the extra space is ignored and a `lateximage` is used instead.

```
\begin{redoxreaction}{7mm}{7mm}
\OX{a,Na} $\rightarrow$ \OX{b,Na}\pch\redox(a,b){oxidation}
\end{redoxreaction}
```

§ 159.2 Code

§ 159.3 Loading modules

Patching chemmacros modules must be done `\AtBeginDocument`, since modules are invoked by the user in the preamble, and each patch is only done if the module is loaded.

```
2 \ExplSyntaxOn
3
4 \newcommand{\@ifchemmacrosmoduleloaded}[1]{%
5 \c_if_l@aded{\c__chemmacros_module_extension_tl}{\c__chemmacros_module_prefix_tl.\#1}%
6 }
7
8 \ExplSyntaxOff
```

§ 159.4 New environments

\makepolymerdelims and redox reactions must be enclosed in a `lateximage` during HTML output. These environments are provided here in HTML mode, and in the `l warp` core in print mode, as a high-level semantic syntax which automatically embeds the contents in a `lateximage` with an appropriate `alt` tag.

Env `polymerdelims`

```
9 \DeclareDocumentEnvironment{polymerdelims}{}  
10 {\begin{lateximage}[(-chemmacros- polymer)]}  
11 {\end{lateximage}}
```

Env `redoxreaction` *{<space above>} {<space below>}*

For HTML output, the above and below space is ignored, and a `lateximage` is used instead. For the print output version, see section 85.

```
12 \DeclareDocumentEnvironment{redoxreaction}[m m]  
13 {\begin{lateximage}[(-chemmacros- redoxreaction)]}  
14 {\end{lateximage}}
```

```
15 \ExplSyntaxOn
```

§ 159.5 Acid-base

```
16 \AtBeginDocument{  
17 @ifchemmacrosmoduleloaded{acid-base}{  
18 \PackageInfo{l warp}{Patching~chemmacros~module~acid-base}  
19  
20 \cs_gset_protected:Npn \chemmacros_p:n #1  
21 {  
22     \begingroup  
23     \boolfalse{mathjax}  
24     \LWR@subsingle dollar*{  
25         \textbackslash{}p\textbackslash{}f\textbackslash{}LWR@HTMLsanitize{\#1}\}  
26     }{  
27         chemmacros_p\protect\LWR@HTMLsanitize{\detokenize\expandafter{\#1}}%  
28     }{  
29     \group_begin:  
30         \mbox  
31         {  
32             \chemmacros_p_style:n {p}  
33             \ensuremath {\#1}  
34         }  
35     \group_end:  
36     }  
37     \endgroup  
38 }}
```

```
39
40 \RenewDocumentCommand \pH {} {
41     \begingroup
42     \boolfalse{mathjax}
43     \LWR@subsingle dollar*\{\textbackslash{}pH\}{chemmacros}{
44         \chemmacros_p:n { \chemmacros_chemformula:n {H} }
45     }
46     \endgroup
47 }
48
49 \RenewDocumentCommand \pOH {} {
50     \begingroup
51     \boolfalse{mathjax}
52     \LWR@subsingle dollar*\{\textbackslash{}pOH\}{chemmacros}{
53         \chemmacros_p:n { \chemmacros_chemformula:n {OH} }
54     }
55     \endgroup
56 }
57
58 \RenewDocumentCommand \pKa {O{}}
59 {
60     \begingroup
61     \boolfalse{mathjax}
62     \LWR@subsingle dollar*\{\textbackslash{}pKa{[]}\}{chemmacros #1} {
63         \chemmacros_p:n
64         {
65             \Ka \ifblank {\#1} {}
66             { {} \c_math_subscript_token { \chemmacros_bold:n {\#1} } }
67         }
68     }
69     \endgroup
70 }
71
72 \RenewDocumentCommand \pKb {O{}}
73 {
74     \begingroup
75     \boolfalse{mathjax}
76     \LWR@subsingle dollar*\{\textbackslash{}pKb{[]}\}{chemmacros #1} {
77         \chemmacros_p:n
78         {
79             \Kb \ifblank {\#1} {}
80             { {} \c_math_subscript_token { \chemmacros_bold:n {\#1} } }
81         }
82     }
83     \endgroup
84 }
85
86 \LetLtxMacro \LWR@chemmacros@origKa \Ka
87 \renewcommand*{\Ka}{%
88     \begingroup
```

```

89   \boolearn{mathjax}
90   \LWR@subsingle$*\{\textbackslash{}Ka\}{chemmacros}\%
91     \LWR@chemmacros@origKa\%
92   }%
93   \endgroup
94 }
95
96 \LetLtxMacro{\LWR@chemmacros@origKb\Kb
97 \renewcommand*\{\Kb}\%
98   \begingroup
99   \boolearn{mathjax}
100  \LWR@subsingle$*\{\textbackslash{}Kb\}{chemmacros}\%
101    \LWR@chemmacros@origKb\%
102  }%
103  \endgroup
104 }
105
106 \LetLtxMacro{\LWR@chemmacros@origKw\Kw
107 \renewcommand*\{\Kw}\%
108   \begingroup
109   \boolearn{mathjax}
110   \LWR@subsingle$*\{\textbackslash{}Kw\}{chemmacros}\%
111     \LWR@chemmacros@origKw
112   }%
113   \endgroup
114 }
115
116 }{}\% \ifchemmacrosmoduleloaded
117 }% AtBeginDocument

```

§ 159.6 Charges

```

118 \AtBeginDocument{
119 \ifchemmacrosmoduleloaded{charges}{
120 \PackageInfo{lwarp}{Patching~chemmacros~module~charges}
121
122 \cs_gset_protected:Npn \fplus {
123   \begingroup
124   \boolearn{mathjax}
125   \LWR@subsingle$*\{\textbackslash{}fplus\}{chemmacros}
126   { \LWR@origensuredmath{\chemformula_fplus:} }
127   \endgroup
128 }
129 \cs_gset_protected:Npn \fminus {
130   \begingroup
131   \boolearn{mathjax}
132   \LWR@subsingle$*\{\textbackslash{}fminus\}{chemmacros}
133   { \LWR@origensuredmath{\chemformula_fminus:} }
134   \endgroup

```

```
135 }
136
137 }{}}% \@ifchemmacrosmoduleloaded
138 }% AtBeginDocument
```

§ 159.7 Nomenclature

```
139 \AtBeginDocument{
140 \ifchemmacrosmoduleloaded{nomenclature}{
141 \PackageInfo{l warp}{Patching~chemmacros~module~nomenclature}
142
143 \cs_gset_protected:Npn \chemmacros_charge:n #1
144 {
145     \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}
146     {\chemmacros_chemformula:n { {}^{\#1} }}
147     {
148         \ifmmode
149             {\chemmacros_chemformula:n { {}^{\#1} }}
150         \else
151             { \textsuperscript{\ensuremath{\#1}} }
152         \fi
153     }
154 }
155
156
157 \LetLtxMacro\LWR@chemmacros@origchemprime\chemprime
158
159 \protected\def\chemprime { \HTMLunicode{2032} }
160
161 \appto\LWR@restoreorigformatting{%
162 \LetLtxMacro\chemprime\LWR@chemmacros@origchemprime%
163 }

164 \ChemCompatibilityFrom{5.8}
165 \cs_gset_protected:Npn \__chemmacros_cip:n #1
166 {
167     \tl_set:Nn \l__chemmacros_tmpa_tl {\#1}
168     \int_step_inline:nnnn {0} {1} {9}
169     {
170         \tl_replace_all:Nnn \l__chemmacros_tmpa_tl
171             {##1}
172             { \l__chemmacros_cip_number_tl ##1 }
173     }
174     {
175         \l__chemmacros_cip_inner_tl
176         \LWR@textcolor{\LWR@textfont{}% l warp
177             \l__chemmacros_tmpa_tl
178         }% l warp
179     }
180 }
```

```
181 \EndChemCompatibility

182 \RenewDocumentCommand \Sconf { O{S} } {
183     \begin{ lateximage } [ \textbackslash textbackslash {} \Sconf { [ ] \#1 { } } ]
184         \chemmacros_sconf : n { \#1 }
185     \end{ lateximage }
186 }

187
188 \RenewDocumentCommand \Rconf { O{R} } {
189     \begin{ lateximage } [ \textbackslash textbackslash {} \Rconf { [ ] \#1 { } } ]
190         \chemmacros_rconf : n { \#1 }
191     \end{ lateximage }
192 }

193 \cs_gset_protected:Npn \chemmacros_hapto:n #1
194 {
195     \begingroup
196     \boolexpr{mathjax}
197     \LWR@subsingle dollar*{ \textbackslash textbackslash {} \hapto \{ \#1 \} }{ \chemmacros }{
198         \chemmacros_coordination_symbol:nnnn
199         { \l_chemmacros_coord_use_hyphen_bool }
200         {
201             \chemmacros_if_compatibility:nnTF { > } { 5.7 }
202             { \c_true_bool }
203             { \c_false_bool }
204         }
205         { \chemeta }
206         { \#1 }
207     }
208     \endgroup
209 }

210
211 \cs_gset_protected:Npn \chemmacros_dento:n #1
212 {
213     \begingroup
214     \boolexpr{mathjax}
215     \LWR@subsingle dollar*{ \textbackslash textbackslash {} \dento \{ \#1 \} }{ \chemmacros }{
216         \chemmacros_coordination_symbol:nnnn
217         { \l_chemmacros_coord_use_hyphen_bool }
218         {
219             \chemmacros_if_compatibility:nnTF { > } { 5.7 }
220             { \c_true_bool }
221             { \c_false_bool }
222         }
223         { \chemkappa }
224         { \#1 }
225     }
226     \endgroup
227 }

228
```

```

229 \cs_gset_protected:Npn \chemmacros_bridge:n #1
230   {
231     \begingroup
232     \boolfalse{mathjax}
233     \LWR@subsingle dollar{\textbackslash bridge{\#1}}{\chemmacros}[
234       \chemmacros_coordination_symbol:nnnn
235       { \l__chemmacros_coord_use_hyphen_bool }
236       { \l__chemmacros_bridge_super_bool }
237       { \chemmu }
238       {\#1}
239     ]
240   \endgroup
241 }
242 }{}% \@ifchemmacrosmoduleloaded
243 }% AtBeginDocument

```

§ 159.8 Particles

```

244 \AtBeginDocument{
245 \ifchemmacrosmoduleloaded{particles}[
246 \PackageInfo{lwarf}{Patching-chemmacros-module-particles}
247
248 \cs_gset_protected:Npn \chemmacros_declare_nucleophile:Nn #1#2
249   {
250     \cs_set_protected:cpn {__chemmacros_ \chemmacros_remove_backslash:N #1:}
251     {
252       \bool_if:NTF \l__chemmacros_nucleophile_elpair_bool
253       {
254         \chemmacros_elpair:n { #2 }
255         \chemmacros_if_compatibility:nnT {>=} {5.3}
256         { \skip_horizontal:N \l__chemmacros_nucleophile_dim }
257         \chemmacros_chemformula:n { {}^{\{-\}} }
258       }
259       { \chemmacros_chemformula:n { #2^{\{-\}} } }
260     }
261     \DeclareDocumentCommand #1 {o}
262     {%
263       \begin{ lateximage }%
264       \group_begin:%
265       \IfNoValueF {##1}%
266       { \chemmacros_set_keys:nn {particles} {##1} }%
267       \use:c {__chemmacros_ \chemmacros_remove_backslash:N #1:}%
268       \group_end:%
269       \end{ lateximage }%
270     }
271   }
272
273 \RenewChemNucleophile \Nuc {Nu}
274 \RenewChemNucleophile \ba {ba}

```

```

275
276 }{}% \@ifchemmacrosmoduleloaded
277 }% AtBeginDocument

```

§ 159.9 Phases

```

278 \AtBeginDocument{
279 \@ifchemmacrosmoduleloaded{phases} {
280 \PackageInfo{l warp}{Patching~chemmacrosmodule~phases}
281
282 \cs_undefine:N \chemmacrospnphase:n
283 \cs_new_protected:Npn \chemmacrospnphase:n #1
284 {
285     \chemmacrospnleavevmode:
286     \bool_if:NTF \l__chemmacrospnphases_sub_bool
287     {
288         \ifnumequal{\value{LWR@lateximagedepth}}{0}
289         {
290             \textsubscript{ (#1) }
291         }
292         {
293             \chemformula_subscript:n { (#1) }
294         }
295     }
296     {
297         \skip_horizontal:N \l__chemmacrospnphases_space_dim
298         \chemmacrostext:n { (#1) }
299     }
300 }
301
302 }{}% \@ifchemmacrosmoduleloaded
303 }% AtBeginDocument

```

§ 159.10 Mechanisms

```

304 \AtBeginDocument{
305 \@ifchemmacrosmoduleloaded{mechanisms} {
306 \PackageInfo{l warp}{Patching~chemmacrosmodule~mechanisms}
307
308 \chemmacrospndefine_keys:nn {textmechanisms}
309 {
310     type .choice: ,
311     type / .code:n =
312     {
313         \chemmacrospnset_mechanisms:nnn { S }
314         {
315             \textsubscript{N}
316         }
317     }

```

```
318     } ,
319     type / 1 .code:n    =
320     {
321       \_\_chemmacros_set_mechanisms:nnn { S }
322       {
323         \textsubscript{N}
324         1
325       }
326       { }
327     } ,
328     type / 2 .code:n    =
329     {
330       \_\_chemmacros_set_mechanisms:nnn { S }
331       {
332         \textsubscript{N}
333         2
334       }
335       { }
336     } ,
337     type / se .code:n   =
338     {
339       \_\_chemmacros_set_mechanisms:nnn { S }
340       {
341         \textsubscript{E}
342       }
343       { }
344     } ,
345     type / 1e .code:n   =
346     {
347       \_\_chemmacros_set_mechanisms:nnn { S }
348       {
349         \textsubscript{E}
350         1
351       }
352       { }
353     } ,
354     type / 2e .code:n   =
355     {
356       \_\_chemmacros_set_mechanisms:nnn { S }
357       {
358         \textsubscript{E}
359         2
360       }
361       { }
362     } ,
363     type / ar .code:n   =
364     {
365       \_\_chemmacros_set_mechanisms:nnn { S }
366       {
367         \textsubscript{E}
```

```
368         }
369         { Ar - }
370     } ,
371     type / e .code:n   =
372     { \_chemmacros_set_mechanisms:nnn { E } { } { } } ,
373     type / e1 .code:n  =
374     { \_chemmacros_set_mechanisms:nnn { E } { 1 } { } } ,
375     type / e2 .code:n  =
376     { \_chemmacros_set_mechanisms:nnn { E } { 2 } { } } ,
377     type / cb .code:n  =
378     {
379         \_chemmacros_set_mechanisms:nnn { E }
380         {
381             1
382             \textsubscript{cb}
383         }
384         {
385     } ,
386     type      .default:n =
387 }
388
389 \cs_gset_protected:Npn \chemmacros_mechanisms:n #1
390 {
391     \tl_if_blank:nTF {#1}
392     { \chemmacros_set_keys:nn {textmechanisms} { type } }
393     { \chemmacros_set_keys:nn {textmechanisms} { type = #1 } }
394     \mbox
395     {
396         \tl_use:N \l_chemmacros_mechanisms_ar_tl
397         \tl_use:N \l_chemmacros_mechanisms_type_tl
398         \tl_use:N \l_chemmacros_mechanisms_mol_tl
399     }
400 }
401
402 \appto\LWR@restoreorigformatting{%
403 \cs_set_protected:Npn \chemmacros_mechanisms:n #1%
404 {%
405     \tl_if_blank:nTF {#1}%
406     { \chemmacros_set_keys:nn {mechanisms} { type } }%
407     { \chemmacros_set_keys:nn {mechanisms} { type = #1 } }%
408     \mbox%
409     {%
410         \tl_use:N \l_chemmacros_mechanisms_ar_t1%
411         \tl_use:N \l_chemmacros_mechanisms_type_t1%
412         \tl_use:N \l_chemmacros_mechanisms_mol_t1%
413     }%
414 }%
415 }
416
417 }{}% \@ifchemmacrosmoduleloaded
```

```
418 }% AtBeginDocument
```

§ 159.11 Newman

```
419 \AtBeginDocument{
420 \@ifchemmacrosmoduleloaded{newman}{
421 \PackageInfo{lwarp}{Patching~chemmacros~module~newman}
422
423 \RenewDocumentCommand \newman {od()m}%
424 {
425   \IfValueTF{#2}
426   {\begin{ lateximage }[\textbackslash newman{#2}\{#3\}] }
427   {\begin{ lateximage }[\textbackslash newman\{#3\}] }
428   \group_begin:
429     \IfNoValueF{#1} { \chemmacros_set_keys:nn {newman} {#1} }
430     \IfNoValueTF{#2}
431       { \chemmacros_newman:nn { } {#3} }
432       { \chemmacros_newman:nn {#2} {#3} }
433   \group_end:
434   \end{ lateximage }
435 }
436
437 }{}% \@ifchemmacrosmoduleloaded
438 }% AtBeginDocument
```

§ 159.12 Orbital

```
439 \AtBeginDocument{
440 \@ifchemmacrosmoduleloaded{orbital} {
441 \PackageInfo{lwarp}{Patching~chemmacros~module~orbital}
442
443 \RenewDocumentCommand \orbital {om}%
444 {
445   \IfValueTF{#1}
446   {
447     \begin{ lateximage }[%
448       \textbackslash orbital{[]}\LWR@HTMLsanitize{#1}{[]}\{#2\}%
449     ][] [margin-left: 1em ; margin-right: 1em]
450   }
451   {
452     \begin{ lateximage }[%
453       \textbackslash orbital\{#2\}%
454     ][] [margin-left: 1em ; margin-right: 1em]
455   }
456   \group_begin:
457     \chemmacros_set_keys:nn {orbital/type} {#2}
458     \IfNoValueTF{#1}
459       { \chemmacros_orbital:n { } }
460       { \chemmacros_orbital:n {#1} }
```

```

461     \group_end:
462     \end{lateximage}
463 }
464
465 }{}% \@ifchemmamacrosmoduleloaded
466 }% AtBeginDocument

```

§ 159.13 Reactions

```

\chemmamacros_declare_reaction_env {\chem} {\math} {\args} {\argumentlist}
467 \AtBeginDocument{
468 \ifchemmamacrosmoduleloaded{reactions}{%
469 \PackageInfo{lwarf}{Patching-chemmamacros-module-reactions}}
470
471 \cs_gset_protected:Npn \chemmamacros_declare_reaction_env:nnnn #1#2#3#4
472 {
473     \exp_args:Nnx \DeclareDocumentEnvironment {\#1} { O{} \prg_replicate:nn {\#3+0} {m} }
474     {
475         \boolfalse{mathjax}%
476         \lwarf
477         \chemmamacros_add_reaction_description:n {\##1}
478         \__chemmamacros_begin_reaction:
479         \chemmamacros_reaction_read:nw {\#2} {\#4}
480     }
481     {
482         \__chemmamacros_end_reaction:
483     }
484 \cs_generate_variant:Nn \chemmamacros_declare_reaction_env:nnnn {nnnV}
485
486 \RenewChemReaction {reaction} {equation}
487 \RenewChemReaction {reaction*} {equation*}
488 \RenewChemReaction {reactions} {align}
489 \RenewChemReaction {reactions*} {align*}
490
491 }{}% \ifchemmamacrosmoduleloaded
492 }% AtBeginDocument

```

§ 159.14 Redox

```

493 \AtBeginDocument{
494 \ifchemmamacrosmoduleloaded{redox}{%
495 \PackageInfo{lwarf}{Patching-chemmamacros-module-redox}}
496
497 \NewDocumentCommand \LWR@chemmamacros@ox { s m >{\SplitArgument{1}{,}}m }
498 {
499     \IfBooleanTF {\#1}
500     { \chemmamacros_ox:nnnn {\#1} {\#2} #3 }
501     { \chemmamacros_ox:nnnn { } {\#2} #3 }

```

```

502   }
503
504 \RenewDocumentCommand \ox { s O{} m }
505 {
506   \begingroup
507   \boolfalse{mathjax}
508   \IfBooleanTF {#1}
509   {
510     \LWR@subsingle dollar*{%
511       \textbackslash ox*{\LWR@HTMLsanitize{#3}}%
512     }%
513     star \protect\LWR@HTMLsanitize{\detokenize\expandafter{#2}}%
514   }%
515   \LWR@chemmacros@ox* {#2} {#3}%
516   contents
517 }%
518 {
519   \LWR@subsingle dollar*{%
520     \textbackslash ox*{\LWR@HTMLsanitize{#3}}%
521   }%
522   \protect\LWR@HTMLsanitize{\detokenize\expandafter{#2}}%
523   \LWR@chemmacros@ox {#2} {#3}%
524   contents
525 }%
526 }
527 \endgroup
528 }
529 }%
530 }%
531 }%
532 }%
533 }%
534 }%
535 }%
536 }%
537 }%
538 name = \chemmacros_translate:n {scheme-name}%
539 }
540 }%
541 }%
542 }%
543 }%

```

§ 159.15 Scheme

Fix for chemmacros as of v5.8b, when using newfloat and babel:

```

532 \AtBeginDocument{
533 \@ifchemmacrosmoduleloaded{scheme}{
534 \PackageInfo{l warp}{Patching~chemmacros~module~scheme}}
535
536 \ifdefstring{\schemename}{los}{
537 \SetupFloatingEnvironment{scheme}%
538 name = \chemmacros_translate:n {scheme-name}%
539 }
540 }%
541 }%
542 }%
543 }%

```

§ 159.16 Spectroscopy

```
544 \AtBeginDocument{
545 @ifchemmacrosmoduleloaded{spectroscopy}{
546 \PackageInfo{l warp}{Patching~chemmacros~module~spectroscopy}
547
548 \ChemCompatibilityTo{5.8}
549 \cs_gset_protected:Npn \__chemmacros_nmr_base:nn #1#2
550 {
551     \tl_if_blank:VF \g__chemmacros_nmr_element_coupled_tl
552     {
553         \tl_put_left:Nn \g__chemmacros_nmr_element_coupled_tl { \{ }
554         \tl_put_right:Nn \g__chemmacros_nmr_element_coupled_tl { \} }
555     }
556     \tl_put_left:Nn \g__chemmacros_nmr_element_coupled_tl { #2}
557 %
558     \chemmacros_chemformula:n { ^{#1} }
559     \textsuperscript{#1}
560     \bool_if:NTF \l__chemmacros_nmr_parse_bool
561         { \chemformula_ch:nV {} \g__chemmacros_nmr_element_coupled_tl }
562         { \chemmacros_chemformula:V \g__chemmacros_nmr_element_coupled_tl }
563     \tl_use:N \l__chemmacros_nmr_element_method_connector_tl
564     \tl_use:N \l__chemmacros_nmr_method_tl
565 }
566 \EndChemCompatibility
567 \ChemCompatibilityFrom{5.8}
568 \cs_gset_protected:Npn \__chemmacros_nmr_base:nn #1#2
569 {
570     \group_begin:
571         \tl_use:N \l__chemmacros_nmr_base_format_tl
572         \tl_if_blank:VF \g__chemmacros_nmr_element_coupled_tl
573         {
574             \tl_put_left:Nn \g__chemmacros_nmr_element_coupled_tl { \{ }
575             \tl_put_right:Nn \g__chemmacros_nmr_element_coupled_tl { \} }
576         }
577         \tl_put_left:Nn \g__chemmacros_nmr_element_coupled_tl { #2}
578         \chemmacros_chemformula:n { ^{#1} }
579         \textsuperscript{#1}
580         \tl_if_blank:VF \g__chemmacros_nmr_element_coupled_tl
581         {
582             \bool_if:NTF \l__chemmacros_nmr_parse_bool
583                 { \chemformula_ch:nV {} \g__chemmacros_nmr_element_coupled_tl }
584                 { \chemmacros_chemformula:V \g__chemmacros_nmr_element_coupled_tl }
585         }
586         \tl_use:N \l__chemmacros_nmr_element_method_connector_tl
587         \tl_use:N \l__chemmacros_nmr_method_tl
588     \group_end:
589 }
590 \EndChemCompatibility
591
```

```
592 \cs_gset_protected:Npn \chemmacros_nmr_position:n #1
593   {
594     \chemmacros_chemformula:x
595     {
596       \exp_not:V \g__chemmacros_nmr_element_tl
597       \bool_if:NF \l__chemmacros_nmr_position_side_bool
598       {
599         \tl_if_eq:NNF \l__chemmacros_nmr_position_tl {^}% l warp
600         { \textsuperscript{\exp_not:n { \#1 } } } % l warp
601         { \textsubscript{\exp_not:n { \#1 } } } % l warp
602 %
603         \exp_not:V \l__chemmacros_nmr_position_tl
604         \exp_not:n { \#1 }
605       }
606     }
607     \bool_if:NT \l__chemmacros_nmr_position_side_bool
608     {
609       \tl_use:N \l__chemmacros_nmr_position_tl
610       \chemmacros_nmr_position:n { \#1 }
611     }
612   }
613 \cs_gset_protected:Npn \__chemmacros_nmr_coupling:w (#1;#2)
614   {
615     \tl_set:Nn \l__chemmacros_nmr_coupling_bonds_tl
616     {
617       \l__chemmacros_nmr_coupling_bonds_pre_tl
618       #1
619       \l__chemmacros_nmr_coupling_bonds_post_tl
620     }
621     \bool_if:NTF \l__chemmacros_nmr_coupling_nuclei_sub_bool
622     {
623       \tl_set:Nn \l__chemmacros_nmr_coupling_nuclei_tl
624       {
625 %
626         \c_math_subscript_token
627         \textsubscript% l warp
628         {
629           \l__chemmacros_nmr_coupling_nuclei_pre_tl
630           \chemmacros_chemformula:n { \#2 }
631           \l__chemmacros_nmr_coupling_nuclei_post_tl
632         }
633       }
634     {
635       \tl_set:Nn \l__chemmacros_nmr_coupling_nuclei_tl
636       {
637         \l__chemmacros_nmr_coupling_nuclei_pre_tl
638         \chemmacros_chemformula:n { \#2 }
639         \l__chemmacros_nmr_coupling_nuclei_post_tl
640       }
641     }
```

```
642      \__chemmacros_nmr_coupling_aux_i:w
643  }
644
645 \AfterEndPreamble{%
646 % \NMR{<num>,<elem>}(<num>,<unit>)[<solvent>] ALL arguments are optional
647 % \NMR* same but without ": \$\delta$" at end
648 \cs_gset_protected:Npn \chemmacros_nmr:nnnn #1#2#3#4
649  {
650    \bool_if:NT \l__chemmacros_nmr_list_bool { \item \scan_stop: }
651    \group_begin:
652      \chemmacros_leave_vmode:
653      \bool_set_false:N \l__chemmacros_nmr_frequency_bool
654      \bool_set_false:N \l__chemmacros_nmr_solvent_bool
655      \tl_if_empty:nF {#3}
656      { \bool_set_true:N \l__chemmacros_nmr_frequency_bool }
657      \tl_if_empty:nF {#4}
658      { \bool_set_true:N \l__chemmacros_nmr_solvent_bool }
659      \bool_if:nT
660      {
661        \l__chemmacros_nmr_frequency_bool
662        ||
663        \l__chemmacros_nmr_solvent_bool
664      }
665      { \bool_set_true:N \l__chemmacros_nmr_delimiters_bool }
666      \bool_if:nT
667      {
668        \l__chemmacros_nmr_frequency_bool
669        &&
670        \l__chemmacros_nmr_solvent_bool
671      }
672      { \bool_set_true:N \l__chemmacros_nmr_comma_bool }
673      \tl_if_empty:nTF {#2}
674      {
675        \__chemmacros_nmr_nucleus:VV
676        \l__chemmacros_nmr_isotope_default_tl
677        \l__chemmacros_nmr_element_default_tl
678      }
679      { \__chemmacros_nmr_nucleus:w #2 \q_stop }
680      \mode_if_math:TF
681      {
682        \text
683        {
684          \group_begin:
685          \tl_use:N \l__chemmacros_nmr_format_tl
686          \LWR@textcurrentcolor{\LWR@textcurrentfont{%
687            \__chemmacros_nmr_base:VV
688            \g__chemmacros_nmr_isotope_tl
689            \g__chemmacros_nmr_element_tl
690            \bool_if:NT \l__chemmacros_nmr_delimiters_bool
691            { ~ ( } }}
```

```
692          \bool_if:NT \l__chemmacros_nmr_frequency_bool
693              { \__chemmacros_nmr_frequency:n {#3} }
694          \bool_if:NT \l__chemmacros_nmr_comma_bool
695              { , ~ }
696          \bool_if:NT \l__chemmacros_nmr_solvent_bool
697              { \chemmacros_chemformula:n {#4} }
698          \bool_if:NT \l__chemmacros_nmr_delimiters_bool
699              { ) }
700          \tl_if_blank:nT {#1} {::}
701 }% lwarp
702         \group_end:
703     }
704     \tl_if_blank:nT {#1}
705     {
706         \delta
707         \text { \l__chemmacros_nmr_delta_tl }
708         \bool_if:NT \l__chemmacros_nmr_use_equal_bool {=}
709     }
710 }
711 {
712     \group_begin:
713     \tl_use:N \l__chemmacros_nmr_format_tl
714 \LWR@textcurrentcolor{\LWR@textcurrentfont}% lwarp
715     \__chemmacros_nmr_base:VV
716         \g__chemmacros_nmr_isotope_tl
717         \g__chemmacros_nmr_element_tl
718     \bool_if:NT \l__chemmacros_nmr_delimiters_bool
719         {~()}
720     \bool_if:NT \l__chemmacros_nmr_frequency_bool
721         { \__chemmacros_nmr_frequency:n {#3} }
722     \bool_if:NT \l__chemmacros_nmr_comma_bool
723         {,~}
724     \bool_if:NT \l__chemmacros_nmr_solvent_bool
725     {
726         \bool_if:NTF \l__chemmacros_nmr_parse_bool

727 %             { \chemformula_ch:nn { } {#4} }% original
728             {\ch{#4}}% lwarp
729             {#4}
730     }
731     \bool_if:NT \l__chemmacros_nmr_delimiters_bool
732         {()}

733 }% lwarp
734     \tl_if_blank:nT {#1} {::}
735     \group_end:
736     \tl_if_blank:nT {#1}
737     {
738         \tl_use:N \c_space_tl
739         \c_math_toggle_token
```

```

740           \delta
741           \c_math_toggle_token
742           \l_chemmacros_nmr_delta_tl
743           \bool_if:NT \l_chemmacros_nmr_use_equal_bool {~=}
744       }
745   }
746 \group_end:
747 }
748 }% AfterEndPreamble
749
750
751 \RenewDocumentCommand \chemmacros_data:w { smo }
752 {
753   \bool_if:NT \l_chemmacros_nmr_list_bool { \item }
754   {
755     \tl_use:N \l_chemmacros_nmr_format_tl #2
756     \tl_use:N \l_chemmacros_nmr_format_tl
757     \LWR@textcurrentcolor{\LWR@textcurrentfont{%
758       \#2
759       \IfNoValueF {#3} { ~ ( #3 ) }
760       \IfBooleanT {#1} { \bool_if:NT \l_chemmacros_nmr_use_equal_bool { : } }
761     }}% lwarp
762   }
763   \IfBooleanF {#1} { \bool_if:NT \l_chemmacros_nmr_use_equal_bool { ~ = } }
764 }
765
766 }{}% \@ifchemmacrosmoduleloaded
767 }% AtBeginDocument

```

§ 159.17 Thermodynamics

```

768 \AtBeginDocument{
769 \@ifchemmacrosmoduleloaded{thermodynamics}{%
770 \PackageInfo{lwarp}{Patching~chemmacros~module~thermodynamics}}
771
772 \cs_gset_protected:Npn \chemmacros_state:nn #1#2
773 {
774   \group_begin:
775   \boolfalse{mathjax}
776   \chemmacros_set_keys:nn {thermodynamics} {#1}
777   \LWR@subsingle dollar*{%
778     \textbackslash state\{\LWR@HTMLsanitize{#2}\}}% alt
779   }{%
780     chemmacros_state% add'l hashing
781     #1% options
782     LSP \tl_use:N \l_chemmacros_state_sp_left_tl% super/subscripts
783     LSB \tl_use:N \l_chemmacros_state_sb_left_tl
784     RSP \tl_use:N \l_chemmacros_state_sp_right_tl
785     RSB \tl_use:N \l_chemmacros_state_sb_right_tl

```

```

786      }
787      {
788          \LWR@origensuredmath{
789              \chemmacros_text:V \l__chemmacros_state_pre_tl
790              \c_math_superscript_token
791                  { \chemmacros_text:V \l__chemmacros_state_sp_left_tl }

```

Only add the subscripts if they are being used. This avoids causing an incorrect depth, as the empty subscript will be measured by TeX but cropped out by *pdfcrop*.

```

792          \tl_if_empty:NTF \l__chemmacros_state_sb_left_tl
793              {}
794              {
795                  \c_math_subscript_token
796                      { \chemmacros_text:V \l__chemmacros_state_sb_left_tl }
797              }
798              #2
799              \c_math_superscript_token
800                  { \chemmacros_text:V \l__chemmacros_state_sp_right_tl }
801          \tl_if_empty:NTF \l__chemmacros_state_sb_right_tl
802              {}
803              {
804                  \c_math_subscript_token
805                      { \chemmacros_text:V \l__chemmacros_state_sb_right_tl }
806              }
807          \chemmacros_text:V \l__chemmacros_state_post_tl
808      }
809      }
810  \group_end:
811 }
812 \cs_generate_variant:Nn \chemmacros_state:nn { nV }
813
814 \cs_gset_protected:Npn \chemmacros_declare_state:Nn #1#2
815 {
816     \chemmacros_define_keys:xn
817         {thermodynamics/\chemmacros_remove_backslash:N #1}
818     {
819         pre           .meta:nn = {chemmacros/thermodynamics} { pre = ##1 } ,
820         post          .meta:nn = {chemmacros/thermodynamics} { post = ##1 } ,
821         superscript-left .meta:nn = {chemmacros/thermodynamics} { superscript-left = ##1 } ,
822         superscript-right .meta:nn = {chemmacros/thermodynamics} { superscript-right = ##1 } ,
823         superscript    .meta:n  = {superscript-right = ##1} ,
824         subscript-left .meta:nn = {chemmacros/thermodynamics} { subscript-left = ##1 } ,
825         subscript-right .meta:nn = {chemmacros/thermodynamics} { subscript-right = ##1 } ,
826         subscript     .meta:n  = {subscript-left = ##1} ,
827         subscript-pos .choices:nn =
828             { left , right }
829             { \tl_set_eq:NN \l__chemmacros_state_sb_pos_tl \l_keys_choice_tl } ,
830         symbol        .tl_set:N = \l__chemmacros_state_symbol_tl ,

```

```

831     unit          .tl_set:N = \l_chemmacros_state_unit_tl
832   }
833 \DeclareDocumentCommand #1 { s0{}D(){}m }
834   {
835     \group_begin:
836       \chemmacros_set_keys:xn
837         {thermodynamics}/\chemmacros_remove_backslash:N #1}
838         {#2}
839     \tl_if_blank:nF {##3}
840     {
841       \chemmacros_set_keys:nx {thermodynamics}
842         { subscript-\l_chemmacros_state_sb_pos_tl = \exp_not:n {##3} }
843     }
844     \chemmacros_state:nV {##2} \l_chemmacros_state_symbol_tl
845     \chemmacros_set_keys_groups:nnn {thermodynamics} {variables} {##2}
846     \IfBooleanF {##1} { = ~ \SI {##4} { \l_chemmacros_state_unit_tl } }
847     \group_end:
848   }
849 }
```

The pre-existing macros are redefined with the new definition:

```

850 \RenewChemState \enthalpy { symbol = H , unit = \kilo\joule\per\mole }
851 \RenewChemState \entropy { symbol = S , unit = \joule\per\kelvin\per\mole , pre = }
852 \RenewChemState \gibbs { symbol = G , unit = \kilo\joule\per\mole }
853
854 }{ }% \@ifchemmacrosmoduleloaded
855 }% AtBeginDocument

856 \ExplSyntaxOff
```

File 64 lwarp-chemnum.sty

§ 160 Package **chemnum**

(Emulates or patches code by CLEMENS NIEDERBERGER.)

Pkg chemnum chemnum is patched for use by lwarp.

for HTML output:

```

1 \LWR@ProvidesPackagePass{chemnum}

2 \ExplSyntaxOn
3
4 \cs_gset_protected:Npn \chemnum_compound_write:n #1
5   {
6     \chemnum_get_compound_property:nn {#1} {pre-main-label-code}
7     \group_begin:
```

```

8      \bool_if:NTF \l__chemnum_compound_local_bool
9      { \l__chemnum_local_label_format_t1 }
10     { \chemnum_get_compound_property:nn {#1} {label-format} }
11     {
12       \LWR@textcurrentfont{
13         \chemnum_get_compound_property:nn {#1} {counter-representation}
14       }
15     }
16   \group_end:
17   \chemnum_get_compound_property:nn {#1} {post-main-label-code}
18 }
19
20 \cs_gset_protected:Npn \chemnum_subcompound_write:nn #1#2
21 {
22   \group_begin:
23   \bool_if:NTF \l__chemnum_compound_local_bool
24   { \l__chemnum_local_label_format_t1 }
25   { \chemnum_get_compound_property:nn {#1} {label-format} }
26   {
27     \LWR@textcurrentfont{
28       \chemnum_get_subcompound_property:nnn {#1} {#2}
29       {counter-representation}
30     }
31   }
32   \group_end:
33 }
34
35 \ExplSyntaxOff

```

File 65 lwarf-chkfloat.sty

§ 161 Package **chkfloat**

Pkg **chkfloat** **chkfloat** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{chkfloat}[2012/08/19]

File 66 lwarf-cite.sty

§ 162 Package **cite**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg **cite** **cite** is patched for use by **lwarf**.

for HTML output: 1 \LWR@ProvidesPackagePass{cite}

For the [super] option, the \kern must be removed:

```
2 \def\LWRCT@biblabel#1{\@citess{#1}\kern-\labelsep\,}
3
4 \ifdef\streq{\@biblabel}{\LWRCT@biblabel}
5 {
6     \def\@biblabel#1{\@citess{#1}}
7 }{}
```

For the [super] option, \textsuperscript is used instead of math superscript:

```
8 \def\@citess#1{\textsuperscript{#1}}
9
10 \DeclareDocumentCommand\citepunct{}{,\relax}
```

File 67 l warp-CJK.sty

§ 163 Package CJK

Pkg CJK CJK does not work with l warp unless called from ctex.

for HTML output:

```
1 \@ifpackageloaded{xeCJK}{}{
2     \LWR@loadnever{CJK}{ctex, xeCJK}
3 }
4
5 \LWR@ProvidesPackagePass{CJK}
```

File 68 l warp-CJKutf8.sty

§ 164 Package CJKutf8

Pkg CJKutf8 CJKutf8 does not work with l warp unless called from ctex.

for HTML output:

```
1 \@ifpackageloaded{xeCJK}{}{
2     \LWR@loadnever{CJKutf8}{ctex, xeCJK}
3 }
4
5 \LWR@ProvidesPackagePass{CJKutf8}
```

File 69 **l warp-clrdblp.g.sty**

§ 165 Package **clrdblp**

Pkg **clrdblp** **clrdblp** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{clrdblp}

File 70 **l warp-cmdtrack.g.sty**

§ 166 Package **cmdtrack**

Pkg **cmdtrack** **cmdtrack** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{cmdtrack}[2012/12/18]

2 \newcommand{\untrack}{[1]}{}

File 71 **l warp-color.g.sty**

§ 167 Package **color**

Pkg **color** Allowed but ignored. **xcolor** is then required as well.

color is superceded by **xcolor**, and **l warp** requires several of the features of **xcolor**. When **color** is requested, **xcolor** is loaded as well.

for HTML output: 1 \LWR@ProvidesPackagePass{color}
2 \RequirePackage{xcolor}

File 72 **l warp-colortbl.g.sty**

§ 168 Package **colortbl**

Pkg **colortbl** **colortbl** is emulated.

⚠ row/cell color Only use **\rowcolor** and **\cellcolor** at the start of a row, in that order.

`colortbl` ignores the overhang arguments.

for HTML output: A placeholder definition is forgotten first:

```
1 \let\rowcolor\relax
2
3 \LWR@ProvidesPackagePass{colortbl}
```

The following `\LWR@HTML` versions are used inside an `HTML tabular`.

`\columncolor` $[\langle model \rangle] \{\langle color \rangle\} [\langle left overhang \rangle] [\langle right overhang \rangle]$
`\LWR@getmynexttoken` is not used here because `\columncolor` is not used inside the data area of the tabular.

```
4 \NewDocumentCommand{\LWR@HTML@columncolor}{O{named} m o o}{%
5 \convertcolorspec{#1}{#2}{HTML}\LWR@columnHTMLcolor%
6 \LWR@addtabularcellcolor%
7 }
8
9 \AtBeginDocument{\LWR@formatted{columncolor}}
```

`\LWR@getmynexttoken` is used for `\rowcolor` because it is used inside the data area of the tabular.

`\rowcolor` $[\langle model \rangle] \{\langle color \rangle\} [\langle left overhang \rangle] [\langle right overhang \rangle]$
10 `\NewDocumentCommand{\LWR@HTML@rowcolor}{O{named} m o o}{%`
11 `\convertcolorspec{#1}{#2}{HTML}\LWR@rowHTMLcolor%`
12 `\LWR@getmynexttoken%`
13 }
14
15 `\AtBeginDocument{\LWR@expandableformatted{rowcolor}}`

`\cellcolor` $[\langle model \rangle] \{\langle color \rangle\} [\langle left overhang \rangle] [\langle right overhang \rangle]$
16 `\NewDocumentCommand{\LWR@HTML@cellcolor}{O{named} m o o}{%`
17 `\convertcolorspec{#1}{#2}{HTML}\LWR@cellHTMLcolor%`
18 `\LWR@addtabularcellcolor%`
19 }
20
21 `\AtBeginDocument{\LWR@formatted{cellcolor}}`

`\arrayrulecolor` $[\langle model \rangle] \{\langle color \rangle\}$

The HTML version for use outside a `tabular`. Inside a `tabular`, `\LWR@HTML@arrayrulecolornexttoken` is used instead.

```
22 \newcommand{\LWR@HTML@arrayrulecolor}[2][named]{%
```

```

23 \convertcolorspec{#1}{#2}{HTML}\LWR@ruleHTMLcolor%
24 }
25
26 \AtBeginDocument{\LWR@expandableformatted{arrayrulecolor}}

```

[*model*] {*color*}

The HTML version for use inside a tabular.

```

27 \newcommand{\LWR@HTML@arrayrulecolornexttoken}[2][named]{%
28 \convertcolorspec{#1}{#2}{HTML}\LWR@ruleHTMLcolor%
29 \LWR@getmynexttoken%
30 }
31
32 \AtBeginDocument{\LWR@expandableformatted{arrayrulecolornexttoken}}

```

\doublerulesepcolor [*model*] {*color*}

The version for use outside a tabular.

```

33 \newcommand{\LWR@HTML@doublerulesepcolor}[2][named]{}
34
35 \AtBeginDocument{\LWR@expandableformatted{doublerulesepcolor}}

```

[*model*] {*color*}

The version for use inside a tabular.

```

36 \newcommand{\LWR@HTML@doublerulesepcolornexttoken}[2][named]{\LWR@getmynexttoken}
37
38 \AtBeginDocument{\LWR@expandableformatted{doublerulesepcolornexttoken}}

```

File 73 lwarf-continue.sty

§ 169 Package **continue**

Pkg **continue** *continue* is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{continue}

2 \newcommand*\{\flagcont}{}%
3 \newcommand*\{\flagend}{}%
4 \newcommand*\{\flagword}{}%
5 \newcommand*\{\preflagword}{}%
6 \newcommand*\{\postflagword}{}%
7 \newlength\contsep
8 \newlength\contdrop

```

File 74 **l warp-copyrightbox.sty**

§ 170 Package **copyrightbox**

(Emulates or patches code by THOMAS FISCHER, IVES VAN DER FLAAS.)

Pkg **copyrightbox** copyrightbox is emulated for use by l warp.

The entire copyright box is placed inside a <div> of class copyrightbox.

The contents are placed inside a <div> of class copyrightboxcontents.

The copyright notice is placed inside a <div> of class copyrightboxnote.

for HTML output: 1 \LWR@ProvidesPackageDrop{copyrightbox}

```
2 \newcommand{\copyrightbox}[3][r]{%
3 \begin{BlockClass}[
4   display: inline-flex;
5   flex-direction: column ;
6 ]\copyrightbox}
7 \begin{BlockClass}{copyrightboxcontents}
8 #2
9 \end{BlockClass}
10 \begin{BlockClass}{copyrightboxnote}
11 #3
12 \end{BlockClass}
13 \end{BlockClass}
14 }
15
16 \newcommand{\CRB@setcopyrightfont}{}
17 \newcommand{\CRB@setcopyrightparagraphstyle}{}
```

File 75 **l warp-crop.sty**

§ 171 Package **crop**

(Emulates or patches code by MELCHIOR FRANZ.)

Pkg **crop** Emulated.

for HTML output: Discard all options for l warp-crop:

```
1 \LWR@ProvidesPackageDrop{crop}
2 \newcommand*{\crop}[1][]{}
3 \newcommand*{\cropdef}[6][]{}
```

File 76 l warp-cuted.sty

§ 172 Package **cuted**

(Emulates or patches code by SIGITAS TOLUŠIS.)

Pkg cuted cuted is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{cuted}
2 \newenvironment{strip}{}{}
3 \newskip\stripsep
4 \def\oldcolsbreak#1{}
```

File 77 l warp-cutwin.sty

§ 173 Package **cutwin**

(Emulates or patches code by PETER WILSON AND ALAN HOENIG.)

Pkg cutwin Emulated.

for HTML output: Discard all options for l warp-cutwin:

```
1 \LWR@ProvidesPackageDrop{cutwin}
2 \newcommand*{\opencutleft}{}
3 \newcommand*{\opencutright}{}
4 \newcommand*{\opencutcenter}{}
5 \newcommand*{\cutfuzz}{}
6
7 \newenvironment{cutout}[4]
8 {\marginpar{\windowpagestuff}}
9 {}
10
11 \newcommand*{\windowpagestuff}{}
12
13 \newcommand*{\pageinwindow}{%
14 \% \begin{minipage}{.3\linewidth}}
```

```
15 \windowpagestuff
16 % \end{minipage}
17 }
18
19 \newenvironment{shapedcutout}[3]
20 {\marginpar{\picinwindow}}
21 {}
22
23 \newcommand*{\putstuffinpic}={}
24
25 \newcommand*{\picinwindow}{%
26 \begin{picture}(0,0)
27 \putstuffinpic
28 \end{picture}}
```

File 78 **l warp-dblfloatfix.sty**

§ 174 Package **dblfloatfix**

Pkg **dblfloatfix** **dblfloatfix** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{dblfloatfix}

File 79 **l warp-dblfnote.sty**

§ 175 Package **dblfnote**

(Emulates or patches code by HIROSHI NAKASHIMA.)

Pkg **dblfnote** **dblfnote** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{dblfnote}

```
2 \newcounter{DFNsloppiness}
3 \newdimen\DFNcolumnsep
4 \newdimen\DFNcolumnwidth
5 \def\DFNallowcbreak{}
6 \def\DFNinhibitcbreak{}
7 \def\DFNtrysingle{}
8 \def\DFNalwaysdouble{}
9 \def\DFNruleboth{}
10 \def\DFNruleleft{}
```

File 80 **l warp-dcolumn.sty**

§ 176 Package **dcolumn**

Pkg **dcolumn** **dcolumn** is emulated by the **l warp** core.

```
1 \LWR@ProvidesPackageDrop{dcolumn}
```

File 81 **l warp-diagbox.sty**

§ 177 Package **diagbox**

(Emulates or patches code by LEO LIU.)

Pkg **diagbox** **diagbox** is patched for use by **l warp**.

for HTML output: 1 \LWR@ProvidesPackagePass{diagbox}

To restore print-mode inside a **lateximage**:

```
2 \LetLtxMacro{\LWR@origdiagbox@double}{\diagbox@double}
3 \LetLtxMacro{\LWR@origdiagbox@triple}{\diagbox@triple}
4
5 \appto{\LWR@restoreorigformatting}{%
6 \LetLtxMacro{\diagbox@double}{\LWR@origdiagbox@double}%
7 \LetLtxMacro{\diagbox@triple}{\LWR@origdiagbox@triple}%
8 }
```

```
\LWR@diagbox@AB {<E/W>} {<A>} {<E/W>} {<B>}
9 \newcommand{\LWR@diagbox@AB}[4]{%
10 \begingroup%
11 \LetLtxMacro{\\\newline}{%
12 \BlockClassSingle{\diagbox#1}{#2}%
13 \BlockClassSingle{\diagbox#3}{#4}%
14 \endgroup%
15 \LWR@stopars%
16 }
```

```
\LWR@diagboxNW {<A>} {<B>}
```

```

17 \newcommand{\LWR@diagboxNW}[2]{%
18 \LWR@diagbox@AB{E}{#2}{W}{#1}%
19 }

```

Likewise for NE, SW, SE:

```

20 \newcommand{\LWR@diagboxNE}[2]{%
21 \LWR@diagbox@AB{W}{#1}{E}{#2}%
22 }
23
24 \let\LWR@diagboxSW\LWR@diagboxNE
25 \let\LWR@diagboxSE\LWR@diagboxNW

```

```

\diagbox@double  {\langle keys\rangle} {\langle A\rangle} {\langle B\rangle}
26 \def\diagbox@double#1#2#3{%
27 \setkeys{diagbox}{dir=NW,#1}%
28 \nameuse{\LWR@diagbox\diagbox@dir}{#2}{#3}%
29 }

```

```

\LWR@diagboxTNW {\langle title\rangle} {\langle A\rangle} {\langle B\rangle}
30 \newcommand{\LWR@diagboxTNW}[3]{%
31 \BlockClassSingle{diagboxtitleN}{#1}%
32 \LWR@diagboxNW{#2}{#3}%
33 }

```

Likewise for NE, SW, SE:

```

34 \newcommand{\LWR@diagboxTNE}[3]{%
35 \BlockClassSingle{diagboxtitleN}{#1}%
36 \LWR@diagboxNE{#2}{#3}%
37 }
38
39 \newcommand{\LWR@diagboxTSW}[3]{%
40 \LWR@diagboxSW{#2}{#3}%
41 \BlockClassSingle{diagboxtitleS}{#1}%
42 \LWR@stopars%
43 }
44
45 \newcommand{\LWR@diagboxTSE}[3]{%
46 \LWR@diagboxSE{#2}{#3}%
47 \BlockClassSingle{diagboxtitleS}{#1}%
48 \LWR@stopars%
49 }

```

```

\diagbox@triple {\langle keys\rangle} {\langle A\rangle} {\langle T\rangle} {\langle B\rangle}

```

```
50 \def\diagbox@triple#1#2#3#4{%
51 \setkeys{diagbox}{dir=NW,#1}%
52 \Cnameuse{LWR@diagboxT\diagbox@dir}{#3}{#2}{#4}%
53 }
```

File 82 **l warp-dprogress.sty**

§ 178 Package **dprogress**

Pkg **dprogress** dprogress is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{dprogress} [2008/02/21]

File 83 **l warp-draftcopy.sty**

§ 179 Package **draftcopy**

Pkg **draftcopy** draftcopy is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{draftcopy}

```
2 \newcommand{\draftcopyVersion}[1]{}
3 \newcommand{\draftcopySetGrey}[1]{}
4 \newcommand{\draftcopysetScale}[1]{}
5 \newcommand{\draftcopySetScaleFactor}[1]{}
6 \newcommand{\draftcopyFirstPage}[1]{}
7 \newcommand{\draftcopyLastPage}[1]{}
8 \newcommand{\draftcopyName}[2]{}
9 \newcommand{\draftcopyPageTransform}[1]{}
10 \newcommand{\draftcopyBottomTransform}[1]{}
11 \newcommand{\draftcopyPageX}[1]{}
12 \newcommand{\draftcopyPageY}[1]{}
13 \newcommand{\draftcopyBottomX}[1]{}
14 \newcommand{\draftcopyBottomY}[1]{}
```

File 84 **l warp-draftfigure.sty**

§ 180 Package **draftfigure**

Pkg **draftfigure** draftfigure is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{draftfigure}[2017/07/19]
2 \RequirePackage{xkeyval}

3 \define@key{draftfigure}{code}={}
4 \define@key{draftfigure}{noframe}[true]{}
5 \define@key{draftfigure}{filename}[true]{}
6 \define@key{draftfigure}{content}[]{}
7 \define@key{draftfigure}{style}[normal]{}
8 \define@key{draftfigure}{position}[left]{}
9 \define@key{draftfigure}{size}[normal]{}
10 \newcommand\setdf[1]{\setkeys{draftfigure}{#1}}
```

File 85 **l warp-draftwatermark.sty**

§ 181 Package **draftwatermark**

(Emulates or patches code by SERGIO CALLEGARI.)

Pkg **draftwatermark** draftwatermark is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{draftwatermark}

2 \newcommand{\SetWatermarkAngle}[1]{}
3 \newcommand{\SetWatermarkColor}[1]{}
4 \newcommand{\SetWatermarkLightness}[1]{}
5 \newcommand{\SetWatermarkFontSize}[1]{}
6 \newcommand{\SetWatermarkScale}[1]{}
7 \newcommand{\SetWatermarkHorCenter}[1]{}
8 \newcommand{\SetWatermarkVertCenter}[1]{}
9 \newcommand{\SetWatermarkText}[1]{}
```

File 86 **l warp-easy-todo.sty**

§ 182 Package **easy-todo**

(Emulates or patches code by JUAN RADA-VILELA.)

Pkg **easy-todo** easy-todo is patched for use by l warp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{easy-todo}
```

\listoftodos Modified to correct buggy use of \flushright.

```

2 \let\LWR@origlistoftodos\listoftodos
3
4 \renewcommand{\listoftodos}{%
5 \begingroup
6 \renewcommand{\flushright}{}
7 \LWR@origlistoftodos
8 \endgroup
9 }

\todoii Modified to use \textcolor instead of \color.
10 \renewcommand{\todoii}[2]{%
11 \ifthenelse{\equal{\@todoobeyfinal}{true}}{%
12   \ifoptionfinal{\todoenable{false}}{\todoenable{true}}{%
13 }{}}%
14 \ifthenelse{\equal{\@todoenable}{true}}{%
15 \refstepcounter{todos}%
16 \noindent{%
17   \todocolor{%
18     \LWR@textcurrentcolor{%
19       \normalfont\scriptsize{\bfseries{\thetodos.\#1}}%
20     }%
21 }%
22 \addcontentsline{lod}{todos}{\protect{\thetodos. }\LWR@isolate{\#2}}%
23 }{}}%
24 }

```

File 87 **lwarf-ebook.sty**

§ 183 Package **ebook**

(Emulates or patches code by JØRGEN STEENSGAARD.)

Pkg **ebook** ebook is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{ebook}

2 \setcounter{secnumdepth}{0}
3 \setcounter{tocdepth}{2}
4
5 \providecommand{\pagefill}[1][0.001mm]{\noindent}
6
7 \providecommand{\ebook}{%
8 \setcounter{secnumdepth}{0}
9 \setcounter{tocdepth}{2}
10 }

```

File 88 **l warp-ellipsis.sty**

§ 184 Package **ellipsis**

(Emulates or patches code by PETER J. HESLIN.)

Pkg **ellipsis** ellipsis is emulated.

```

1 \LWR@ProvidesPackageDrop{ellipsis}
2
3 \newcommand{\ellipsisgap}{0.1em}
4
5 \newcommand*{\midwordellipsis}{\textellipsis}
```

File 89 **l warp-embrac.sty**

§ 185 Package **embrac**

(Emulates or patches code by CLEMENS NIEDERBERGER.)

Pkg **embrac** embrac is nullified for HTML and used as-is for print.

for HTML output:

```

1 \LWR@ProvidesPackagePass{embrac}

2 \LetLtxMacro{\LWR@orig@HTML@emph}{\LWR@HTML@emph}
3 \RenewDocumentCommand{\LWR@HTML@emph}{s m}{\LWR@orig@HTML@emph{#2}}
4
5 \LetLtxMacro{\LWR@orig@HTML@textit}{\LWR@HTML@textit}
6 \RenewDocumentCommand{\LWR@HTML@textit}{s m}{\LWR@orig@HTML@textit{#2}}
7
8 \LetLtxMacro{\LWR@orig@HTML@textsl}{\LWR@HTML@textsl}
9 \RenewDocumentCommand{\LWR@HTML@textsl}{s m}{\LWR@orig@HTML@textsl{#2}}
10
11 \ifxetexorluatex
12     \LetLtxMacro{\LWR@orig@HTML@textsi}{\LWR@HTML@textsi}
13     \RenewDocumentCommand{\LWR@HTML@textsi}{s m}{%
14         \LWR@orig@HTML@textsi{#2}}
15 \fi
16
17 \AtBeginDocument{
18     \LWR@formatted{emph}
19     \LWR@formatted{textit}}
```

```

20      \LWR@formatted{textsl}
21      \ifxetexorluatex
22          \LWR@formatted{textsi}
23      \fi
24 }
25
26 \newcommand{\LWR@HTML@EmbracOff}{}
27 \LWR@formatted{EmbracOff}
28
29 \newcommand{\LWR@HTML@EmbracOn}{}
30 \LWR@formatted{EmbracOn}

```

File 90 **l warp-emptypage.sty**

§ 186 Package **emptypage**

Pkg **emptypage** **emptypage** is ignored.

for HTML output: Discard all options for l warp-emptypage:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

File 91 **l warp-endfloat.sty**

§ 187 Package **endfloat**

Pkg **endfloat** **endfloat** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{endfloat}

```

2 \newcommand\figureplace{}
3 \newcommand\tableplace{}
4 \newcommand\floatplace[1]{}
5 \newcounter{posttable}
6 \newcounter{postfigure}
7 \newcommand*\theposttbl(){}
8 \newcommand*\thepostfig(){}
9 \newcommand{\AtBeginFigures}[1]{}
10 \newcommand{\AtBeginTables}[1]{}
11 \newcommand{\AtBeginDelayedFloats}[1]{}
12 \newcommand*\processdelayedfloats(){}
13 \newcommand*\efloatseparator(){}
14 \def\efloattype{}
15 \providecommand\efloatheading[1]{}

```

```
16 \providecommand\efloatpreamble{  
17 \providecommand\efloatpostamble{  
}
```

File 92 **lwarf-endheads.sty**

§ 188 Package **endheads**

Pkg **endheads** **endheads** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{endheads}

```
2 \newcommand{\changesinglepageabbrev}[1]{  
3 \newcommand{\changemultiplepageabbrev}[1]{  
4 \newcommand{\changetitlesname}[1]{  
5 \newcommand{\changetitlesheader}[1]{  
6 \newcommand{\changetitlescontentsname}[1]{  
7 \newcommand{\changechaptertitlesline}[1]{  
8 \newcommand{\checknoteheaders}{}  
9 \newif\ifnotesincontents\notesincontentsfalse  
10 \newcommand{\notesincontents}{\notesincontentstrue}  
11 \newif\ifendnoteheaders\endnoteheadersfalse  
12 \newcommand{\setupendnoteheaders}{{%  
13     \endnoteheaderstrue%  
14 }  
15 \newif\iftitleinnotes\titleinnotestrue  
16 \newcommand{\styleforchapternotebegin}{}  
17 \newcommand{\styleforchapternoteend}{}  
18 \newcommand{\setstyleforchapternotebegin}[1]{%  
19     \renewcommand{\styleforchapternotebegin}{#1}%  
20 }  
21 \newcommand{\setstyleforchapternoteend}[1]{%  
22     \renewcommand{\styleforchapternoteend}{#1}%  
23 }  
24 \newcommand{\resetendnotes}{}  
25 \newif\ifnotesbychapter\notesbychapterfalse  
26 \newcommand{\notesbychapter}{\notesbychaptertrue}
```

File 93 **lwarf-endnotes.sty**

§ 189 Package **endnotes**

(Emulates or patches code by JOHN LAVAGNINO.)

Pkg **endnotes** Used as-is.

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

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

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

```
\ForceHTMLPage
\theendnotes
```

for HTML output:

```
1 \LWR@ProvidesPackagePass{endnotes}

2 \def\enoteformat{%
3   \rightskip\z@ \leftskip\z@ \parindent=1.8em
4   \leavevmode
5   \llap{
6     \makeenmark
7   }
8 }
9
10 \def\@makeenmark{\hbox{\LWR@htmlspan{sup}{\normalfont\theenmark}}}
11 \def\makeenmark{\@makeenmark}
```

File 94 **lwarf-enumerate.sty**

§ 190 Package **enumerate**

Pkg **enumerate** `enumerate` is supported with no changes.

This package is only required because it was used in the past to drop and then emulate the package. It cannot be removed because an older version which dropped the package may still remain, for example in a local vs. distribution directory, but it is now supported directly by `lwarf` and thus must no longer be dropped.

for HTML output:

```
1 \LWR@ProvidesPackagePass{enumerate}
```

File 95 **lwarf-enumitem.sty**

§ 191 Package **enumitem**

(Emulates or patches code by JAVIER BEZOS.)

Pkg **enumitem** `enumitem` is supported with minor adjustments.

```

for HTML output: 1 \LWR@ProvidesPackagePass{enumitem}

for HTML output: 2 \begin{warpHTML}

    \newlist {⟨name⟩} {⟨type⟩} {⟨maxdepth⟩}
    \renewlist {⟨name⟩} {⟨type⟩} {⟨maxdepth⟩}

```

For enumitem lists, new lists must have the start and end actions assigned to the new environment. Renewed lists already have their actions assigned, and thus need no changes.

```

3 \let\LWR@enumitem@orignewlist\newlist
4
5 \renewcommand*\newlist[3]{%
6 \LWR@enumitem@orignewlist[#1]{#2}{#3}%
7 \AtBeginEnvironment{#1}{\@nameuse{\LWR@#2start}}%
8 \AtEndEnvironment{#1}{\@nameuse{\LWR@#2end}}%
9 }
10
11 \def\DrawEnumitemLabel{}

12 \end{warpHTML}

```

File 96 lwarf-epigraph.sty

§ 192 Package **epigraph**

(Emulates or patches code by PETER WILSON.)

Pkg epigraph epigraph is emulated.

```

for HTML output: 1 \LWR@ProvidesPackageDrop{epigraph}

    2 \DeclareDocumentCommand{\qitem}{m m}
    3 {
    4 \begin{BlockClass}{qitem}
    5 #1
    6 \ifbool{FormatWP}
    7 {\begin{BlockClass}[border-top:1px solid gray]{epigraphsource}}
    8 {\begin{BlockClass}{epigraphsource}}
    9 #2
    10 \end{BlockClass}
    11 \end{BlockClass}
    12 }

```

```

13 \DeclareDocumentCommand{\epigraph}{m m}
14 {
15 \begin{LWR@BlockClassWP}{\LWR@print@mbox{text-align:right}}{}{epigraph}
16 \qitem[#1]{#2}
17 \end{LWR@BlockClassWP}
18 }
19
20 \DeclareDocumentEnvironment{epigraphs} {}
21 {\LWR@BlockClassWP{\LWR@print@mbox{text-align:right}}{}{epigraph}}
22 {\endLWR@BlockClassWP}

```

Use css to format epigraphs.

The following are null commands for source compatibility:

```

23 \newenvironment*{flushepinormal}{}{}

24 \@ifclassloaded{memoir}{
25 \setlength{\epigraphwidth}{.5\linewidth}
26 \renewcommand{\textflush}{flushepinormal}
27 \renewcommand{\epigraphhead}[2][0]{#2}
28 \renewcommand{\dropchapter}[1]{}
29 \renewcommand*{\undodrop}){}
30 }% not memoir
31 \newlength{\epigraphwidth}
32 \setlength{\epigraphwidth}{.5\linewidth}
33 \newcommand{\textflush}{flushepinormal}
34 \newcommand{\epigraphflush}{flushright}
35 \newcommand{\sourceflush}{flushright}
36 \newcommand*{\epigraphsize}{\small}
37 \newlength{\epigraphrule}
38 \newlength{\beforeepigraphskip}
39 \newlength{\afterepigraphskip}
40 \newcommand{\epigraphhead}[2][0]{#2}
41 \newcommand{\dropchapter}[1]{}
42 \newcommand*{\undodrop}){}
43 }% not memoir
44
45 \let\cleartoevenpage\relax% also in nextpage
46 \newcommand{\cleartoevenpage}[1][]{}

```

File 97 lwarf-epsfig.sty

§ 193 Package **epsfig**

Pkg **epsfig** **epsfig** is emulated for use by **lwarf**.

 Only the L^AT_EX2e syntax is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{epsfig}[2017/06/25]

A few additional keys to capture the filename:

```

2 \RequirePackage{graphics}
3
4 \define@key{igraph}{file}{%
5   \xdef\LWR@epsfig@filename{\#1}%
6 }
7
8 \define@key{igraph}{figure}{%
9   \xdef\LWR@epsfig@filename{\#1}%
10 }
11
12 \define@key{igraph}{prolog}{}
13
14 \define@key{igraph}{silent}{}[]{}

```

The captured filename is used as the argument to \includegraphics:

```

15 \newcommand{\epsfig}[1]{\includegraphics[\#1]{\LWR@epsfig@filename}}
16
17 \newcommand{\psfig}[1]{\includegraphics[\#1]{\LWR@epsfig@filename}}

```

File 98 lwarf-epstopdf.sty

§ 194 Package **epstopdf**

Pkg epstopdf Previous versions of lwarf had a nullified verison, but now epstopdf-base is supported. lwarf-epstopdf becomes a placeholder to overwrite previous versions.

See package epstopdf-base for details.

for HTML output: 1 \LWR@ProvidesPackagePass{epstopdf}[2016/05/15]

File 99 lwarf-epstopdf-base.sty

§ 195 Package **epstopdf-base**

Pkg epstopdf-base

⚠ convert to .svg

Images with an .eps extension will be converted to .pdf. The HTML output uses the .svg version, so use

Enter ⇒ lwarfpmk pdftosvg <listofPDFfiles>

to generate .svg versions.

for HTML output: 1 \LWR@ProvidesPackagePass{epstopdf-base}[2016/05/15]

Redefine to remember the image filename, replacing .pdf with .svg. Use the epstopdf print version inside a `lateximage`.

```
2 \newcommand*{\LWR@HTML@ETE@OrgGin@setfile}[3]{%
3   \edef\tempone{\#3}%
4   \StrSubstitute{\tempone}{.pdf}{.svg}[\tempone]%
5   \StrSubstitute{\tempone}{.PDF}{.SVG}[\tempone]%
6   \xdef\parsedfilename{\tempone}%
7 }%
8
9 \LWR@formatted{ETE@OrgGin@setfile}
```

\includegraphics in HTML mode redefines \Gin@setfile to be \LWR@HTML@Gin@setfile, which is now redirected to epstopdf's version:

```
10 \renewcommand*{\LWR@HTML@Gin@setfile}[3]{%
11   \ETE@Gin@setfile{\#1}{\#2}{\#3}%
12 }
```

Allow .eps images to be found if a suffix is not provided:

```
13 \AtBeginDocument{
14 \DeclareGraphicsExtensions{%
15   .eps,.EPS,.svg,.SVG,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG%
16 }
17 \DeclareGraphicsRule{.svg}{svg}{.svg}{}
18 \DeclareGraphicsRule{.SVG}{svg}{.SVG}{}
19 }
```

Likewise when inside a `lateximage`:

```
20 \appto{\LWR@restoreorigformatting}{%
21 \DeclareGraphicsExtensions{%
22   .eps,.EPS,.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG%
23 }%
24 }
```

File 100 **lwarf-errata.sty**

§ 196 Package **errata**

(Emulates or patches code by MICHAEL KOHLHASE.)

Pkg **errata** errata is patched for use by lwarf.

This is for v0.3 of errata. A newer version of errata with more features is under development, at which time the lwarf version will have to be updated.

for HTML output: Macros are being defined with the math dollar, so enable the HTML version during package loading:

1 \StartDefiningMath

Now load the package:

2 \LWR@ProvidesPackagePass{errata}[2006/11/12]

Patches for dynamic inline math:

```

3 \xpatchcmd{\erratumAdd}
4   {$_a^{\arabic{erratum}}$}
5 %   {\inlinemathother$_a^{\arabic{erratum}}$\inlinemathnormal}
6   {\textsubscript{a}\textsuperscript{\arabic{erratum}}}
7   {}
8   {\LWR@patcherror{erratum}{erratumAdd}}
9

10 \xpatchcmd{\erratumDelete}
11   {$_d^{\arabic{erratum}}$}
12 %   {\inlinemathother$_d^{\arabic{erratum}}$\inlinemathnormal}
13   {\textsubscript{d}\textsuperscript{\arabic{erratum}}}
14   {}
15   {\LWR@patcherror{erratum}{erratumDelete}}
16

17 \xpatchcmd{\erratumReplace}
18   {$_r^{\arabic{erratum}}$}
19 %   {\inlinemathother$_r^{\arabic{erratum}}$\inlinemathnormal}
20   {\textsubscript{r}\textsuperscript{\arabic{erratum}}}
21   {}
22   {\LWR@patcherror{erratum}{erratumReplace}}
23

24 \xpatchcmd{\erratum}
25   {$_a$}

```

```

26 %      {\inlinemathother{_a$\inlinemathnormal}
27     {\textsubscript{a}}}
28     {}
29     {\LWR@patcherror{erratum}{erratumDelete}}
30
31 \xpatchcmd{\erratum}
32   {$_d^{\@thefnmark}$$}
33 %      {\inlinemathother{_d^{\@thefnmark} $\inlinemathnormal}
34     {\textsubscript{d}\textsuperscript{\@thefnmark}}}
35     {}
36     {\LWR@patcherror{erratum}{eDelete}}
37
38 \xpatchcmd{\erratum}
39   {$_r^{\@thefnmark}$$}
40 %      {\inlinemathother{_r^{\@thefnmark} $\inlinemathnormal}
41     {\textsubscript{r}\textsuperscript{\@thefnmark}}}
42     {}
43     {\LWR@patcherror{erratum}{eReplace}}

```

Finish the current page's errata before closing and reloading the list:

```
44 \preto\PrintErrata{\LWR@orignewpage}
```

No longer defining math macros with the HTML \$:

```
45 \StopDefiningMath
```

File 101 **lwarf-eso-pic.sty**

§ 197 Package **eso-pic**

(Emulates or patches code by ROLF NIEPRASCHK.)

Pkg **eso-pic** **eso-pic** is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{eso-pic}

2 \newcommand*\LenToUnit(){}
3 \newcommand{\AtPageUpperLeft}[1]{}
4 \newcommand{\AtPageLowerLeft}[1]{}
5 \newcommand{\AtPageCenter}[1]{}
6 \newcommand{\AtStockLowerLeft}[1]{}
7 \newcommand{\AtStockUpperLeft}[1]{}
8 \newcommand{\AtStockCenter}[1]{}
9 \newcommand{\AtTextUpperLeft}[1]{}
10 \newcommand{\AtTextLowerLeft}[1]{}

```

```
11 \newcommand{\AtTextCenter}[1]{}
12 \NewDocumentCommand{\AddToShipoutPictureBG}{s +m}{}

13 \newcommand{\AddToShipoutPicture}{\AddToShipoutPictureBG}
14 \NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
15 \newcommand*{\ClearShipoutPictureBG}={}
16 \newcommand*{\ClearShipoutPicture}={}
17 \newcommand*{\ClearShipoutPictureFG}={}
18 \newcommand{\gridSetup}[6][]{}
```

File 102 **l warp-everypage.sty**

§ 198 Package **everypage**

(Emulates or patches code by SERGIO CALLEGARI.)

Pkg **everypage** everypage is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{everypage}

2 \newcommand*{\AddEverypageHook}[1]{}
3 \newcommand*{\AddThispageHook}[1]{}
```

File 103 **l warp-everyshi.sty**

§ 199 Package **everyshi**

(Emulates or patches code by MARTIN SCHRÖDER.)

Pkg **everyshi** Emulated.

for HTML output: Discard all options for l warp-everyshi:

```
1 \LWR@ProvidesPackageDrop{everyshi}

2 \newcommand*{\EveryShipout}[1]{}
3 \newcommand*{\AtNextShipout}[1]{}
```

File 104 l warp-extramarks.sty**§ 200 Package extramarks**

(Emulates or patches code by PIET VAN OOSTRUM.)

Pkg **extramarks** extramarks is emulated.

for HTML output: Discard all options for l warp-extramarks:

```
1 \LWR@ProvidesPackageDrop{extramarks}

2 \newcommand*{\extramarks}[2]{}
3 \newcommand*{\firstleftxmark}{}
4 \newcommand*{\lastleftxmark}{}
5 \newcommand*{\firstrightxmark}{}
6 \newcommand*{\lastrightxmark}{}
7 \newcommand*{\firstxmark}{}
8 \newcommand*{\lastxmark}{}
9 \newcommand*{\topxmark}{}
10 \newcommand*{\topleftxmark}{}
11 \newcommand*{\firstleftmark}{}
12 \newcommand*{\lastrightmark}{}
```

File 105 l warp-fancybox.sty**§ 201 Package fancybox**

(Emulates or patches code by TIMOTHY VAN ZANDT.)

Pkg **fancybox** fancybox is supported with some patches.

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

```
\newenvironment{FramedEqn}
{
  \latextimage% NEW
  \setlength{\fboxsep}{15pt}
  ...{...
  \[\fbox{\TheSbox}\]}
  \endlatextimage% NEW
}
```

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

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

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

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

```
\newenvironment{FramedVerb}[1] % width
{
  \VerbatimEnvironment
  \fminipage{#1}
  \begin{Verbatim}
  \end{Verbatim}
  \endfminipage
}
```

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

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

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

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

and likewise for equations or display math.

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

```
1 \LWR@ProvidesPackagePass{fancybox}
```

After the preamble is loaded, after any patches to Verbatim:

```
2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching fancybox.}
```

\VerbatimFootnotes Patched to use the new version.

```
4 \def\VerbatimFootnotes{%
5 \let\@footnotetext\V@footnotetext%
6 \let\LWR@footnotetext\V@footnotetext% lwarp
7 }
```

\V@footnotetext Patches in a subset of lwarp's \LWR@footnotetext to the fancyvrb version of \V@footnotetext.

```
8 \def\V@footnotetext{%
9 \LWR@traceinfo{V@footnotetext}%
10 \global\setbox\LWR@footnotes=\vbox\bgroup%
```

Add to any current footnotes:

```
11 \unvbox\LWR@footnotes%
```

Remember the footnote number for \ref:

```
12 \protected@edef@\currentlabel{%
13   \csname p@footnote\endcsname\@thefnmark%
14 }% @currentlabel
```

Use HTML superscripts in the footnote even inside a lateximage:

```
15 \renewrobustcmd{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

Use paragraph tags if in a tabular data cell or a lateximage:

```
16 \ifthenelse{%
17   \boolean{\LWR@doingstartpars} \AND%
18   \cnttest{\value{\LWR@lateximagedepth}}{=}{0}%
19 }%
20 {}%
21 {\LWR@htmltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```

22     \@makefntext{}%
23     \bgroup%
24     \aftergroup{\V@@@footnotetext}%
25     \ignorespaces%
26 }%
27 }% AfterEndPreamble

28 \renewcommand*{\@shadowbox}[1]{%
29 \ifbool{FormatWP}{%
30 {\InlineClass[border:1px solid black]{shadowbox}{#1}}%
31 {\InlineClass{shadowbox}{#1}}%
32 }%
33
34 \renewcommand*{\@doublebox}[1]{%
35 \ifbool{FormatWP}{%
36 {\InlineClass[border:1px double black]{doublebox}{#1}}%
37 {\InlineClass{doublebox}{#1}}%
38 }%
39
40 \renewcommand*{\@ovalbox}[2]{%
41 \ifbool{FormatWP}{%
42 {\InlineClass[border:1px solid black; border-radius:1ex]{ovalbox}{#2}}%
43 }%
44     \ifthenelse{\isequivalentto{#1}{\thinlines}}{%
45         {\InlineClass{ovalbox}{#2}}%
46         {\InlineClass{Ovalbox}{#2}}%
47 }%
48 }

```

Convert minipages, parboxes, and lists into linear text using the LWR@nestspan environment:

```

49 \let\LWR@origSbox\Sbox
50
51 \def\Sbox{\LWR@origSbox\LWR@nestspan}
52
53
54 \let\LWR@origendSbox\endSbox
55
56 \def\endSbox{\endLWR@nestspan\LWR@origendSbox}

```

Beqnarray is adapted for MATHJAX or enclosed inside a `latexitimage`:

```

57 \RenewEnviron{Beqnarray}{%
58 {\LWR@eqnarrayfactor}}

```

```

59
60 \csgpreto{Beqnarray}{\boolfalse{LWR@numbereqnarray}}

```

\GenericCaption is enclosed in an HTML block:

```

61 \renewcommand{\GenericCaption}[1]{%
62 \LWR@figcaption{%
63 \LWR@isolate{#1}%
64 \endLWR@figcaption{%
65 }%

```

Btrivlist is enclosed in an HTML block. This is a tabular, and does not use \item.

```

\trivlist {\langle l/c/r\rangle} [\langle t/c/b\rangle]
66 \RenewDocumentEnvironment{Btrivlist}{m o}
67 {%
68   \begin{BlockClass}{Btrivlist}%
69   \tabular{#1}%
70 }%
71 {%
72   \endtabular%
73   \end{BlockClass}%
74 }%

```

Btrivlist is also neutralized when used inside a span:

```

75 \AtBeginEnvironment{LWR@nestspan}{%
76 \RenewDocumentEnvironment{Btrivlist}{m o}{}{}%
77 }%

```

lwarp's handling of \item is patched to accept fancybox's optional arguments:

```

78 \let\LWRFB@origitemizeitem\LWR@itemizeitem
79 \let\LWRFB@origdescitem\LWR@descitem
80
81 \RenewDocumentCommand{\LWR@itemizeitem}{d()o}{%
82   \IfValueTF{#2}{%
83     \LWRFB@origitemizeitem[#2]%
84   }{%
85     \LWRFB@origitemizeitem%
86   }%
87 }%
88
89 \RenewDocumentCommand{\LWR@descitem}{d()o}{%
90   \IfValueTF{#2}{%
91     \LWRFB@origdescitem[#2]~%
92   }{%

```

```

93           \LWRFB@origdescitem%
94       }%
95 }

96 \RenewDocumentCommand{\LWR@nestspanitem}{d()}{%
97 \if@newlist\else{\LWR@htmllagc{br /}}\fi%
98 \LWR@origitem%
99 }

```

The various boxed lists become regular lists:

```

100 \renewenvironment{Bitemize}[1][] {\begin{itemize}}{\end{itemize}}
101 \renewenvironment{Benumerate}[1][] {\begin{enumerate}}{\end{enumerate}}
102 \renewenvironment{Bdescription}[1][] {\begin{description}}{\end{description}}

```

\boxput simply prints one then the other argument, side-by-side instead of above and behind:

```

103 \RenewDocumentCommand{\boxput}{s d() m m}{%
104 \IfBooleanTF{#1}{#3\quad#4}{#4\quad#3}%
105 }

```

Neutralized commands:

```

106 \RenewDocumentCommand{\fancyput}{s d() m m}{}
107 \RenewDocumentCommand{\thisfancyput}{s d() m m}{}
108
109 \RenewDocumentCommand{\fancypage}{m m m}{}
110 \RenewDocumentCommand{\thisfancypage}{m m m}{}
111
112 \def\LandScape#1{}
113 \def\endLandScape{}
114 \def\@Landscape#1#2#3{}
115 \def\endLandscape{}

```

Low-level patches for `UseVerbatim` and friends:

```

116 \let\LWRFB@UseVerbatim\UseVerbatim
117 \renewcommand*{\UseVerbatim}[1]{%
118 \LWR@atbeginverbatim{3}{Verbatim}%
119 \LWRFB@UseVerbatim{#1}%
120 \LWR@afterendverbatim{.5}%
121 }
122
123 \let\LWRFB@LUseVerbatim\LUseVerbatim
124
125 \renewcommand*{\LUseVerbatim}[1]{%

```

```
126 \LWR@atbeginverbatim{3}{LVerbatim}%
127 \noindent%
128 \LWRFB@LUseVerbatim{#1}%
129 \LWR@afterendverbatim{.5}%
130 }
131
132 \def\@BUseVerbatim[#1]#2{%
133 \LWR@atbeginverbatim{3}{BVerbatim}%
134 \LWRFB@UseVerbatim{#2}%
135 \LWR@afterendverbatim{.5}%
136 }
```

File 106 **l warp-fancyhdr.sty**

§ 202 Package **fancyhdr**

(Emulates or patches code by PIET VAN OOSTRUM.)

Pkg **fancyhdr** fancyhdr is nullified.

for HTML output: Discard all options for l warp-fancyhdr:

```
1 \LWR@ProvidesPackageDrop{fancyhdr}

2 \newcommand*\fancyhead[2][]{}
3 \newcommand*\fancyfoot[2][]{}
4 \newcommand*\fancyhf[2][]{}
5 \newcommand*\fancypagestyle[2][]{}
6 \newcommand*\lhead[2][]{}
7 \newcommand*\chead[2][]{}
8 \newcommand*\rhead[2][]{}
9 \newcommand*\lfoot[2][]{}
10 \newcommand*\cfoot[2][]{}
11 \newcommand*\rfoot[2][]{}
12 \newcommand*\headrulewidth{}%
13 \newcommand*\footrulewidth{}%
14 \newcommand*\fancyheadoffset[2][]{}
15 \newcommand*\fancyfootoffset[2][]{}
16 \newcommand*\fancyhfoffset[2][]{}
17 \newcommand*\iffloatpage[2]{#2}
18 \newcommand*\ifftopfloat[2]{#2}
19 \newcommand*\iffbotfloat[2]{#2}
```

File 107 **l warp-fancyref.sty**

§ 203 Package **fancyref**

Pkg **fancyref** fancyref is emulated.

for HTML output: 1 \LWR@ProvidesPackagePass{fancyref}

To remove the margin option, if \fancyrefhook is anything other than the paren option, then force it to the default instead. (Comparing to the margin option was not possible since l warp has revised the meaning of \mbox so the comparison failed.)

```

2 \newcommand*{\LWRfref@parenfancyrefhook}[1]{(#1)}
3
4 \ifdef\streq{\fancyrefhook}{\LWRfref@parenfancyrefhook}
5 {}{
6   \renewcommand*{\fancyrefhook}[1]{#1}%
7 }
```

Modified to ignore the page number and varioref.

```

8 \renewcommand*{\@f@ref}[4]{%
9   \@ifundefined{#1r@#2@#3}{%
10     \PackageError{fancyref}{%
11       \backslash\!backslash\!char\#1ref\space format ``#2''%
12       undefined\MessageBreak
13       for label type ``#3''%
14     }{%
15       The format ``#2'' was not defined for the label type
16       ``#3''\MessageBreak
17       and the \backslash\!backslash\!char\#1ref\space command. Perhaps
18       you have only misspelled its name.\MessageBreak
19       Otherwise you will have to define it with
20       \protect\new#1refformat\MessageBreak
21       prior to using it.%
22     }%
23   }{%
24     \fancyrefhook{%
25       \@nameuse{#1r@#2@#3}%
26       {\ref{#3\fancyrefargdelim#4}}%
27 %       {\pageref{#3\fancyrefargdelim#4}}% original
28 %       {\@fancyref@page@ref{#3\fancyrefargdelim#4}}% original
29       {}% l warp
30       {}% l warp
31     }%
```

```
32 }%
33 }%
```

File 108 **l warp-fancytabs.sty**

§ 204 Package **fancytabs**

Pkg **fancytabs** fancytabs is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{fancytabs}

2 \newcommand{\fancytab}[3][RIGHT]{}
3 \newcommand{\fancytabsStyle}[1]{}
4 \newcommand{\fancytabsHeight}[1]{}
5 \newcommand{\fancytabsWidth}[1]{}
6 \newcommand{\fancytabsCount}[1]{}
7 \newcommand{\fancytabsLeftColor}[1]{}
8 \newcommand{\fancytabsRightColor}[1]{}
9 \newcommand{\fancytabsTop}[1]{}
10 \newcommand{\fancytabsTextVPos}[1]{}
11 \newcommand{\fancytabsTextHPos}[1]{}
12 \newcommand{\fancytabsGap}[1]{}
13 \newcommand{\fancytabsFloor}[1]{}
14 \newcommand{\fancytabsRotate}[1]{}
```

File 109 **l warp-fancyvrb.sty**

§ 205 Package **fancyvrb**

(Emulates or patches code by TIMOTHY VAN ZANDT.)

Pkg **fancyvrb** fancyvrb is supported with some patches.

⚠ **\VerbatimFootnotes**
⚠ sectioning or
displaymath

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

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

and likewise for equations or display math.

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

```
1 \RequirePackage{xcolor}%
2 for \convertcolorspec
3 \LWR@ProvidesPackagePass{fancyvrb}
```

Initial default patch for fancyvrb:

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

After the preamble is loaded, after any patches to Verbatim:

```
5 \AfterEndPreamble{
6 \LWR@traceinfo{Patching fancyvrb.}}
```

`\VerbatimFootnotes` Patched to use the new version.

```
7 \def\VerbatimFootnotes{%
8 \let\@footnotetext\V@footnotetext%
9 \let\footnote\V@footnote%
10 \let\LWR@footnotetext\V@footnotetext% l warp
11 }
```

`\V@@footnotetext` Patches in a subset of l warp's `\LWR@footnotetext` to the fancyvrb version of `\V@footnotetext`.

```
12 \def\V@@footnotetext{%
13 \LWR@traceinfo{V@footnotetext}%
14 \global\setbox\LWR@footnotes=\vbox\bgroup%
```

Add to any current footnotes:

```
15 \unvbox\LWR@footnotes%
```

Remember the footnote number for `\ref`:

```
16 \protected@edef@\currentlabel{%
17 \csname p@footnote\endcsname\@thefnmark%
18 }% @currentlabel
```

Use HTML superscripts in the footnote even inside a `lateximage`:

```
19 \renewrobustcmd{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

Use paragraph tags if in a tabular data cell or a `lateximage`:

```
20 \ifthenelse{%
21 \boolean{\LWR@doingstartpars} \AND%
22 \cnttest{\value{\LWR@lateximagedepth}}{=}{0}%
23 }%
```

```
24      {}%
25      {\LWR@htmltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
26      \@makefntext{}%
27      \bgroup%
28      \aftergroup{\V@@@footnotetext}%
29      \ignorespaces%
30 }%
31 \preto\FVB@Verbatim{\LWR@forcenewpage}
32 \preto\FVB@LVerbatim{\LWR@forcenewpage}
33 % \preto\FVB@BVerbatim{\LWR@forcenewpage}% Fails, so done below.
```

Simplified to remove PDF formatting:

```
34 \def\FV@BeginListFrame@Single{%
35   \FV@SingleFrameLine{\z@}%
36 }
37
38 \def\FV@EndListFrame@Single{%
39   \FV@SingleFrameLine{\@ne}%
40 }
41
42 \def\FV@BeginListFrame@Lines{%
43   \FV@SingleFrameLine{\z@}%
44 }
45
46 \def\FV@EndListFrame@Lines{%
47   \FV@SingleFrameLine{\@ne}%
48 }
49
50 \renewcommand*\FV@SingleFrameSep{}
```

Adds HTML formatting:

```
51 \def\FV@BUseVerbatim#1{%
52   \LWR@atbeginverbatim[\LWR@FVstyle]{0}{verbatim}%
53   \FV@BVerbatimBegin#1\FV@BVerbatimEnd%
54   \LWR@afterendverbatim{0}%
55 }
```

`\LWR@FVstyle` Holds the style of the verbatim.

```
56 \newcommand*\LWR@FVstyle{}
```

The following patches to `Verbatim` are executed at the start and end of the environment, depending on the choice of `frame`. Original code is from the `fancyvrb` package.

```

57 \newcommand*{\LWR@fvstartnone}{%
58 \LWR@traceinfo{fvstartnone}%
59 % \hbox to\z@{%
60 \LWR@atbeginverbatim[\LWR@FVstyle]{0}{verbatim}%
61 }%
62 }
63
64 \newcommand*{\LWR@f vendnone}{%
65 \LWR@traceinfo{f vendnone}%
66 % \hbox to\z@{%
67 \LWR@afterendverbatim{0}%
68 }%
69 }
70
71 \newcommand*{\LWR@fvstartsingl e}{%
72 \LWR@traceinfo{fvstartsingl e}%
73 \LWR@fvstartnone%
74 \FV@BeginListFrame@Single%
75 }
76
77 \newcommand*{\LWR@fvendsingl e}{%
78 \LWR@traceinfo{fvendsingl e}%
79 \FV@EndListFrame@Single%
80 \LWR@f vendnone%
81 }
82
83 \newcommand*{\LWR@fvstartline}{%
84 \LWR@traceinfo{fvstartline}%
85 \LWR@fvstartnone%
86 % \setlength{\LWR@templengthone}{\baselineskip}%
87 \FV@BeginListFrame@Lines%
88 % \setlength{\baselineskip}{\LWR@templengthone}%
89 % \setlength{\baselineskip}{5pt}%
90 }
91
92 \newcommand*{\LWR@f vendline}{%
93 \LWR@traceinfo{f vendline}%
94 \FV@EndListFrame@Lines%
95 \LWR@f vendnone%
96 }
```

The following patches select the start/left/right/end behaviors depending on `frame`. Original code is from the `fancyvrb` package.

```
97 \newcommand*{\LWR@FVfindbordercolor}{%
```

```
98 \FancyVerbRuleColor%
99 \LWR@findcurrenttextcolor%
100 \color{black}%
101 }
102
103 % border width of \FV@FrameRule
104 \newcommand*{\LWR@FVborderstyle}[1]{%
105 padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ; %
106 \LWR@FVfindbordercolor %
107 border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt %
108 solid \LWR@origpound\LWR@tempcolor ; %
109 }
110
111 \def\FV@Frame@none{%
112 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle}%
113 \let\FV@BeginListFrame\LWR@fvstartnone%
114 \let\FV@LeftListFrame\relax%
115 \let\FV@RightListFrame\relax%
116 \let\FV@EndListFrame\LWR@fvendnone}
117
118 \FV@Frame@none% default values
119
120 \def\FV@Frame@singles{%
121 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{} }%
122 \let\FV@BeginListFrame\LWR@fvstartsingle%
123 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
124 \let\FV@RightListFrame\FV@RightListFrame@Single%
125 \let\FV@EndListFrame\LWR@fvendsingle}
126
127 \def\FV@Frame@lines{%
128 \renewcommand*{\LWR@FVstyle}{%
129     \LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}\LWR@FVborderstyle{-bottom}%
130 }%
131 \let\FV@BeginListFrame\LWR@fvstartline%
132 \let\FV@LeftListFrame\relax%
133 \let\FV@RightListFrame\relax%
134 \let\FV@EndListFrame\LWR@fvendline}
135
136 \def\FV@Frame@topline{%
137 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}}%
138 \let\FV@BeginListFrame\LWR@fvstartline%
139 \let\FV@LeftListFrame\relax%
140 \let\FV@RightListFrame\relax%
141 \let\FV@EndListFrame\LWR@fvendnone}
142
143 \def\FV@Frame@bottomline{%
144 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-bottom}}%
145 \let\FV@BeginListFrame\LWR@fvstartnone%
146 \let\FV@LeftListFrame\relax%
147 \let\FV@RightListFrame\relax%
```

```

148 \let\FV@EndListFrame\LWR@fvendline}
149
150 \def\FV@Frame@leftline{%
151 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-left}}}%
152 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
153 \ifx\FancyVerbFillColor\relax%
154 \let\FV@FrameFillLine\relax%
155 \else%
156 \tempdima\FV@FrameRule\relax%
157 \multiply\tempdima-\tw@%
158 \edef\FV@FrameFillLine{%
159 {\noexpand\FancyVerbFillColor{\vrule\@width\number\tempdima sp}%
160 \kern-\number\tempdima sp}}%
161 \fi%
162 \let\FV@BeginListFrame\LWR@fvstartnone%
163 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
164 \let\FV@RightListFrame\relax%
165 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the fancyvrb package.

```

166 \def\FV@SingleFrameLine#1{%
167 %    \hbox to\z@{%
168 %        \kern\leftmargin
169 %        \ifnum#1=\z@\relax
170 %            \let\FV@Label\FV@LabelBegin
171 %        \else
172 %            \let\FV@Label\FV@LabelEnd
173 %        \fi
174 %        \ifx\FV@Label\relax
175 %            \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
176 %        \else
177 %            \ifnum#1=\z@
178 %                \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
179 %                \ifx\FV@LabelPositionTopLine\relax
180 %                \else
181 %                    \LWR@FVfindbordercolor
182 %                    \LWR@htmltagc{%
183 %                        div class="fancyvrblabel" % extra space
184 %                        style="color: \LWR@origpound\LWR@tempcolor"%
185 %                    }
186 %                    \LWR@print@textrm{\FV@LabelBegin}%
187 %                    \LWR@htmltagc{/div}
188 %                \fi
189 %            \else
190 %                \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
191 %                \ifx\FV@LabelPositionBottomLine\relax
192 %                \else

```

```

193      \LWR@FVfindbordercolor
194
195      \LWR@htmltagc{%
196          div class="fancyvrblabel" % extra space
197          style="color: \LWR@origpound\LWR@tempcolor"%
198      }
199      \LWR@print@textrm{\FV@LabelEnd}
200      \LWR@htmltagc{/div}
201      \fi
202      \fi
203      \fi
204 %
205 %
206 }
```

Processes each line, adding optional line numbers. Original code is from the `fancyvrb` package.

```

207 \def\FV@ListProcessLine#1{%
208     \hbox to \hsize{%
209 %         \kern\leftmargin
210     \hbox to \VerbatimHTMLWidth {%
211         \ifcsvvoid{FV@LeftListNumber}{}{\kern 2.5em}%
212         \FV@LeftListNumber%
213 %
214         \FV@LeftListFrame
215         \FancyVerbFormatLine{#1}%
216 %
217         \FV@RightListFrame
218         \FV@RightListNumber%
219     }%
220     \hss% required to avoid underfull hboxes
221 }
```

Env `BVerbatim`

```

222 \AtBeginEnvironment{BVerbatim}
223 {%
224 \LWR@forcenewpage% instead of \preto
225 \LWR@atbeginverbatim{0}{bverbatim}%
226 }
227
228 \AfterEndEnvironment{BVerbatim}
229 {%
230 \LWR@afterendverbatim{0}%
231 }
```

End of the modifications to make at the end of the preamble:

```
232 } % \AfterEndPreamble
```

File 110 **lwarp-figcaps.sty**

§ 206 Package **figcaps**

(Emulates or patches code by PATRICK W. DALY.)

Pkg **figcaps** Emulated.

for HTML output: Discard all options for lwarp-figcaps:

```
1 \LWR@ProvidesPackageDrop{figcaps}

2 \newcommand*{\figcapson}={}
3 \newcommand*{\figcapsoff}={}
4 \newcommand*{\printfigures}={}
5 \newcommand*{\figmarkon}={}
6 \newcommand*{\figmarkoff}={}
7 \def\figurecapname{Figure Captions}
8 \def\tablepagename{Tables}
9 \def\figurepagename{Figures}
```

File 111 **lwarp-figsize.sty**

§ 207 Package **figsize**

(Emulates or patches code by ANTHONY A. TANBAKUCHI.)

Pkg **figsize** **figsize** is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{figsize}
```

Emulates a virtual 6×9 inch textsize.

```
2 \newlength{\figwidth}
3 \newlength{\figheight}
4
5 \newcommand{\SetFigLayout}[3][0]{%
6 \setlength{\figheight}{8in}%
7 \setlength{\figheight}{\figheight / #2}%
8 %
```

```

9 \setlength{\figwidth}{5.5in}%
10 \setlength{\figwidth}{\figwidth / #3}%
11 }

```

File 112 **l warp-fix2col.sty**

§ 208 Package **fix2col**

Pkg fix2col fix2col is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{fix2col}

File 113 **l warp-fixme.sty**

§ 209 Package **fixme**

(Emulates or patches code by DIDIER VERNA.)

Pkg fixme fixme is patched for use by l warp.

⚠ **external layouts** External layouts (\fxloadlayouts) are not supported.

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

```

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

```

for HTML output: 1 \LWR@ProvidesPackagePass{fixme}

Restore l warp’s version of \@wrindex, ignoring the fixme package’s target option:

```
2 \let\@wrindex\LWR@wrindex
```

Float-related macros required by l warp:

```

3 \newcommand{\ext@fixme}{lox}
4
5 \renewcommand{\l@fixme}[2]{\hypertocfloat{1}{fixme}{lox}{#1}{#2}}

```

Other modifications:

```

6 \def\FXFaceInlineHTMLStyle{font-weight:bold}
7
8 \renewcommand*\FXLayoutInline[3]{ %
9   \InlineClass[\FXFaceInlineHTMLStyle]{fixmeinline}%
10   {\@fxtextstd{\#1}{\#2}{\#3}}%
11 }
12
13 \def\FXFaceEnvHTMLStyle{font-weight:bold}
14
15 \renewcommand*\FXEnvLayoutPlainBegin[2]{%
16   \BlockClass[\FXFaceEnvHTMLStyle]{fixmebold}%
17   \ignorespaces#2 \fxnotename{\#1}: \ignorespaces}%
18
19 \renewcommand*\FXEnvLayoutPlainEnd[2]{\endBlockClass}%
20
21 \renewcommand*\FXEnvLayoutSignatureBegin[2]{%
22   \BlockClass[\FXFaceEnvHTMLStyle]{fixmebold}%
23   \fxnotename{\#1}: \ignorespaces}%
24
25 \renewcommand*\FXEnvLayoutSignatureEnd[2]{\@fxsignature{\#2}\endBlockClass}%
26
27 \def\FXFaceSignatureHTMLStyle{font-style:italic}
28
29 \DeclareRobustCommand*\@fxsignature[1]{%
30   \ifthenelse{\equal{\#1}{}}{%
31     {}%
32     { -- \InlineClass[\FXFaceSignatureHTMLStyle]{fixmesignature}{\#1}}%
33   }%
34
35
36 \def\FXFaceTargetHTMLStyle{font-style:italic}
37
38 \renewcommand\FXTargetLayoutPlain[2]{%
39   \InlineClass[\FXFaceTargetHTMLStyle]{fixmetarget}{\#2}}%
40 }

```

File 114 **lwarf-fixmetodonotes.sty**

§ 210 Package **fixmetodonotes**

(Emulates or patches code by GIOELE BARABUCCI.)

Pkg **fixmetodonotes** **fixmetodonotes** is patched for use by **lwarf**.

for HTML output:

```

1 \LWR@ProvidesPackagePass{fixmetodonotes}

2 \renewcommand{\NOTES@addtolist}[2]{%

```

```

3      \refstepcounter{NOTES@note}%
4 %    \phantomsection% REMOVED
5      \addcontentsline{notes}{NOTES@note}{%
6          \protect\numberline{\theNOTES@note}{\#1}: {\#2}}%
7      }%
8 }
9
10 \renewcommand{\NOTES@marker}[2]{\fbox{%
11     \textcolor{#2}{% WAS \color
12         \textbf{\#1}}%
13 }}
```

14

15 \renewcommand{\NOTES@colorline}[2]{%
16 \bgroup%
17 \ULon{\LWR@backgroundcolor{\#1}{\#2}}%
18 }

File 115 **lwarf-flafter.sty**

§ 211 Package **flafter**

Pkg flafter flafter is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{flafter}
2 \providecommand\f@trace[1]{}
```

File 116 **lwarf-flippdf.sty**

§ 212 Package **flippdf**

Pkg flippdf flippdf is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{flippdf}[2006/06/30]
2 \newcommand\FlipPDF{}
3 \newcommand\UnFlipPDF{}
```

File 117 **lwarf-float.sty**

§ 213 Package **float**

(Emulates or patches code by ANSELM LINGNAU.)

Pkg float float is emulated.

Float styles boxed and ruled are emulated by css and a float class according to style.

The HTML <figure> class is set to the float type, so css may also be used to format the float and its caption, according to float type. Furthermore, an additional class is set to the float style: plain, plaintop, boxed, or ruled, so css may be used to format by float style as well. Default formatting by css is provided for ruled and boxed styles.

for HTML output: 1 \LWR@ProvidesPackageDrop{float}[2001/11/08]

\listof See section 74.2 for the \listof command.

\LWR@floatstyle The default float style:

2 \newcommand*\{\LWR@floatstyle\}{plain}

\newfloat {\langle 1: type\rangle} {\langle 2: placement\rangle} {\langle 3: ext\rangle} [{\langle 4: within\rangle}]

Emulates the \newfloat command from the float package.

“placement” is ignored.

```
3 \NewDocumentCommand{\newfloat}{m m m o}{%
4 \IfValueTF{#4}{%
5 {\\DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}}{%
6 {\\DeclareFloatingEnvironment[fileext=#3]{#1}}{}}
```

Remember the float style:

7 \csedef{\LWR@floatstyle@#1}{\LWR@floatstyle}%

newfloat package automatically creates the \listof command for new floats, but float does not, so remove \listof here in case it is manually created later.

```
8 \cslet{\listof#1s}\relax%
9 \cslet{\listof#1es}\relax%
```

Likesize, newfloat also creates \l@<type>, but float does not, so remove it here:

```
10 \cslet{\l@#1}\relax%
11 }
```

\floatname {\langle type\rangle} {\langle name\rangle}

Sets the text name of the float, such as “Figure”.

```
12 \NewDocumentCommand{\floatname}{m +m}{%
13     \SetupFloatingEnvironment{#1}{name=#2}}%
14 }
```

```
\floatplacement  {⟨type⟩} {⟨placement⟩}
    Float placement is ignored.

15 \newcommand*{\floatplacement}[2]{%
16     \SetupFloatingEnvironment{#1}{placement=#2}%
17 }

\floatstyle   {⟨style⟩}
    Remember the style for future floats:

18 \newcommand{\floatstyle}[1]{%
19     \def\LWR@floatstyle{#1}%
20 }

\restylefloat * {⟨type⟩}
    Remember the style for this float:

21 \NewDocumentCommand{\restylefloat}{s m}{%
22     \csedef{\LWR@floatstyle@#2}{\LWR@floatstyle}%
23 }
```

File 118 **lwarf-floatflt.sty**

§ 214 Package **floatflt**

(Emulates or patches code by MATS DAHLGREN.)

Pkg **floatflt** Emulated.

for HTML output: Discard all options for lwarf-floatflt:

```
1 \LWR@ProvidesPackageDrop{floatflt}
```

Env [⟨⟩] **offset** {⟨type⟩} {⟨width⟩} Borrowed from the lwarf version of keyfloat:

```
2 \NewDocumentEnvironment{KFLTfloatflt@marginfloat}{O{-1.2ex} m m}%
3 {%
4 \setlength{\LWR@templengthone}{#3}%
5 \LWR@BlockClassWP{%
6     float:right; %
7     width:\LWR@printlength{\LWR@templengthone}; % extra space
8     margin:10pt%
9 }{%
10    width:\LWR@printlength{\LWR@templengthone}%
11 }}
```

```

12 {marginblock}%
13 \captionsetup{type=#2}%
14 }
15 {%
16 \endLWR@BlockClassWP%
17 }

Env floatingfigure [placement] [width]
18 \DeclareDocumentEnvironment{floatingfigure}{o m}
19  {\begin{KFLTfloatfltnomarginfloat}{figure}{#2}}
20  {\end{KFLTfloatfltnomarginfloat}}
```



```

Env floatingtable [placement]
21 \DeclareDocumentEnvironment{floatingtable}{o}
22  {\begin{KFLTfloatfltnomarginfloat}{table}{1.5in}}
23  {\end{KFLTfloatfltnomarginfloat}}
```

File 119 **l warp-floatpag.sty**§ 215 Package **floatpag***(Emulates or patches code by VYTAS STATULEVIČIUS AND SIGITAS TOLUŠIS.)*

Pkg floatpag Emulated.

for HTML output: Discard all options for l warp-floatpag:

```

1 \LWR@ProvidesPackageDrop{floatpag}

2 \newcommand*\{\floatpagestyle}[1]{}
3 \newcommand*\{\rotfloatpagestyle}[1]{}
4 \newcommand*\{\thisfloatpagestyle}[1]{}
```

File 120 **l warp-floatrow.sty**§ 216 Package **floatrow***(Emulates or patches code by OLGA LAPKO.)*

Pkg floatrow floatrow is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{floatrow}

- ⚠ Misplaced alignment tab character & Use `\StartDefiningTabulars` and `\StopDefiningTabulars` before and after defining macros using `\ttabbox` with a tabular inside. See section 9.9.
- ⚠ subfig package When combined with the `subfig` package, while inside a `subfloatrow` `\ffigbox` and `\ttabbox` must have the caption in the first of the two of the mandatory arguments.
- ⚠ `\FBwidth`, `\FBheight` The emulation of `floatrow` does not support `\FBwidth` or `\FBheight`. These values are pre-set to `.3\linewidth` and `2in`. Possible solutions include:
- Use fixed lengths. `l warp` will scale the `HTML` lengths appropriately.
 - Use `warpprint` and `warpHTML` environments to select appropriate values for each case.
 - Inside a `warpHTML` environment, manually change `\FBwidth` or `\FBheight` before the `\ffigbox` or `\ttabbox`. Use `\FBwidth` or `\FBheight` normally afterwards; it will be used as expected in print output, and will use your custom-selected value in `HTML` output. This custom value will be used repeatedly, until it is manually changed to a new value.

After everything has loaded, remember whether `subcaption` was loaded. If not, it is assumed that `subfig` is used instead:

```

2 \newbool{LWR@subcaptionloaded}
3
4 \AtBeginDocument{
5 \@ifpackageloaded{subcaption}
6 {\booltrue{LWR@subcaptionloaded}}
7 {\boolfalse{LWR@subcaptionloaded}}
8 }

\fbox [⟨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{\fbox}{o m o o +m +m}{%
10 \ifbool{LWR@subcaptionloaded}{%
11 {%

```

For `subcaption`:

```

12 \ifbool{LWR@insubfloatrow}{%
13 {%

```

`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*{\caption}{}
48 \newlength{\FBwidth}
49 \setlength{\FBwidth}{.3\linewidth}
50 \newlength{\FBheight}
51 \setlength{\FBheight}{2in}
52 \newcommand*{\useFCwidth}{}
53 \newcommand{\floatsetup}[2][]{}
54 \newcommand{\thisfloatsetup}[1]{}
55 \newcommand{\clearfloatsetup}[1]{}
56 \newcommand*{\killfloatstyle}{}

\newfloatcommand {⟨1 command⟩} {⟨2 captype⟩} [⟨3 preamble⟩] [⟨4 default width⟩]
Preamble and default width are ignored.

57 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
58 @namedef{#1}{%
59 \floatbox{#2}{%
60 }%
61 }}

\renewfloatcommand {⟨1 command⟩} {⟨2 captype⟩} [⟨3 preamble⟩] [⟨4 default width⟩]
Preamble and default width are ignored.

62 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
63 @namedef{#1}{%
64 \floatbox{#2}{%
65 }%
66 }}

\ffigbox [⟨width⟩] [⟨height⟩] [⟨vposn⟩] {⟨caption commands⟩} {⟨contents⟩}
67 \newfloatcommand{ffigbox}{figure}[\nocapbeside] []

\ttabbox [⟨width⟩] [⟨height⟩] [⟨vposn⟩] {⟨caption commands⟩} {⟨contents⟩}
68 \newfloatcommand{ttabbox}{table}[\caption][\FBwidth]

\fcapside [⟨width⟩] [⟨height⟩] [⟨vposn⟩] {⟨caption commands⟩} {⟨contents⟩}
69 \newfloatcommand{fcapside}{figure}[\capbeside] []

Env   floatrow [⟨numfloats⟩]
The row of floats is placed into a <div> of class floatrow.

70 \newenvironment*{floatrow}[1][2]

```

```

71 {
72 \LWR@forcenewpage
73 \BlockClass{floatrow}

```

While inside the floatrow, divide the \linewidth by the number of floats.

```

74 \booltrue{\LWR@infloatrow}
75 \setlength{\linewidth}{6in/#1}
76 }
77 {
78 \boolfalse{\LWR@infloatrow}
79 \endBlockClass
80 }

```

Keys for \DeclareNewFloatType:

```

81 \newcommand*{\LWR@frowkeyplacement}={}
82 \newcommand*{\LWR@frowkeyname}={}
83 \newcommand*{\LWR@frowkeyfileext}={}
84 \newcommand*{\LWR@frowkeywithin}={}
85 \newcommand*{\LWR@frowkeycapstyle}={}
86
87 \define@key{frowkeys}{placement}{}
88 \define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}
89 \define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}
90 \define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}
91 \define@key{frowkeys}{relatedcapstyle}{}

```

\DeclareNewFloatType {<type>} {<options>}

Use \listof{type}{Title} to print a list of the floats.

```
92 \newcommand*{\DeclareNewFloatType}[2]{%
```

Reset key values:

```

93 \renewcommand*{\LWR@frowkeyplacement}{}
94 \renewcommand*{\LWR@frowkeyname}{}
95 \renewcommand*{\LWR@frowkeyfileext}{}
96 \renewcommand*{\LWR@frowkeywithin}{}
97 \renewcommand*{\LWR@frowkeycapstyle}{}

```

Read new key values:

```

98 \LWR@traceinfo{about to setkeys frowkeys}%
99 \setkeys{frowkeys}{#2}%
100 \LWR@traceinfo{finished setkeys frowkeys}%

```

Create a new float with optional [within]:

```

101 \ifthenelse{\equal{\LWR@frowkeywithin}{}}{%
102 }%
103   \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\ %

```

```

104      \LWR@frowkeyfileext}%
105      \newfloat{\#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}%
106 }%
107 {%
108   \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\ %
109     \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
110   \newfloat{\#1}{\LWR@frowkeyplacement}%
111   {\LWR@frowkeyfileext}[\LWR@frowkeywithin]%
112   \LWR@traceinfo{finished newfloat #1}%
113 }%

```

Rename the float if a name was given:

```

114 \ifthenelse{\equal{\LWR@frowkeyname}{}}{%
115 }%
116 {\floatname{\#1}{\LWR@frowkeyname}}%
117 }

```

Not used:

```

118 \newcommand{\buildFBBOX}[2]{}
119 \newcommand*{\CenterFloatBoxes}{}%
120 \newcommand*{\TopFloatBoxes}{}%
121 \newcommand*{\BottomFloatBoxes}{}%
122 \newcommand*{\PlainFloatBoxes}{}%
123
124 \newcommand{\capsubrowsettings}{}%
125
126 \NewDocumentCommand{\RawFloats}{o o}{}%

```

\RawCaption {*text*}

To be used inside a minipage or parbox.

```
127 \newcommand{\RawCaption}[1]{#1}
```

\floatfoot {*text*}

Places additional text inside a float, inside a css <div> of class floatfoot.

```

128 \NewDocumentCommand{\floatfoot}{s +m}{%
129 \begin{BlockClass}{floatfoot}%
130 #2%
131 \end{BlockClass}%
132 }

```

Used to compute \linewidth.

```

133 \newbool{\LWR@insubfloatrow}%
134 \boolfalse{\LWR@insubfloatrow}%

```

```
Env  subfloatrow  [ $\langle num\_floats \rangle$ ]
135 \newenvironment*{subfloatrow}[1][2]
136 {
```

The row of floats is placed into a <div> of class floatrow:

```
137 \LWR@forcenewpage
138 \BlockClass{floatrow}
```

While inside the floatrow, LWR@insubfloatrow is set true, which tells \floatbox to use \subfigure or \subtable.

```
139 \begingroup
140 \booltrue{LWR@insubfloatrow}
141 }
142 {
143 \endgroup
144 \endBlockClass
145 \boolfalse{LWR@insubfloatrow}
146 }
```

File 121 l warp-fltrace.sty

§ 217 Package **fltrace**

Pkg fltrace fltrace is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{fltrace}

2 \def\tracefloats{}
3 \def\tracefloatsoff{}
4 \def\tracefloatvals{}
```

File 122 l warp-flushend.sty

§ 218 Package **flushend**

(Emulates or patches code by SIGITAS TOLUŠIS.)

Pkg flushend Emulated.

for HTML output: Discard all options for l warp-flushend:

```
1 \LWR@ProvidesPackageDrop{flushend}
```

```

2 \newcommand*{\flushend}{}
3 \newcommand*{\raggedend}{}
4 \newcommand*{\flushcolsend}{}
5 \newcommand*{\raggedcolsend}{}
6 \newcommand*{\atColsBreak}[1]{}
7 \newcommand*{\atColsEnd}[1]{}
8 \newcommand*{\showcolsendlrule}{}

```

File 123 **lwarf-fnbreak.sty**

§ 219 Package **fnbreak**

Pkg **fnbreak** **fnbreak** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{fnbreak}

2 \newcommand*{\fnbreakverbose}{}
3 \newcommand*{\fnbreaknonverbose}{}
4 \newcommand*{\fnbreaklabel}{}
5 \newcommand*{\fnbreaknolabel}{}

```

File 124 **lwarf-fncychap.sty**

§ 220 Package **fncychap**

(Emulates or patches code by ULF A. LINDGREN.)

Pkg **fncychap** **fncychap** is emulated.

for HTML output: Discard all options for lwarf-fncychap:

```

1 \LWR@ProvidesPackageDrop{fncychap}

2 \def\mghrulefill#1{}
3 \def\ChNameLowerCase{}
4 \def\ChNameUpperCase{}
5 \def\ChNameAsIs{}
6 \def\ChTitleLowerCase{}
7 \def\ChTitleUpperCase{}
8 \def\ChTitleAsIs{}
9 \newcommand{\ChRuleWidth}[1]{}
10 \newcommand{\ChNameVar}[1]{}
11 \newcommand{\ChNumVar}[1]{}
12 \newcommand{\ChTitleVar}[1]{}

```

```

13 \newcommand{\TheAlphaChapter}={}
14 \newcommand{\DOCH}{}
15 \newcommand{\DOTI}[1]{}
16 \newcommand{\DOTIS}[1]{}
17 \newlength{\mylen}
18 \newlength{\myhi}
19 \newlength{\px}
20 \newlength{\py}
21 \newlength{\pyy}
22 \newlength{\pxx}
23 \newlength{\RW}
24 \newcommand{\FmN}[1]{#1}
25 \newcommand{\FmTi}[1]{#1}

```

File 125 **l warp-fnlineno.sty**

§ 221 Package **fnlineno**

Pkg **fnlineno** **fnlineno** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{fnlineno}

File 126 **l warp-fnpos.sty**

§ 222 Package **fnpos**

(Emulates or patches code by HIROSHI NAKASHIMA.)

Pkg **fnpos** **fnpos** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{fnpos}

 2 \newcommand*{\makeFNbottom}={}
 3 \newcommand*{\makeFNmid}={}
 4 \newcommand*{\makeFNbelow}={}
 5 \newcommand*{\makeFNabove}={}

File 127 **l warp-fontenc.sty**

§ 223 Package **fontenc**

Pkg **fontenc** If using pdfLATEX, l warp used to require fontenc be loaded before l warp, but now

l warp itself loads \fontenc with T1 encoding, which l warp requires. fontenc is now allowed to be loaded with another encoding after l warp.

l warp-fontenc is no longer necessary, but is still provided to overwrite older versions.

for HTML output: 1 \LWR@ProvidesPackagePass{fontenc}

File 128 **l warp-footmisc.sty**

§ 224 Package **footmisc**

(Emulates or patches code by ROBIN FAIRBAIRNS.)

Pkg footmisc footmisc is emulated.

l warp incidentally happens to emulate the stable option.

1 \LWR@ProvidesPackageDrop{footmisc}

Some nullified commands:

```

2 \newcommand{\footnotelayout}{}%
3 \newcommand{\setfnsymbol}[1]{}%
4 \NewDocumentCommand{\DefineFNsymbols}{s m o m}{}%
5
6 \newdimen\footnotemargin
7 \footnotemargin1.8em\relax
8
9 \newcommand*\hangfootparskip{0.5\baselineskip}
10 \newcommand*\hangfootparindent{0em}%
11
12 \let\pagefootnoterule\footnoterule
13 \let\mpfootnoterule\footnoterule
14 \def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}%
15
16 \providecommand*\multiplefootnotemarker{3sp}%
17 \providecommand*\multfootsep{,}

```

Using cleveref:

18 \providecommand*\footref[1]{\labelcref{#1}}

The following work as-is:

```

19 \newcommand\mpfootnotemark{%
20   \@ifnextchar[%]

```

```

21      \@xmpfootnotemark%
22      {%
23          \stepcounter\@mpfn%
24          \protected@xdef\@thefnmark{\thempfn}%
25          \@footnotemark%
26      }%
27 }
28 \def\@xmpfootnotemark[#1]{%
29     \begingroup%
30     \csname c@\@mpfn\endcsname #1\relax%
31     \unrestored@protected@xdef\@thefnmark{\thempfn}%
32     \endgroup%
33     \@footnotemark%
34 }

```

File 129 **l warp-footnote.sty**

§ 225 Package **footnote**

(Emulates or patches code by MARK WOODING.)

Pkg **footnote** footnote is used with minor patches.

for HTML output: 1 \LWR@ProvidesPackagePass{footnote}

Removed print-version formatting:

```

2 \def\fn@startnote{%
3 %   \parboxrestore%
4   \protected@edef\@currentlabel{\csname p@\@mpfn\endcsname\@thefnmark}%
5 %   \color@begingroup% *** conflicts with l warp
6 }
7
8 % \let\fn@endnote\color@endgroup% *** conflicts with l warp
9 \def\fn@endnote{%
10 \LWR@htmlltagc{/ \LWR@tagregularparagraph}%
11 \LWR@orignewline%
12 }

```

Removed print-version formatting:

```

13 \def\fn@startfntext{%
14   \setbox\z@\vbox\bgroup%
15   \fn@startnote%
16   \fn@prefntext%
17   \ignorespaces%
18 }

```

Removed print-version formatting, added closing paragraph tag:

```

19 \def\fn@endfntext{%
20   \LWR@htmltagc{/\\LWR@tagregularparagraph}%
21   \LWR@newline%
22   \fn@postfntext%
23 \egroup%
24 \begingroup%
25   \let\@makefntext\@empty%
26   \let\@finalstrut\@gobble%
27   \LetLtxMacro\rule\@gobbletwo% *8* also the optional argument?
28   \footnotetext{\unvbox\z@}%
29 \endgroup%
30 }
```

These have been redefined, so re-\let them again:

```

31 \let\endfootnote\fn@endfntext
32 \let\footnotetext\endfootnote
```

File 130 **l warp-footnotehyper.sty**

§ 226 Package **footnotehyper**

Pkg **footnotehyper** footnotehyper is a hyperref-safe version of footnote. For l warp, footnotehyper is emulated.

for HTML output: Discard all options for l warp-footnotehyper:

```

1 \RequirePackage{footnote}
2 \LWR@ProvidesPackageDrop{footnotehyper}
```

File 131 **l warp-footnoterange.sty**

§ 227 Package **footnoterange**

(Emulates or patches code by H.-MARTIN MÜNCH.)

Pkg **footnoterange** footnoterange is patched for use by l warp.

for HTML output:

```

1 \LWR@ProvidesPackagePass{footnoterange}

2 \csletcs{footnoterange}{footnoterange*}
3 \csletcs{endfootnoterange}{endfootnoterange*}
```

File 132 **l warp-footnpag.sty**

§ 228 Package **footnpag**

Pkg **footnpag** **footnpag** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{footnpag}

File 133 **l warp-forest.sty**

§ 229 Package **forest**

(Emulates or patches code by SAŠO ŽIVANOVIĆ.)

Pkg **forest** **forest** is patched for use by **l warp**.

 **\Forest*** The starred version of the macro **\Forest*** is not supported. **l warp** encases each **lateximage** in an environment, so the global results of the starred **\Forest*** are lost.

for HTML output: 1 \LWR@ProvidesPackagePass{forest}

```
2 \BeforeBeginEnvironment{forest}{\begin{lateximage}[forest]}
```

3

```
4 \AfterEndEnvironment{forest}{\end{lateximage}}
```

5

```
6 \RenewDocumentCommand{\Forest}{s D(){} m}{%
```

7 \forest@config{#2} %

8 \IfBooleanTF{#1}{%

9 \PackageError{l warp-forest} %

10 {Starred \Forest is not supported} %

11 {L warp uses an environment for images, but \Forest* cannot work in an environment.} %

12 \let\forest@next\forest@env %

13 }{\let\forest@next\forest@group@env} %

14 \begin{lateximage}[forest] % l warp

15 \forest@next{#3} %

16 \end{lateximage} % l warp

17 }

File 134 **l warp-framed.sty**

§ 230 Package **framed**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg **framed** **framed** is supported and patched by l warp.

for HTML output: Accept all options for l warp-framed:

```
1 \LWR@ProvidesPackagePass{framed}
2 \RequirePackage{xcolor}%
3
4 \renewenvironment{framed}{%
5 \LWR@forcenewpage
6 \BlockClass{framed}%
7 }
8 {\endBlockClass}
9
10 \renewenvironment{oframed}{%
11 \LWR@forcenewpage
12 \BlockClass{framed}%
13 }
14 {\endBlockClass}
15
16
17 \renewenvironment{shaded}{%
18 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
19 \LWR@forcenewpage
20 \BlockClass[background: \LWR@origpound\LWR@tempcolor]{shaded}%
21 }
22 {\endBlockClass}
23
24 \renewenvironment{shaded*}{%
25 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
26 \LWR@forcenewpage
27 \BlockClass[background: \LWR@origpound\LWR@tempcolor]{shaded}%
28 }
29 {\endBlockClass}
30
31
32 \renewenvironment{leftbar}{%
33 \LWR@forcenewpage
34     \BlockClass{framedleftbar}
35     \def\FrameCommand{}%
```

```
36 \MakeFramed {}
37 }%
38 {\endMakeFramed\endBlockClass}
39
40
41 \renewenvironment{snugshade}{%
42 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
43 \LWR@forcenewpage
44 \BlockClass[background: \LWR@origpound\LWR@tempcolor]{snugframed}%
45 }
46 {\endBlockClass}
47
48 \renewenvironment{snugshade*}{%
49 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
50 \LWR@forcenewpage
51 \BlockClass[background: \LWR@origpound\LWR@tempcolor]{snugframed}%
52 }
53 {\endBlockClass}
54
55 \let\oframed\framed
56 \let\endoframed\endframed
57
58
59 \RenewEnviron{titled-frame}[1]{%
60 \CustomFBox{#1}{\Opt}{\Opt}{\Opt}{\Opt}{\BODY}}
61 }

\CustomFBox {\langle toptitle\rangle} {\langle bottitle\rangle} {\langle thicknessstop\rangle} {\langle bottom\rangle} {\langle left\rangle} {\langle right\rangle}
{\langle text contents\rangle}

62 \renewcommand{\CustomFBox}[7]{%
63 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
64 \LWR@forcenewpage
65 \begin{BlockClass}[border: 3px solid \LWR@origpound\LWR@tempcolor]{framed}%
66 \ifthenelse{\isempty{#1}}{}{%
67   \begin{BlockClass}[background: \LWR@origpound\LWR@tempcolor]{framedtitle}%
68     \textcolor{TFTTitleColor}{\textbf{#1}}%
69   \end{BlockClass}%
70 }% not empty
71
72 #7
73
74 \ifthenelse{\isempty{#2}}{}{%
75   \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
76   \begin{BlockClass}[background: \LWR@origpound\LWR@tempcolor]{framedtitle}%
77     \textcolor{TFTTitleColor}{\textbf{#2}}%
78   \end{BlockClass}%
79 }% not empty
80 \end{BlockClass}
```

```

81 }

\TitleBarFrame [⟨marker⟩] {⟨title⟩} {⟨contents⟩}

82 \renewcommand\TitleBarFrame[3] []{
83 \CustomFBox
84   {#2}{%}
85   \fboxrule\fboxrule\fboxrule\fboxrule
86   {#3}%
87 }

88 \renewcommand{\TF@Title}[1]{#1}

MakeFramed {⟨settings⟩}

89 \let\MakeFramed\relax
90 \let\endMakeFramed\relax
91
92 \NewEnviron{MakeFramed}[1]{%
93 \FrameCommand{\begin{minipage}{\linewidth}\BODY\end{minipage}}%
94 }

\fb@put@frame {⟨frame cmd no split⟩} {⟨frame cmd split⟩}

95 \renewcommand*{\fb@put@frame}[2]{%
96 \relax%
97 \@tempboxa%
98 }

```

File 135 **lwarf-ftnright.sty**

§ 231 Package **ftnright**

Pkg **ftnright** **ftnright** is ignored.

for HTML output: Discard all options for **lwarf-ftnright**:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

File 136 **lwarf-fullminipage.sty**

§ 232 Package **fullminipage**

Pkg **fullminipage** **fullminipage** is nullified.

for HTML output: 1 \LWR@ProvidesPackageDrop{fullminipage}[2014/07/06]
2 \newenvironment{fullminipage}[1] [] {}{}

File 137 **l warp-fullpage.sty**

§ 233 Package **fullpage**

Pkg **fullpage** **fullpage** is ignored.

for HTML output: Discard all options for l warp-fullpage:

1 \LWR@ProvidesPackageDrop{fullpage}[1994/06/01]

File 138 **l warp-fullwidth.sty**

§ 234 Package **fullwidth**

(Emulates or patches code by MARCO DANIEL.)

Pkg **fullwidth** **fullwidth** is emulated.

A minipage is used, of no HTML width.

for HTML output: 1 \LWR@ProvidesPackageDrop{fullwidth}
2 \newenvironment*{fullwidth}[1] [] {}%
3 \minipage{fullwidth}%
4 \minipage{\linewidth}%
5 }%
6 {}%
7 \endminipage%
8 }

File 139 **l warp-fwlw.sty**

§ 235 Package **f wlw**

Pkg **f wlw** **f wlw** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{f wlw}

```
2 \newbox\FirstWordBox      \global\setbox\FirstWordBox\hbox{ }
3 \newbox\NextWordBox       \global\setbox\NextWordBox\hbox{ }
4 \newbox\LastWordBox        \global\setbox\LastWordBox\hbox{ }
5 \def\ps@fwlhead{}
6 \def\ps@NextWordFoot{}
```

File 140 **l warp-gentombow.sty**

§ 236 Package **gentombow**

Pkg gentombow gentombow is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{gentombow}

2 \newcommand{\settombowbanner}[1]{}
3 \newcommand{\settombowbannerfont}[1]{}
4 \newcommand{\settombowwidth}[1]{}
5 \newcommand{\settombowbleed}[1]{}
6 \newcommand{\settombowcolor}[1]{}
```

File 141 **l warp-geometry.sty**

§ 237 Package **geometry**

(Emulates or patches code by HIDEO UMEKI.)

Pkg geometry geometry is preloaded by l warp, but must be nullified as seen by the user's source code.

for HTML output: Discard all options for l warp-geometry:

```
1 \LWR@ProvidesPackageDrop{geometry}

2 \renewcommand*\{geometry}[1]{}
3 \renewcommand*\{newgeometry}[1]{}
4 \renewcommand*\{restoregeometry}{}
5 \renewcommand*\{savegeometry}[1]{}
6 \renewcommand*\{loadgeometry}[1]{}
```

File 142 **lwarf-glossaries.sty**

§ 238 Package **glossaries**

(Emulates or patches code by NICOLA L.C. TALBOT.)

Pkg **glossaries**
processing glossaries
 Opt **GlossaryCmd**
 Default: **makeglossaries**

Opt **lwarfmk printglossary**
 Opt **lwarfmk htmlglossary**

⚠ **makeglossaries not found** In some situations it may be required to modify the default command, such as to add the **perl** command in front:

```
\usepackage[
  GlossaryCmd={perl makeglossaries},
] {lwarf}
```

xindy language To set the language to use for processing glossaries with *xindy*:

```
\usepackage[
  GlossaryCmd={makeglossaries -L english},
] {lwarf}
```

Other options for **makeglossaries** may be set as well.

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

Numbered section, on its own HTML page:

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

Unnumbered section, inline with the current HTML page:

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

Unnumbered section, on its own HTML page:

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

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

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

print/HTML versions The print and HTML versions of the glossary differ in their internal page numbers. Separate commands for generating print and HTML glossaries are used, even though the page number is currently ignored.

for HTML output:

```
1 \PassOptionsToPackage{xindy}{glossaries}
2 \LWRProvidesPackagePass{glossaries}
3 \setupglossaries{nonumberlist}
4 \setglossarystyle{index}
```

Patched to fix TOC pointing to the previous page:

```
5 \renewcommand*{\@p@glossarysection}[2]{%
6   \glsclearpage
7   \phantomsection
8   \ifdef\empty\@@glossarysecstar
9   {%
10     \csname\@@glossarysec\endcsname{#2}%
11   }%
12   {%
```

In the original, the TOC entry was made before the section, thus linking to the phantomsection in the printed version, but for HTML, this caused the link to point to the page before the glossaries, which could be a different HTML file. Here, the TOC entry is made after the section is created:

```
13   \csname\@@glossarysec\endcsname*{#2}%
14   \gls@toc{#1}{\@@glossarysec}%
15 }%
16 \@@glossaryseclabel
17 }
```

`lwarp`'s sectioning commands cannot handle robust macros when splitting HTML into named filenames. `glossaries` uses `\translate` in sectioning names, and `\translate` is robust and cannot be expanded. The following pre-expands the translations at this moment, making use of `\translatelet`.

```

18 \newcommand*{\LWR@comp@glossaryname}{\translate{Glossary}}
19
20 \ifdefstreq{\glossaryname}{\LWR@comp@glossaryname}{
21     \translatelet{\LWR@translatetemp}{\glossaryname}
22     \edef{\glossaryname}{\LWR@translatetemp}
23 }{}
24
25 \newcommand*{\LWR@comp@acronymname}{\translate{Acronym}}
26
27 \ifdefstreq{\acronymname}{\LWR@comp@acronymname}{
28     \translatelet{\LWR@translatetemp}{\acronymname}
29     \edef{\acronymname}{\LWR@translatetemp}
30 }{}
31
32 \newcommand*{\LWR@comp@glssymbolsgroupname}{\translate{Symbols (glossaries)}}
33
34 \ifdefstreq{\glssymbolsgroupname}{\LWR@comp@glssymbolsgroupname}{
35     \translatelet{\LWR@translatetemp}{\glssymbolsgroupname}
36     \edef{\glssymbolsgroupname}{\LWR@translatetemp}
37 }{}
38
39 \newcommand*{\LWR@comp@glsnumbersgroupname}{\translate{Numbers (glossaries)}}
40
41 \ifdefstreq{\glsnumbersgroupname}{\LWR@comp@glsnumbersgroupname}{
42     \translatelet{\LWR@translatetemp}{\glsnumbersgroupname}
43     \edef{\glsnumbersgroupname}{\LWR@translatetemp}
44 }{}

```

File 143 **lwarf-gmeometric.sty**

§ 239 Package **gmeometric**

Pkg **gmeometric** *gmeometric* is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop[gmeometric]

File 144 **lwarf-graphics.sty**

§ 240 Package **graphics**

(Emulates or patches code by D. P. CARLISLE.)

Pkg **graphics** *graphics* is emulated.

for HTML output: 1 \LWR@ProvidesPackagePass[graphics]

§ 240.1 Graphics extensions

```
\DeclareGraphicsExtensions {⟨list⟩}
```

`\AtBeginDocument` allow SVG files instead of PDF:

```
2 \AtBeginDocument{  
3 \DeclareGraphicsExtensions{.svg,.SVG,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}  
4 \DeclareGraphicsRule{.svg}{svg}{.svg}{}  
5 \DeclareGraphicsRule{.SVG}{svg}{.SVG}{}  
6 }
```

Inside a `lateximage`, allow PDF instead of SVG:

```
7 \ifpdf  
8 \appto\LWR@restoreorigformatting{  
9 \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%  
10 }  
11 \else% \ifpdf  
12 \ifXeTeX  
13 \appto\LWR@restoreorigformatting{  
14 \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%  
15 }  
16 \else  
17 \appto\LWR@restoreorigformatting{  
18 \DeclareGraphicsExtensions{.eps,.EPS,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%  
19 }  
20 \fi  
21 \fi
```

§ 240.2 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.

Used to store the user's selected dimensions and HTML class.

The class defaults to "inlineimage" unless changed by a `class=xyz` option.

```
22 \newlength{\LWR@igwidth}  
23 \newlength{\LWR@igheight}  
24 \newcommand*{\LWR@igwidthstyle}{}  
25 \newcommand*{\LWR@igheightstyle}{}  
26 \newcommand*{\LWR@igorigin}{}  
27 \newcommand*{\LWR@igangle}{}  
28 \newcommand*{\LWR@igxscale}{1}  
29 \newcommand*{\LWR@igyscale}{1}  
30 \newcommand*{\LWR@igclass}{inlineimage}
```

```
31 \newcommand*{\LWR@igalt}{(image)}
```

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:

```
32 \define@key{igraph}{width}{%
33 \setlength{\LWR@igwidth}{#1}%
34 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}{%
35 {}}
```

Default to use the converted fixed length given:

```
36 \renewcommand*{\LWR@igwidthstyle}[width:\LWR@printlength{\LWR@igwidth}]%
```

If ex or em dimensions were given, use those instead:

```
37 \IfEndWith{#1}{ex}{%
38 {\renewcommand*{\LWR@igwidthstyle}[width:#1]{} yes ex
39 {} not ex
40 \IfEndWith{#1}{em}{%
41 {\renewcommand*{\LWR@igwidthstyle}[width:#1]{} yes em
42 {} not em
43 \IfEndWith{#1}{\%}{%
44 {\renewcommand*{\LWR@igwidthstyle}[width:#1]{} yes percent
45 {} not percent
46 \IfEndWith{#1}{px}{%
47 {\renewcommand*{\LWR@igwidthstyle}[width:#1]{} yes px
48 {} not px
49 }{}} end of length > 0pt
50 }}
```

If an optional height was given, set an HTML style:

```
51 \define@key{igraph}{height}{%
52 \setlength{\LWR@igheight}{#1}%
53 \ifthenelse{\lengthtest{\LWR@igheight > 0pt}}{%
54 {}}
```

Default to use the converted fixed length given:

```
55 \renewcommand*{\LWR@igheightstyle}[%
56 height:\LWR@printlength{\LWR@igheight} % extra space
57 {}]
```

If ex or em dimensions were given, use those instead:

```

58  \IfEndWith{#1}{ex}%
59  {\renewcommand*\LWR@igheightstyle{height:#1}}% yes ex
60  {}% not ex
61  \IfEndWith{#1}{em}%
62  {\renewcommand*\LWR@igheightstyle{height:#1}}% yes em
63  {}% not em
64  \IfEndWith{#1}{\%}%
65  {\renewcommand*\LWR@igheightstyle{height:#1}}% yes percent
66  {}% not percent
67  \IfEndWith{#1}{px}%
68  {\renewcommand*\LWR@igheightstyle{height:#1}}% yes px
69  {}% not px
70 }{}% end of length > 0pt
71 }

```

Handle origin key:

```

72 \define@key{igraph}{origin}[c]{%
73   \renewcommand*\LWR@igorigin{#1}%
74 }

```

Handle angle key:

```
75 \define@key{igraph}{angle}{\renewcommand*\LWR@igangle{#1}}
```

Handle class key:

```
76 \define@key{igraph}{class}{\renewcommand*\LWR@igclass{#1}}
```

Handle alt key:

```
77 \define@key{igraph}{alt}{\renewcommand*\LWR@igalt{#1}}
```

It appears that `graphicx` does not have separate keys for `xscale` and `yscale`. `scale` adjusts both at the same time.

```

78 \define@key{igraph}{scale}{%
79   \renewcommand*\LWR@igxscale{#1}%
80   \renewcommand*\LWR@igyscale{#1}%
81 }

```

Numerous ignored keys:

```

82 \define@key{igraph}{bb}(){}
83 \define@key{igraph}{bbllx}){}
84 \define@key{igraph}{bbly}){}
85 \define@key{igraph}{bburx}){}
86 \define@key{igraph}{bbury}){}

```

```

87 \define@key{igraph}{natwidth}{}{}}
88 \define@key{igraph}{natheight}{}{}}
89 \define@key{igraph}{hiresbb}[true]{}{}}
90 \define@key{igraph}{viewport}{}{}}
91 \define@key{igraph}{trim}{}{}}
92 \define@key{igraph}{totalheight}{}{}}
93 \define@key{igraph}{keepaspectratio}[true]{}{}}
94 \define@key{igraph}{clip}[true]{}{}}
95 \define@key{igraph}{draft}[true]{}{}}
96 \define@key{igraph}{type}{}{}}
97 \define@key{igraph}{ext}{}{}}
98 \define@key{igraph}{read}{}{}}
99 \define@key{igraph}{command}{}{}}

```

New in v1.1a:

```

100 \define@key{igraph}{quite}{}{}}
101 \define@key{igraph}{page}{}{}}
102 \define@key{igraph}{pagebox}{}{}}
103 \define@key{igraph}{interpolate}[true]{}{}}

```

New in v1.1b:

```
104 \define@key{igraph}{decodearray}{}{}}
```

§ 240.3 Printing HTML styles

`\LWR@rotstyle {<prefix>} {<degrees>}`

Prints the rotate style with the given prefix.

prefix is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:rotate` style.

```

105 \newcommand*{\LWR@rotstyle}[2]{%
106   #1transform:rotate(-#2deg);%
107 }

```

`\LWR@scalesstyle {<prefix>} {<xscale>} {<yscale>}`

Prints the scale style with the given prefix.

prefix is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:scale` style.

```

108 \newcommand*{\LWR@scalesstyle}[3]{%
109   #1transform:scale(#2,#3);%
110 }

```

§ 240.4 \includegraphics

Bool LWR@infloatrow Used to compute \linewidth.

```
111 \newbool{LWR@infloatrow}
112 \boolfalse{LWR@infloatrow}
```

\LWR@opacity For HTML, used only for \includegraphics.

\LWR@opacity may be set by the transparent package.

```
113 \def\LWR@opacity{1}
```

\LWR@imagesizebox Used to determine the actual image size if needed.

```
114 \newsavebox{\LWR@imagesizebox}
```

\LWR@HTML@Gin@setfile {\langle w \rangle} {\langle h \rangle} {\langle filename \rangle} Sets the parsed filename for HTML output.

```
115 \newcommand*{\LWR@HTML@Gin@setfile}[3]{%
116   \xdef\LWR@parsedfilename{\#3}%
117 }
```

Key Gin class CSS class for the image.

Define the new class key for the print-mode version of \includegraphics, which is enabled inside a lateximage.

```
118 \AtBeginDocument{
119 \define@key{Gin}{class}{}
120 \define@key{Gin}{alt}{}
121 }
```

\LWR@replaceEPSSVG Usually, references to EPS files become SVG files, but if the epstopdf package is being used, it automatically converts EPS to PDF, and the following must NOT be done.

```
122 \AtBeginDocument{
123 \@ifpackageloaded{epstopdf}
124 {
125   \newcommand*{\LWR@replaceEPSSVG}(){}
126 }{%
127   \newcommand*{\LWR@replaceEPSSVG}{%
128     \StrSubstitute{\LWR@tempone}{.eps}{.svg}[\LWR@tempone]%
129     \StrSubstitute{\LWR@tempone}{.EPS}{.SVG}[\LWR@tempone]%
130   }
131 }{%
132 }
```

```
\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.
```

If using \epsfig or \psfig from the epsfig package, #4 will be \LWR@epsfig@filename, which will have been set by the file or figure keys. Therefore, #4 must not be used until after the keys have been processed.

```
133 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
134 {%
```

Start the image tag on a new line, allow PDF output word wrap:

```
135 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute \linewidth, \textwidth, \textheight arguments with a 6x9 inch size until the next \endgroup.

```
136 \begingroup%
137 \ifthenelse{\cnttest{\value{\LWR@minipagedepth}}=}{0}%
138 {%
139   \ifbool{\LWR@infloatrow}%
140   {}%
141   {%
142     \setlength{\linewidth}{6in}%
143     \setlength{\textwidth}{6in}%
144     \setlength{\textheight}{9in}%
145   }%
146 }{%
147 }
```

For correct em sizing during the width and height conversions:

```
147 \large%
```

Reset some defaults, possibly will be changed below if options were given:

```
148 \setlength{\LWR@igwidth}{Opt}%
149 \setlength{\LWR@igheight}{Opt}%
150 \renewcommand*{\LWR@igwidthstyle}{}%
151 \renewcommand*{\LWR@igheightstyle}{}%
152 \renewcommand*{\LWR@igorigin}{}%
153 \renewcommand*{\LWR@igangle}{}%
154 \renewcommand*{\LWR@igxscale}{1}%
155 \renewcommand*{\LWR@igyscale}{1}%
156 \renewcommand*{\LWR@igclass}{inlineimage}%
157 }
```

```
157 \renewcommand*{\LWR@igalt}{(image)}%
```

If #3 is empty, only one optional argument was given, thus `graphicx` syntax:

```
158 \IfValueF{#3}{%
159     \IfValueTF{#2}{%
160         {\setkeys{igraph}{#2}}%
161         {\setkeys{igraph}{}}
162 }%
```

Fully expand and detokenize the filename, changing the file extension to `.svg` if necessary.

```
163 \begingroup%
164 \LetLtxMacro\Gin@setfile{\LWR@HTML@Gin@setfile}%
165 \edef\LWR@tempone{#4}%
166 \StrSubstitute{\LWR@tempone}{.pdf}{.svg}[\LWR@tempone]%
167 \StrSubstitute{\LWR@tempone}{.PDF}{.SVG}[\LWR@tempone]%
168 \LWR@replaceEPSSVG%
169 \xdef\LWR@parsedfilename{\LWR@tempone}%
170 \Ginclusion@graphics{\detokenize\expandafter{\LWR@tempone}}%
171 \endgroup%
172 \filename@parse{\LWR@parsedfilename}%
173 \LWR@traceinfo{\LWR@parsedfilename is \LWR@parsedfilename}%
174 % \LWR@sanitize{\LWR@parsedfilename}%

```

If formatting for a word processor, find and set the actual image size, without rotation, using PDF instead of SVG to find the original bounding box:

```
175 \ifbool{FormatWP}{%
176     \begingroup%
177     \LWR@restoreorigformatting%
178     \ifpdf%
179         \appto{\LWR@restoreorigformatting}{%
180             \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
181         }%
182     \else%
183         \ifXeTeX%
184             \appto{\LWR@restoreorigformatting}{%
185                 \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
186             }%
187             \else%
188                 \appto{\LWR@restoreorigformatting}{%
189                     \DeclareGraphicsExtensions{.eps,.EPS,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
190                 }%
191             \fi%
192     \fi% \ifpdf%
193     \define@key{Gin}{angle}{%
194         \IfBooleanTF{#1}{%
```

```

195 {%
196     \IfValueTF{#3}%
197     {%
198         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics*[#2][#3]{#4}}%
199     }%
200     {%
201         \IfValueTF{#2}%
202         {%
203             \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics*[#2]{#4}}%
204         }{%
205             \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics*[#4]}%
206         }%
207     }%
208 }% starred
209 {%
210     \IfValueTF{#3}%
211     {%
212         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics[#2][#3]{#4}}%
213     }%
214     {%
215         \IfValueTF{#2}%
216         {%
217             \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics[#2]{#4}}%
218         }{%
219             \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics{#4}}%
220         }%
221     }%
222 }% not starred
223 \endgroup%
224 \settowidth{\LWR@igwidth}{\usebox{\LWR@imagesizebox}}%
225 \global\renewcommand*{\LWR@igwidthstyle}{width:\LWR@printlength{\LWR@igwidth}}%
226 \settoheight{\LWR@igheight}{\usebox{\LWR@imagesizebox}}%
227 \global\renewcommand*{\LWR@igheightstyle}{height:\LWR@printlength{\LWR@igheight}}%
228 }% FormatWP

```

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.

```
229 \LWR@traceinfo{\LWR@includegraphicsb: about to create href}%
230 \href{\LWR@parsedfilename}%
231 {%
232 \LWR@traceinfo{\LWR@includegraphicsb: about to \LWR@htmltag}%
233 \LWR@htmltag{%
234 img src="%"}

235 \detokenize\expandafter{\LWR@parsedfilename}%
```

```
236 " \LWR@orignewline%
```

Only include a style tag if a width, height, angle, or scale was given:

```
237 \ifthenelse{%
238     \NOT\equal{\LWR@igwidthstyle}{} \OR
239     \NOT\equal{\LWR@igheightstyle}{} \OR
240     \NOT\equal{\LWR@igorigin}{} \OR
241     \NOT\equal{\LWR@igangle}{} \OR
242     \NOT\equal{\LWR@igxscale}{1} \OR
243     \NOT\equal{\LWR@igyscale}{1}
244 }%
245 {%
246     \LWR@origtilde{} style="%"
247     \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{} }{%
248         {\LWR@igwidthstyle; }{}%
249     }{%
250     \ifthenelse{\NOT\equal{\LWR@igheightstyle}{} }{%
251         {\LWR@igheightstyle; }{}%
252     }{%
253         \ifthenelse{\NOT\equal{\LWR@igorigin}{} }{%
254             {%
255                 \LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; %
256                 \LWR@orignewline%
257             }{}%
258         }{%
259             \ifthenelse{\NOT\equal{\LWR@igangle}{} }{%
260                 {%
261                     \LWR@rotstyle{-ms-}{\LWR@igangle} % extra space
262                     \LWR@rotstyle{-webkit-}{\LWR@igangle} % extra space
263                     \LWR@rotstyle{}{\LWR@igangle }%
264                 }{}%
265             }{%
266             \ifthenelse{%
267                 \NOT\equal{\LWR@igxscale}{1}\OR%
268                 \NOT\equal{\LWR@igyscale}{1}%
269             }{%
270                 {%
271                     \LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale} % extra space
272                     \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale} % extra space
273                     \LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyscale}%
274                 }{}%
275             }{%
276             " \LWR@orignewline%
277         }{}%
278     }{%
279     }
```

Set the class and alt tag:

```
277 \LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%
278 \LWR@origtilde{} alt="\LWR@igalt" \LWR@orignewline%
```

```
279 }% end of image tags
280 }% end of href
```

Return to original page size and font size:

```
281 \endgroup
282 \LWR@traceinfo{LWR@includegraphicsb done}%
283 }
```

\includegraphics [⟨key=val⟩] {⟨filename⟩}

Handles width and height, converted to fixed width and heights.

The user should always use no file suffix in the document source.

```
284 \AtBeginDocument{
285
286 \LWR@traceinfo{Patching includegraphics.}
287
288 \LetLtxMacro\LWR@originincludegraphics\includegraphics
289
290 \renewcommand*\{\includegraphics}
291 {%
```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```
292 \LWR@traceinfo{includegraphics}%
293 \LWR@ensuredoingapar%
294 \LWR@includegraphicsb%
295 }% includegraphics
296 }% AtBeginDocument
```

§ 240.5 Boxes

\LWR@rotboxorigin Holds the origin key letters.

```
297 \newcommand*\{\LWR@rotboxorigin}{}{}
```

\LWR@originname {⟨letter⟩}

Given one L^AT_EX origin key value, translate into an HTML origin word:

```
298 \newcommand*\{\LWR@originname}[1]{%
299 \ifthenelse{\equal{#1}{t}}{top}{%
300 \ifthenelse{\equal{#1}{b}}{bottom}{}}
```

```

301 \ifthenelse{\equal{#1}{c}}{center}{}%
302 \ifthenelse{\equal{#1}{l}}{left}{}%
303 \ifthenelse{\equal{#1}{r}}{right}{}%
304 }

\LWR@originnames {⟨letters⟩}

```

Given one- or two-letter L^AT_EX origin key values, translate into HTML origin words:

```

305 \newcommand*{\LWR@originnames}[1]{%
306 \StrChar{#1}{1}[\LWR@strresult]%
307 \LWR@originname{\LWR@strresult}%
308 \StrChar{#1}{2}[\LWR@strresult]%
309 \LWR@originname{\LWR@strresult}%
310 }

```

Handle the origin key for \rotatebox:

```

311 \define@key{krotbox}{origin}{%
312 \renewcommand*{\LWR@rotboxorigin}{#1}%
313 }

```

These keys are ignored:

```

314 \define@key{krotbox}{x}{}
315 \define@key{krotbox}{y}{}
316 \define@key{krotbox}{units}{}

```

\rotatebox [⟨keyval list⟩] {⟨angle⟩} {⟨text⟩}

```
317 \AtBeginDocument{
```

The HTML version:

```
318 \NewDocumentCommand{\LWR@HTML@rotatebox}{O{} m +m}{%
```

Reset the origin to “none-given”:

```
319 \renewcommand*{\LWR@rotboxorigin}{}
```

Process the optional keys, which may set \LWR@rotateboxorigin:

```
320 \setkeys{krotbox}{#1}%
```

Select inline-block so that HTML will transform this span:

```
321 \LWR@htmlltagc{span style="display: inline-block; %
```

If an origin was given, translate and print the origin information:

```
322 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}%
323 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%
```

Print the rotation information:

```
324 \LWR@rotstyle{-ms-}{}{#2} % extra space
325 \LWR@rotstyle{-webkit-}{}{#2} % extra space
326 \LWR@rotstyle{}{}{#2} % extra space
327 "}\LWR@orignewline%
```

Print the text to be rotated:

```
328 \begin{LWR@nestspan}%
329 #3%
```

Close the span:

```
330 \LWR@htmntagc{/span}%
331 \end{LWR@nestspan}%
332 }
```

The high-level interface:

```
333 \LWR@formatted{rotatebox}
334
335 }% AtBeginDocument
```

\scalebox {⟨h-scale⟩} [⟨v-scale⟩] {⟨text⟩}

```
336 \AtBeginDocument{
```

The HTML version:

```
337 \NewDocumentCommand{\LWR@HTML@scaledbox}{m o m}{%
```

Select inline-block so that HTML will transform this span:

```
338 \LWR@htmntagc{span style="display: inline-block; %
```

Print the scaling information:

```
339 \LWR@scalestyle{-ms-}{}{#1}{\IfNoValueTF{#2}{#1}{#2}} % extra space
340 \LWR@scalestyle{-webkit-}{}{#1}{\IfNoValueTF{#2}{#1}{#2}} % extra space
341 \LWR@scalestyle{}{}{#1}{\IfNoValueTF{#2}{#1}{#2}} % extra space
342 "}%
```

Print the text to be scaled:

```
343 \begin{LWR@nestspan}%
344 #3%
```

Close the span:

```
345 \LWR@htmntagc{/span}%
346 \end{LWR@nestspan}%
347 }
```

The high-level interface:

```
348 \LWR@formatted[scalebox]%
349
350 }% AtBeginDocument
```

```
\reflectbox {\langle text\rangle}

351 \AtBeginDocument{
352
353 \newcommand{\LWR@HTML@reflectbox}[1]{%
354 \scalebox{-1}[1]{\#1}%
355 }% \reflectbox
356
357 \LWR@formatted[reflectbox]
358
359 }% AtBeginDocument
```

```
\resizebox {\langle h-length\rangle} {\langle v-length\rangle} {\langle text\rangle}
```

Simply prints its text argument.

```
360 \AtBeginDocument{
361
362 \NewDocumentCommand{\LWR@HTML@resizebox}{s m m}{%
363 #4%
364 }
365
366 \LWR@formatted[resizebox]
367
368 }% AtBeginDocument
```

File 145 **l warp-graphicx.sty**

§ 241 Package **graphicx**

Pkg **graphicx** **graphicx** is emulated.

graphicx loads **graphics**, which also loads **l warp-graphics**, which remembers the original **graphics** definitions for use inside a **lateximage**, and then patches them **\AtBeginDocument** for **HTML** output.

l warp-graphics handles the syntax of either **graphics** or **graphicx**.

for HTML output: 1 \LWR@ProvidesPackagePass{graphicx}[2017/06/01]

File 146 **l warp-grffile.sty**

§ 242 Package **grffile**

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



matching PDF and SVG

l warp-grffile now exists as a placeholder since **grffile** used to be emulated by **l warp**, and thus older versions of **l warp-grffile** may exist and should be overwritten by this newer version.

for HTML output: 1 \LWR@ProvidesPackagePass{grffile}

File 147 **l warp-grid.sty**

§ 243 Package **grid**

Pkg **grid** **grid** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{grid}

2 \newenvironment*{gridenv}{}{}

File 148 **l warp-grid-system.sty**

§ 244 Package **grid-system**

(Emulates or patches code by MARCUS BITZL.)

Pkg grid-system grid-system is patched for use by l warp.

for HTML output: 1 \LWR@ProvidesPackagePass{grid-system}[2014/02/16]

(\ifdef is in case the older syntax is removed.)

```
2 \AtBeginEnvironment{Row}{\setlength{\ linewidth}{6in}}
3
4 \ifdef{\endrow}{%
5   \AtBeginEnvironment{row}{\setlength{\ linewidth}{6in}}%
6 }{}%
7
8 \renewcommand{\gridsystem@finishcell}{\hspace{\gridsystem@cellsep}}
```

File 149 **l warp-gridset.sty**

§ 245 Package **gridset**

Pkg gridset gridset is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{gridset}

```
2 \newcommand*{\gridbase}{}%
3 \newcommand*{\gridinterval}{}%
4 \newcommand*{\savepos}[1]{}%
5 \newcounter{gridcnt}%
6 \newcommand*{\vskipnextgrid}{}%
7 \newcommand*{\thegridinfo}[1]{}%
8 \newcommand*{\theinfo}[1]{}%
9 \newcommand*{\theypos}[1]{}%
```

File 150 **l warp-hang.sty**

§ 246 Package **hang**

(Emulates or patches code by ANDREAS NOLDA.)

Pkg **hang** **hang** is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{hang}

2 \newlength{\hangingindent}
3 \setlength{\hangingindent}{1em}
4 \newlength{\hangingleftmargin}
5 \setlength{\hangingleftmargin}{0em}
6
7 \newcommand*{\LWR@findhangingleftmargin}{%
8 \setlength{\LWR@templengthone}{\hangingleftmargin}%
9 \addtolength{\LWR@templengthone}{\hangingindent}%
10 }
11
12 \newenvironment{hangingpar}
13 {
14     \LWR@findhangingleftmargin%
15     \BlockClass[% 
16         \LWR@print@mbox{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
17         \LWR@print@mbox{text-indent:-\LWR@printlength{\hangingindent}}% 
18     ]%
19     {hangingpar}%
20 }
21 {\endBlockClass}
22
23 \newenvironment{hanginglist}
24 {
25     \renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}%
26     \renewcommand*{\LWR@printopenlist}{%
27         \LWR@findhangingleftmargin%
28         ul style=%
29             \LWR@print@mbox{list-style-type:none;} % extra space
30             \LWR@print@mbox{%
31                 margin-left:\LWR@printlength{\LWR@templengthone}%
32             } ; % extra space
33             \LWR@print@mbox{%
34                 text-indent:-\LWR@printlength{\hangingindent}}%
35         }%
36     "%
```

```

37      }%
38      \let\item\LWR@itemizeitem%
39      \list{}{ }%
40 }
41 {\endlist}
42
43 \newenvironment{compacthang}
44 {\hanginglist}
45 {\endhanginglist}
46
47 \newlength{\labeledleftmargin}
48 \setlength{\labeledleftmargin}{0em}
49
50 \newenvironment{labeledpar}[2]
51 {%
52     \BlockClass[%
53         \LWR@findhangingleftmargin%
54         \LWR@print@mbox{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
55         \LWR@print@mbox{text-indent:-\LWR@printlength{\hangingindent}}]%
56     ]{\labeledpar}#2%
57 }
58 {\endBlockClass}
59
60 \newenvironment{labeledlist}[1]
61 {\hanginglist}
62 {\endhanginglist}
63
64 \newenvironment{compactlabel}[1]
65 {\hanginglist}
66 {\endhanginglist}

```

File 151 **l warp-hanging.sty**

§ 247 Package **hanging**

Pkg **hanging** **hanging** is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{hanging}

2 \@ifclassloaded{memoir}{
3 \let\hangpara\relax
4 \let\hangparas\relax
5 \let\endhangparas\relax
6 \let\hangpunct\relax
7 \let\endhangpunct\relax
8 }{ }

```

```

\hangpara  {\langle indent\rangle} {\langle afternum\rangle}
    Use hangparas instead.
    9 \newcommand*{\hangpara}[2]{}

Env  hangparas  {\langle indent\rangle} {\langle afternum\rangle}
    10 \newenvironment*{hangparas}[2]
    11 {%
    12     \BlockClass[%
    13         \LWR@print@mbox{margin-left:\LWR@printlength{\#1}} ; %
    14         \LWR@print@mbox{text-indent:-\LWR@printlength{\#1}}%]
    15     ]%
    16     {hangingpar}%
    17 }
    18 {\endBlockClass}

Env  hangpunct
    19 \newenvironment*{hangpunct}
    20 {\BlockClass{hangpunct}}
    21 {\endBlockClass}

    22 \newcommand{\nhpt}{.}
    23 \newcommand{\nhlq}{`}
    24 \newcommand{\nhrq}{'}

```

File 152 lwarf-hypcap.sty

§ 248 Package **hypcap**

Pkg hypcap hypcap is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{hypcap}

2 \newcommand*{\capstart}(){}
3 \newcommand*{\hypcapspace}){}
4 \newcommand*{\hypcapedef}[1]{}%
5 \newcommand*{\capstartfalse}{}%
6 \newcommand*{\capstarttrue}{}%
```

File 153 **l warp-hypdestopt.sty**

§ 249 Package **hypdestopt**

Pkg hypdestopt hypdestopt is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{hypdestopt}

File 154 **l warp-hypernat.sty**

§ 250 Package **hypernat**

Pkg hypernat hypernat is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{hypernat}

File 155 **l warp-hyperref.sty**

§ 251 Package **hyperref**

(Emulates or patches code by SEBASTIAN RAHTZ, HEIKO OBERDIEK.)

Pkg hyperref hyperref is emulated.

for HTML output:

```
1 \% \LWR@ProvidesPackageDrop{hyperref}
2 \typeout{Using the l warp html version of package 'hyperref', discarding options.}
3 \typeout{    Are not using ProvidesPackage, so that other packages}
4 \typeout{    do not attempt to patch l warp's version of 'hyperref'.}
5 \% \ProvidesPackage{l warp-\#1-\#2}
6 \DeclareOption*{}
7 \ProcessOptions\relax

8 \newcommand*\{\hypersetup}[1]{}
9 \newcommand*\{\hyperbaseurl}[1]{}
```

\hyperimage {\langle URL\rangle} {\langle alt text\rangle}

Insert an image with alt text:

```

10 \NewDocumentCommand{\LWR@hyperimageb}{m +m}{%
11 \LWR@ensuredoingapar%
12 \def\LWR@templink{#1}%
13 \onelevel@sanitize\LWR@templink%
14 \LWR@htmlltag{img src="\LWR@templink" alt="#2" class="hyperimage"}%
15 \LWR@ensuredoingapar%
16 \endgroup%
17 }
18
19 \newrobustcmd*\hyperimage{%
20 \begingroup%
21 \catcode`\#=12%
22 \catcode`\%=12%
23 \catcode`\&=12%
24 \catcode`\~=12%
25 \catcode`\_=12%
26 \LWR@hyperimageb%
27 }
28

```

\hyperdef {<1: category>} {<2: name>} {<3: text>}

Creates an HTML anchor to category.name with the given text.

```

29 \NewDocumentCommand{\LWR@hyperdef}{m m +m}{%
30 \LWR@ensuredoingapar%
31 \LWR@sublabel{#1.#2}%
32 #3%
33 \endgroup%
34 }
35
36 \newcommand*\hyperdef{%
37 \begingroup%
38 \catcode`\#=12%
39 \catcode`\%=12%
40 \catcode`\&=12%
41 \catcode`\~=12%
42 \catcode`\_=12%
43 \LWR@hyperdefb%
44 }
45

```

\LWR@hyperrefb {<1: URL>} {<2: category>} {<3: name>} {<4: text>}

Creates an HTML link to URL#category.name with the given text.

```

46 \newcommand{\LWR@hyperreffinish}[1]{%
47 \begingroup%
48 \RenewDocumentCommand{\ref}{}{\LWR@ref@ignorestar}%
49 #1%

```

```

50 \endgroup%
51 \LWR@htmlltag{/a}%
52 }
53
54 \newcommand*{\LWR@hyperrefbb}[3]{%
55 \LWR@htmlltag{%
56     a href="%"
57         \detokenize\expandafter{\#1}\LWR@hashmark%
58         \detokenize\expandafter{\#2}. \detokenize\expandafter{\#3}%
59     "%"
60 }%
61 \endgroup%
62 \LWR@hyperreffinish%
63 }
64
65 \newrobustcmd*{\LWR@hyperrefb}{%
66 \begingroup%
67 \catcode`\#=12%
68 \catcode`\%=12%
69 \catcode`\&=12%
70 \catcode`\~=12%
71 \catcode`\_=12%
72 \LWR@hyperrefbb%
73 }

```

\LWR@hyperrefc [⟨label⟩] {⟨text⟩}

Creates text as an `LATEX` link to the `LATEX` label.

```

74
75 \NewDocumentCommand{\LWR@hyperrefcb}{O{label}}{%
76 \LWR@startref{#1}%
77 \endgroup%
78 \LWR@hyperreffinish%
79 }
80
81 \newcommand*{\LWR@hyperrefc}{%
82 \begingroup%
83 \catcode`\#=12%
84 \catcode`\%=12%
85 \catcode`\&=12%
86 \catcode`\~=12%
87 \catcode`\_=12%
88 \LWR@hyperrefcb%
89 }

```

\hyperref {⟨1: URL⟩} {⟨2: category⟩} {⟨3: name⟩} {⟨4: text⟩} — or —
[⟨1: label⟩] {⟨2: text⟩}

```
90 \DeclareRobustCommand*{\hyperref}{%
```

```
91 \LWR@ensuredoingapar%
92 \@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
93 }
```

\hypertarget {⟨name⟩} {⟨text⟩}

Creates an anchor to name with the given text.

```
94 \NewDocumentCommand{\hypertargetb}{m +m}{%
95 \label{LWR-ht-#1}%
96 #2%
97 \endgroup%
98 }%
99
100 \newcommand*{\hypertarget}{%
101 \begingroup%
102 \catcode`\#=12%
103 \catcode`\%=12%
104 \catcode`\&=12%
105 \catcode`\~=12%
106 \catcode`\_=12%
107 \LWR@hypertargetb%
108 }
```

\hyperlink {⟨name⟩} {⟨text⟩}

Creates a link to the anchor created by hypertarget, with the given link text.

Declared because also defined by memoir.

```
109 \DeclareDocumentCommand{\hyperlinkb}{m}{%
110 \LWR@hyperrefcb[LWR-ht-#1]%
111 }%
112
113 \DeclareDocumentCommand{\hyperlink}{}
114 \LWR@ensuredoingapar%
115 \begingroup%
116 \catcode`\#=12%
117 \catcode`\%=12%
118 \catcode`\&=12%
119 \catcode`\~=12%
120 \catcode`\_=12%
121 \LWR@hyperlinkb%
122 }
```

\autoref * {⟨label⟩}

For HTML, \cleveref is used instead.

```
123 \NewDocumentCommand{\autoref}{s m}{%
124 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}}
```

```
125 }

\autopageref  {\langle label\rangle}
For HTML, \cleverref is used instead.

126 \NewDocumentCommand{\autopageref}{s m}{%
127 \IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
128 }

\pdfstringdef {\langle macroname\rangle} {\langle TEXstring\rangle}
129 \newcommand{\pdfstringdef}[2]{}

\pdfbookmark [{\langle level\rangle}] {\langle text\rangle} {\langle name\rangle}
130 \newcommand{\pdfbookmark}[3]{}

\currentpdfbookmark {\langle text\rangle} {\langle name\rangle}
131 \newcommand{\currentpdfbookmark}[2]{}

\subpdfbookmark {\langle text\rangle} {\langle name\rangle}
132 \newcommand{\subpdfbookmark}[2]{}

\belowpdfbookmark {\langle text\rangle} {\langle name\rangle}
133 \newcommand{\belowpdfbookmark}[2]{}

\texorpdfstring {\langle TEXstring\rangle} {\langle PDFstring\rangle}
134 \newcommand{\texorpdfstring}[2]{#1}

\hypercalcbp {\langle dimen\rangle} From hyperref.
135 \def\hypercalcbp#1{%
136 \strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax
137 }%
```



```
\Acrobatmenu {\langle menuoption\rangle} {\langle text\rangle}
138 \newcommand{\Acrobatmenu}[2]{}

\TextField [\langle parameters\rangle] {\langle label\rangle}
139 \DeclareRobustCommand{\TextField}[2]{}
```

```
\CheckBox  [{parameters}]{label}  
140 \DeclareRobustCommand{\CheckBox}[2][]{}  
  
\ChoiceMenu [{parameters}]{label}{choices}  
141 \DeclareRobustCommand{\ChoiceMenu}[3][]{}  
  
\PushButton [{parameters}]{label}  
142 \DeclareRobustCommand{\PushButton}[2][]{}  
  
\Submit  [{parameters}]{label}  
143 \DeclareRobustCommand{\Submit}[2][]{}  
  
\Reset   [{parameters}]{label}  
144 \DeclareRobustCommand{\Reset}[2][]{}  
  
\Gauge   [{parameters}]{label}  
145 \DeclareRobustCommand{\Gauge}[2][]{}  
  
\LayoutTextField {label}{field}  
146 \newcommand*{\LayoutTextField}[2]{}  
  
\LayoutChoiceField {label}{field}  
147 \newcommand*{\LayoutChoiceField}[2]{}  
  
\LayoutCheckField {label}{field}  
148 \newcommand*{\LayoutCheckField}[2]{}  
  
\MakeRadioField {width}{height}  
149 \newcommand*{\MakeRadioField}[2]{}  
  
\MakeCheckField {width}{height}  
150 \newcommand*{\MakeCheckField}[2]{}  

```

```
\MakeTextField  {\langle width\rangle} {\langle height\rangle}
151 \newcommand*{\MakeTextField}[2]{}

\MakeChoiceField {\langle width\rangle} {\langle height\rangle}
152 \newcommand*{\MakeChoiceField}[2]{}

\MakeFieldButton {\langle text\rangle}
153 \newcommand{\MakeFieldButton}[1]{}
```

File 156 l warp-hyperxmp.sty

§ 252 Package **hyperxmp**

Pkg hyperxmp Emulated.

for HTML output: Discard all options for l warp-hyperxmp:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```

File 157 l warp-hyphenat.sty

§ 253 Package **hyphenat**

Pkg hyphenat hyphenat is emulated during HTML output, while the print-mode version is used inside a `lateximage`.

for HTML output: 1 \LWR@ProvidesPackagePass{hyphenat}

```
2 \LetLtxMacro{\LWRHYNAT@origtextnhtt}{\textnhtt}
3 \LetLtxMacro{\LWRHYNAT@orignhttfamily}{\nhttfamily}
4 \LetLtxMacro{\LWRHYNAT@orignohyphens}{\nohyphens}
5 \LetLtxMacro{\LWRHYNAT@origbshyp}{\bshyp}
6 \LetLtxMacro{\LWRHYNAT@origfshyp}{\fshyp}
7 \LetLtxMacro{\LWRHYNAT@origdothyp}{\dothyp}
8 \LetLtxMacro{\LWRHYNAT@origcolonhyp}{\colonhyp}
9 \LetLtxMacro{\LWRHYNAT@orighyp}{\hyp}
10
11 \LetLtxMacro{\textnhtt}{\texttt}
12 \LetLtxMacro{\nhttfamily}{\ttfamily}
```

```

13
14 \renewcommand{\nohyphens}[1]{#1}
15 \renewrobustcmd{\bshyp}{%
16   \ifmmode\backslash else\textbackslash fi%
17 }
18 \renewrobustcmd{\fshyp}{/}
19 \renewrobustcmd{\dothyp}{.}
20 \renewrobustcmd{\colonhyp}{:}
21 \renewrobustcmd{\hyp}{-}
22
23 \appto\LWR@restoreorigformatting{%
24 \LetLtxMacro{textnhtt}{\LWRHYNAT@origtextnhtt}%
25 \LetLtxMacro{nhttfamily}{\LWRHYNAT@orignhttfamily}%
26 \LetLtxMacro{nohyphens}{\LWRHYNAT@orignohyphens}%
27 \LetLtxMacro{bshyp}{\LWRHYNAT@origbshyp}%
28 \LetLtxMacro{fshyp}{\LWRHYNAT@origfshyp}%
29 \LetLtxMacro{dothyp}{\LWRHYNAT@origdothyp}%
30 \LetLtxMacro{colonhyp}{\LWRHYNAT@origcolonhyp}%
31 \LetLtxMacro{hyp}{\LWRHYNAT@orighyp}%
32 }

```

File 158 **lwarf-idxlayout.sty**

§ 254 Package **idxlayout**

(Emulates or patches code by THOMAS TITZ.)

Pkg **idxlayout** Emulated.

for HTML output: Discard all options for lwarf-idxlayout:

```

1 \LWR@ProvidesPackageDrop{idxlayout}

2 \newcommand{\LWR@indexprenote}{}
3
4 \preto\printindex{
5
6 \LWR@orignewpage
7 \LWR@startpars
8
9 \LWR@indexprenote
10
11 }
12
13 \newcommand{\setindexprenote}[1]{\renewcommand{\LWR@indexprenote}{#1}}
14 \newcommand*{\noindexprenote}{\renewcommand{\LWR@indexprenote}{}}
15

```

```
16 \newcommand{\idxlayout}[1]{}
17 \newcommand*{\indexfont}{}
18 \newcommand*{\indexjustify}{}
19 \newcommand*{\indexsubdelim}{}
20 \newcommand*{\indexstheadcase}{}
```

File 159 **l warp-ifoddpage.sty**

§ 255 Package **ifoddpage**

(Emulates or patches code by MARTIN SCHARRER.)

Pkg **ifoddpage** ifoddpage is emulated.

for HTML output: Discard all options for l warp-ifoddpage:

```
1 \LWR@ProvidesPackageDrop{ifoddpage}

2 \newif\ifoddpage
3
4 \newif\ifoddpageoroneside
5
6 \DeclareRobustCommand{\checkoddpage}{\oddpage=true\oddpageoroneside=false}
7
8 \def\oddpage@page{1}
9
10 \def@ifoddpage{%
11     \expandafter\@firstoftwo
12 }
13
14 \def@ifoddpageoroneside{%
15     \expandafter\@firstoftwo
16 }
```

File 160 **l warp-imakeidx.sty**

§ 256 Package **imakeidx**

(Emulates or patches code by ENRICO GREGORIO.)

Pkg **imakeidx** imakeidx is patched for use by l warp.

letter headings When using *makeindex*, to match the print and HTML output's display of index letter headings, specify the l warp.ist style:

```
\makeindex[options={-s lwarf.ist}]
```

(For HTML the `lwarf.ist` style is used automatically, which displays letter headings. When using `xindy` the default style also displays letter headings.)

index setup See section 9.5.15 for how to setup `lwarfmk` to process the indexes with `imakeidx`, both with and without shell escape.

for HTML output: 1 \LWR@ProvidesPackagePass{imakeidx}

Use the new HTML suffix:

```
2 \catcode`\_=12%
3 \define@key{imki}{name}{\def\imki@name{\#1_html}}
4 \catcode`\_=8%
```

\printindex The HTML version of `\printindex`:

```
5 \catcode`\_=12%
6
7 \renewcommand*{\printindex}[1][\imki@jobname]{%
8 \LWR@orignewpage%
9 \LWR@startpars%
10 \ifstreq{\#1}{\imki@jobname}{%
11   \ifundefined{\#1@idxfile}{%
12     \imki@error{\#1}%
13   }{%
14     \imki@putindex{\#1}%
15   }%
16 }{%
17   \ifundefined{\#1_html@idxfile}{\imki@error{\#1_html}}{\imki@putindex{\#1_html}}%
18 }%
19 }
20
21 \catcode`\_=8%
```

\@index The HTML version of `\index`:

```
22 \catcode`\_=12%
23
24 \def\@index[#1]{%
25   \ifstreq{\#1}{\imki@jobname}{%
26     \ifundefined{\#1@idxfile}{%
27       \imki@error{\#1}%
28     }{%
29       \PackageWarning{imakeidx}{Undefined index file '#1'}%
30       \begingroup
31         \@sanitize
32         \imki@nowrindex%
```

```

33      }%
34      {%
35          \edef\@idxfile{\#1}%
36          \begingroup
37          \csanitize
38          \wrindex\@idxfile%
39      }%
40  }%
41 {%
42     \ifundefined{#1_html@\idxfile}%
43     {%
44         \PackageWarning{imakeidx}{Undefined index file '#1_html'}%
45         \begingroup
46         \csanitize
47         \imki@nowrindex%
48     }%
49     {%
50         \edef\@idxfile{#1_html}%
51         \begingroup
52         \csanitize
53         \wrindex\@idxfile%
54     }%
55   }%
56 }
57
58 \catcode`\_=8%

```

```

\item
\subitem
\subsubitem HTML versions of \item, etc.:
59 \appto\theindex{%
60   \let\item\LWR@indexitem%
61   \let\subitem\LWR@indexsubitem%
62   \let\subsubitem\LWR@indexsubsubitem%
63 }

```

\imki@wrindexentrysplit {{file}} {{entry}} {{page}}
\imki@wrindexentryunique {{file}} {{entry}} {{page}}

While writing index entries, adds an HTML label, and writes the label's index instead of the page number:

```

64 \renewcommand\imki@wrindexentrysplit[3]{%
65 \addtocounter{LWR@autoindex}{1}%
66 \LWR@new@label{LWRindex-\arabic{LWR@autoindex}}%
67 \expandafter\protected@write\csname#1@idxfile\endcsname{}{%
68   {\string\indexentry{#2}{\arabic{LWR@autoindex}}}}%
69 }

```

```

70
71 \renewcommand\imki@wrindexentryunique[3]{%
72 \addtocounter{LWR@autoindex}{1}%
73 \LWR@new@label{LWRindex-\arabic{LWR@autoindex}}%
74 \protected@write\@indexfile{%
75 {\string\indexentry[#1]{#2}{\arabic{LWR@autoindex}}}}%
76 }
77
78 \def\imki@wrindexsplit#1#2{%
79 \imki@wrindexentrysplit{#1}{#2}{\thepage}%
80 \endgroup\imki@showidxentry{#1}{#2}%
81 \esphack%
82 }
83
84 \def\imki@wrindexunique#1#2{%
85 \imki@wrindexentryunique{#1}{#2}{\thepage}%
86 \endgroup\imki@showidxentry{#1}{#2}%
87 \esphack%
88 }
89

```

\LWR@imki@setxdydefopts

Sets the `xindy` HTML options, ignoring the user's settings.

```

90 \newcommand*\LWR@imki@setxdydefopts{%
91   \edef\imki@options{ \space %
92     -M \space \LWR@xindyStyle\space %
93     -L \space \LWR@xindyLanguage\space %
94     -C \space \LWR@xindyCodepage\space %
95   }%
96 }

```

\LWR@imki@setdefopts {*user options*}

Sets the `HTML` options, added to the user's settings, depending on whether `makeindex` or `xindy` are used.

For `makeindex`, the user's choice is ignored, and only the `lwarp` version is used. (Only one style at a time is possible.)

For `xindy`, multiple modules may be specified, and the `lwarp` version is appended.

```

97 \newcommand*\LWR@imki@setdefopts[1]{%
98 \ifblank{#1}{%
99   \edef\imki@options{\space -s \space \LWR@makeindexStyle \space}%
100  \ifdefstring{\imki@progdefault}{xindy}{\LWR@imki@setxdydefopts}{}%
101  \ifdefstring{\imki@progdefault}{texindy}{\LWR@imki@setxdydefopts}{}%
102  \ifdefstring{\imki@progdefault}{truexindy}{\LWR@imki@setxdydefopts}{}%
103 }{%
104   \edef\imki@options{\space #1 \space}%

```

```
105 }%
106 }
```

\imki@makeindex Use the new HTML options:

```
107 \xpatchcmd{\imki@makeindex}
108   {\let\imki@options\space}
109   {\LWR@imki@setdefopts{}}
110   {}
111   {\LWR@patcherror{imakeidx}{makeindex}}
```

Use the new HTML options.

```
112 \define@key{\imki}{options}{\LWR@imki@setdefopts{#1}}
```

\imki@resetdefaults Use the new HTML options:

```
113 \xpatchcmd{\imki@resetdefaults}
114   {\def\imki@options{}}
115   {\LWR@imki@setdefopts{}}
116   {}
117   {\LWR@patcherror{imakeidx}{resetdefaults}}
```

theindex was already defined \AtBeginDocument by the lwarp core, so it must be redefined here similarly, but patched for imakeidx:

Env theindex

```
118 \AtBeginDocument{
119 \renewenvironment*{\theindex}{%
120   \imki@maybeaddtotoc
121   \imki@indexlevel{\indexname}
122   \let\item\LWR@indexitem%
123   \let\subitem\LWR@indexsubitem%
124   \let\subsubitem\LWR@indexsubsubitem%
125 }{}%
126 }% AtBeginDocument
```

Update to the new defaults:

```
127 \imki@resetdefaults
```

Update to the new patches:

\AtBeginDocument is because \wrindex is previously defined as \AtBeginDocument in the lwarp core.

```
128 \ifimki@splitindex
129   \let\imki@startidx\imki@startidxunique
130   \AtBeginDocument{\let\@wrindex\imki@wrindexunique}
```

```

131  \let\imki@putindex\imki@putindexunique
132  \let\imki@wrindexentry\imki@wrindexentryunique
133  \let\imki@startidxsplit@\undefined
134  \let\imki@wrindexsplit@\undefined
135  \let\imki@putindexsplit@\undefined
136 \else
137  \let\imki@startidx\imki@startidxsplit
138  \AtBeginDocument{\let\@wrindex\imki@wrindexsplit}
139  \let\imki@putindex\imki@putindexsplit
140  \let\imki@wrindexentry\imki@wrindexentriesplit
141  \let\imki@startidxunique@\undefined
142  \let\imki@wrindexunique@\undefined
143  \let\imki@putindexunique@\undefined
144 \fi

```

File 161 **l warp-indentfirst.sty**

§ 257 Package **indentfirst**

Pkg **indentfirst** **indentfirst** is ignored.

Discard all options for **l warp-indentfirst**:

for HTML output: 1 \LWR@ProvidesPackageDrop{indentfirst}

File 162 **l warp-index.sty**

§ 258 Package **index**

(Emulates or patches code by DAVID M. JONES.)

Pkg **index** **index** is patched for use by **l warp**.

for HTML output: 1 \LWR@ProvidesPackagePass{index}

Use **\theLWR@autoindex** instead of **\thepage**. **\@tempswattrue** is used to force an immediate write to the index file instead of waiting until the end of the page.

```

2 \xpatchcmd{\newindex}
3   {\x@newindex[\thepage]}
4   {%
5     \@tempswattrue%
6     \x@newindex[\theLWR@autoindex]%
7   }

```

```

8      {}
9      {\LWR@patcherror{index}{newindex}}
10
11 \xpatchcmd{\renewindex}
12   {\x@renewindex[the\page]}
13   {%
14     \@tempswatrue%
15     \x@renewindex[\the\LWR@autoindex]%
16   }
17   {}
18   {\LWR@patcherror{index}{renewindex}}

```

Patched to set a new autoindex:

```

19 \xpatchcmd{\@wrindex}
20   {\begingroup}
21   {%
22     \addtocounter{\LWR@autoindex}{1}%
23     \LWR@new@label{\LWR@index-\arabic{\LWR@autoindex}}% l warp
24     \begingroup% l warp
25   }
26   {}
27   {\LWR@patcherror{index}{@wrindex}}

```

\AtBeginDocument l warp core \lets \@wrindex to \LWR@wrindex. Since the index package has been loaded, \let to its version instead:

```

28 \let\LWR@index@\wrindex@\wrindex
29
30 \AtBeginDocument{
31 \let\@wrindex\LWR@index@\wrindex
32 }

```

Modified to add \index@prologue:

```

33 \AtBeginDocument{
34 \renewenvironment*{\the\index}{%
35   \LWR@indexsection{\indexname}%
36   \ifx\index@prologue\empty\else
37     \index@prologue
38     \bigskip
39   \fi
40   \let\item\LWR@indexitem%
41   \let\subitem\LWR@indexsubitem%
42   \let\subsubitem\LWR@indexsubsubitem%
43 }{}%
44 }% AtBeginDocument

```

Disabled:

```
45 \def\@showidx#1{}
46 \let\@texttop\relax
47 \renewcommand*\{\raggedbottom}{}%
48 \renewcommand*\{\flushbottom}{}%
49 \renewcommand*\{\markboth}[2]{}%
50 \renewcommand*\{\markright}[1]{}
```

File 163 **l warp-intopdf.sty**

§ 259 Package **intopdf**

Pkg **intopdf** **intopdf** is emulated.

The MIME type and description are ignored for now.

for HTML output: 1 \LWR@ProvidesPackageDrop{intopdf}

```
2 \NewDocumentCommand{\attachandlink}{m o m m}{%
3   \href{#1}{#4}%
4 }
```

File 164 **l warp-keyfloat.sty**

§ 260 Package **keyfloat**

(Emulates or patches code by BRIAN DUNN.)

Pkg **keyfloat** **keyfloat** is supported with minor adjustments.

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

for HTML output: 1 \LWR@ProvidesPackagePass{keyfloat}

After `keyfloat` has loaded:

```
2 \AtBeginDocument{%
3 \RenewDocumentCommand{\KFLT@onefigureimage}{}{}
```

```
4 {%
5 \LWR@traceinfo{KFLT@onefigureimage}%
6 % \begin{lrbox}{\KFLT@envbox}%
7 \ifthenelse{\NOT\equal{\KFLT@lw}{}}{%
8 {\includegraphics{%
9 [scale=\KFLT@s, width=\KFLT@imagewidth]{\KFLT@i}}}%
10 {%
11 \ifthenelse{\dimtest{\KFLT@w}>}{0pt}{%
12 {%
13 \ifthenelse{\dimtest{\KFLT@h}>}{0pt}{%
14 {%
15 \includegraphics{%
16 [scale=\KFLT@s,%
17 width=\KFLT@imagewidth, height=\KFLT@h]{\KFLT@i}}%
18 }% w and h
19 {%
20 \includegraphics{%
21 [scale=\KFLT@s, width=\KFLT@imagewidth]{\KFLT@i}}%
22 }% only w
23 }% width is given
24 {%
25 \ifthenelse{\dimtest{\KFLT@h}>}{0pt}{%
26 {\includegraphics{%
27 [scale=\KFLT@s, height=\KFLT@h]{\KFLT@i}}}%
28 {\includegraphics{%
29 [scale=\KFLT@s]{\KFLT@i}}%
30 }% width is not given
31 }% not linewidth
32 \% \end{lrbox}%
33 \% \unskip%
34 \% \KFLT@findenvboxwidth%
35 \% \begin{turn}{\KFLT@r}%
36 \% \KFLT@frame{\usebox{\KFLT@envbox}}%
37 \% \unskip%
38 \% \end{turn}%
39 \LWR@traceinfo{KFLT@onefigureimage: done}%
40 }

41 \RenewDocumentEnvironment{KFLT@boxinner}{}%
42 {%
43 \LWR@traceinfo{KFLT@boxinner}%
44 \LWR@stopars%
45 }%
46 {%
47 \LWR@startpars%
48 \LWR@traceinfo{KFLT@boxinner: done}%
49 }

50 \DeclareDocumentEnvironment{KFLT@marginfofloat}{0{-1.2ex} m}
```

```

51 {%
52 \LWR@BlockClassWP{float:right; width:2in; margin:10pt}{\marginblock}%
53 \captionsetup{type=#2}%
54 }%
55 {%
56 \endLWR@BlockClassWP%
57 }

58 \DeclareDocumentEnvironment{marginfigure}{o}
59   {\begin{KFLT@\marginfloat}{figure}}
60   {\end{KFLT@\marginfloat}}
61
62 \DeclareDocumentEnvironment{margintable}{o}
63   {\begin{KFLT@\marginfloat}{table}}
64   {\end{KFLT@\marginfloat}}

65 \DeclareDocumentEnvironment{keywrap}{m +m}
66 {%
67 \LWR@ensuredoingapar%
68 \setlength{\LWR@templengthone}{#1}%
69 \begin{LWR@BlockClassWP}{%
70   float:right; width:\LWR@printlength{\LWR@templengthone}; % extra space
71   margin:10pt}%
72 }%
73 {%
74   width:\LWR@printlength{\LWR@templengthone}%
75 }%
76 \marginblock}%
77 \setlength{\ linewidth}{.95\LWR@templengthone}%
78 #2%
79 \end{LWR@BlockClassWP}%
80 }
81 {%
82 }

83 }% AtBeginDocument

```

File 165 **lwarf-layaureo.sty**

§ 261 Package **layaureo**

Pkg **layaureo** **layaureo** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{layaureo}[2004/09/16]

File 166 **l warp-layout.sty**

§ 262 Package **layout**

(Emulates or patches code by KENT MCPHERSON, JOHANNES BRAAMS, HIDEO UMEKI.)

Pkg **layout** layout is emulated.

for HTML output: Discard all options for l warp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}  
2 \NewDocumentCommand{\layout}{s}{}  
3 \newcommand\leading[1]{}
```

File 167 **l warp-leading.sty**

§ 263 Package **leading**

Pkg **leading** leading is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{leading}[2008/12/11]
2 \newcommand\leading[1]{}

File 168 **l warp-letterspace.sty**

§ 264 Package **letterspace**

(Emulates or patches code by R SCHLICHT.)

Pkg **letterspace** letterspace is a subset of microtype, which is pre-loaded by l warp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for l warp-letterspace:

```
1 \LWR@ProvidesPackageDrop{letterspace}  
2 \newcommand*\lsstyle{}  
3 \newcommand\textls[2][]{}
```

```

4 \def\textls#1#{}%
5 \newcommand*\lslig[1]{#1}

```

File 169 **l warp-lettrine.sty**

§ 265 Package **lettrine**

(Emulates or patches code by DANIEL FLIPO.)

Pkg **lettrine** Emulated.

for HTML output: Discard all options for l warp-lettrine:

```

1 \LWR@ProvidesPackageDrop{lettrine}

```

The initial letter is in a of class **lettrine**, and the following text is in a of class **lettrinetext**. \lettrine [*keys*] {*letter*} {*additional text*}

```

2 \DeclareDocumentCommand{\lettrine}{o m m}{%
3 \InlineClass{lettrine}{#2}\InlineClass{lettrinetext}{#3} % extra space
4 }
5
6 \newcounter{DefaultLines}
7 \setcounter{DefaultLines}{2}
8 \newcounter{DefaultDepth}
9 \newcommand*{\DefaultOptionsFile}{\relax}
10 \newcommand*{\DefaultLoversize}{0}
11 \newcommand*{\DefaultLraise}{0}
12 \newcommand*{\DefaultLhang}{0}
13 \newdimen\DefaultFindent
14 \setlength{\DefaultFindent}{\z@}
15 \newdimen\DefaultNindent
16 \setlength{\DefaultNindent}{0.5em}
17 \newdimen\DefaultSlope
18 \setlength{\DefaultSlope}{\z@}
19 \newdimen\DiscardVskip
20 \setlength{\DiscardVskip}{0.2\p@}
21 \newif\ifLettrineImage
22 \newif\ifLettrineOnGrid
23 \newif\ifLettrineRealHeight
24
25 \newcommand*{\LettrineTextFont}{\scshape}
26 \newcommand*{\LettrineFontHook}{}
27 \newcommand*{\LettrineFont}[1]{\InlineClass{lettrine}{#1}}
28 \newcommand*{\LettrineFontEPS}[1]{\includegraphics[height=1.5ex]{#1}}

```

File 170 **l warp-lineno.sty**

§ 266 Package **lineno**

(Emulates or patches code by STEPHAN I. BÖTTCHER.)

Pkg **lineno** **lineno** is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{lineno}

2 \newcommand*\resetlinenumber[1][\@ne]{}
3
4 \def\linenumbers{%
5     \@ifnextchar[\{\resetlinenumber\}]{%
6         \@ifstar{\resetlinenumber}{\relax}{\relax}{}%
7     }%
8
9 \newcommand*{\nolinenumbers}{}
10
11 \namedef{linenumbers*}{\par\linenumbers*}
12 \namedef{runninglinenumbers*}{\par\runninglinenumbers*}
13
14 \def\endlinenumbers{\par}
15 \let\endrunninglinenumbers\endlinenumbers
16 \let\endpagewiselinenumbers\endlinenumbers
17 \expandafter\let\csname endlinenumbers*\endcsname\endlinenumbers
18 \expandafter\let\csname endrunninglinenumbers*\endcsname\endlinenumbers
19 \let\endnolinenumbers\endlinenumbers
20
21 \def\pagewiselinenumbers{\linenumbers\setpagewiselinenumbers}
22
23 \def\runninglinenumbers{\setrunninglinenumbers\linenumbers}
24
25 \def\setpagewiselinenumbers{}
26
27 \def\setrunninglinenumbers{}
28
29 \def\linenomath{%
30 \namedef{linenomath*}{\relax}%
31 \def\endlinenomath{%
32 \expandafter\let\csname endlinenomath*\endcsname\endlinenomath
33
34 \let\linelabel\label
35
36 \def\switchlinenumbers{\@ifstar{}{}}}
```

```
37 \def\setmakelinenumbers#1{\@ifstar{}{}}
38
39 \def\leftlinenumbers{\@ifstar{}{}}
40 \def\rightlinenumbers{\@ifstar{}{}}
41
42 \newcounter{linenumber}
43 \newcount\c@pagewiselinenumber
44 \let\c@runninglinenumber\c@linenumber
45
46 \def\runningpagewiselinenumbers{}
47 \def\realpagewiselinenumbers{}
48
49
50 \NewDocumentCommand\modulolinenumbers{s o}{}
51
52 \chardef\c@linenumbermodulo=5
53 \modulolinenumbers[1]
54
55 \newcommand*\firstlinenumber[1]{}
56
57 \newcommand\internallinenumbers{}
58 \let\endinternallinenumbers\endlinenumbers
59 \namedef{internallinenumbers*}{\internallinenumbers}
60 \expandafter\let\csname endinternallinenumbers*\endcsname\endlinenumbers
61
62 \newcommand*{\linenoplaceholder}[1]{% redefine per language
63     (line number reference for \detokenize\expandafter{\#1})
64 }
65
66 \newcommand*{\lineref}[2][]{\linenoplaceholder{\#2}}
67 \newcommand*{\linerefP}[2][]{\linenoplaceholder{\#2}}
68 \newcommand*{\linerefR}[2][]{\linenoplaceholder{\#2}}
69
70 \newcommand\quotelinenumbers
71   {\@ifstar\linenumbers{\@ifnextchar[\linenumbers{\linenumbers*}}}
72
73 \newdimen\linenumbersep
74 \newdimen\linenumberwidth
75 \newdimen\quotelinumbersep
76
77 \quotelinumbersep=\linenumbersep
78 \let\quotelinumberfont\linenumberfont
79
80 \def\linenumberfont{\normalfont\tiny\sffamily}
81
82
83 \linenumberwidth=10pt
84 \linenumbersep=10pt
85
86 \def\thelinenumbers{}
```

```

87
88 \def\LineNumber{}
89 \def\makeLineNumber{}
90 \def\makeLineNumberLeft{}
91 \def\makeLineNumberRight{}
92 \def\makeLineNumberOdd{}
93 \def\makeLineNumberEven{}
94 \def\makeLineNumberRunning{}
95
96
97 \newenvironment{numquote} {\begin{quote}}{\end{quote}}
98 \newenvironment{numquotation} {\begin{quotation}}{\end{quotation}}
99 \newenvironment{numquote*} {\begin{quote}}{\end{quote}}
100 \newenvironment{numquotation*} {\begin{quotation}}{\end{quotation}}
101
102 \newdimen\bframerule
103 \bframerule=\fboxrule
104
105 \newdimen\bframesep
106 \bframesep=\fboxsep
107
108 \newenvironment{bframe}{%
109   \LWR@forceminwidth{\bframerule}%
110   \BlockClass[%
111     border:\LWR@printlength{\LWR@atleastonept} solid black ; %
112     padding:\LWR@printlength{\bframesep}%
113   ]{bframe}%
114 }%
115 }%
116 \endBlockClass

```

File 171 **lwarf-lips.sty**

§ 267 Package **lips**

(Emulates or patches code by MATT SWIFT.)

Pkg lips lips is emulated.

```

1 \% \LWR@ProvidesPackageDrop{lips}
2 \PackageInfo{lwarf}{Using the lwarf version of package ‘lips’.}%
3 \ProvidesPackage{lwarf-lips}
4
5 \NewDocumentCommand{\Lips}{}{\textellipsis}
6
7 \NewDocumentCommand{\BracketedLips}{}{[\textellipsis]}
8

```

```

9 \let\lips\Lips
10 \let\olips\lips
11
12 \DeclareOption*{}
13 \DeclareOption{mla}{%
14 \let\lips\BracketedLips
15 }
16 \ProcessOptions\relax
17
18 \newcommand \LPNobreakList {}

```

File 172 **lwarp-listings.sty**

§ 268 Package **listings**

(Emulates or patches code by CARSTEN HEINZ, BROOKS MOSES, JOBST HOFFMANN.)

Pkg **listings** **listings** is supported with some limitations. Text formatting is not yet supported.

for HTML output: 1 \begin{warpHTML}

2 \LWR@ProvidesPackagePass{listings}

Force flexible columns. Fixed columns inserts spaces in the PDF output.

3 \lst@column@flexible

Patches to embed listings inside pre tags:

```

4 \let\LWR@origlst@Init\lst@Init
5 \let\LWR@origlst@DeInit\lst@DeInit
6
7 \let\LWR@origlst@EveryPar\lst@EveryPar
8
9 \renewcommand{\l@lstlisting}[2]{\hypertocfloat{1}{lstlisting}{lol}{#1}{#2}}

```

\lstset {<options>}

Use the **listings** literate option to replace HTML entities:

```

10 \def\lstset@#1{\endgroup%
11 %   \ifx\empty\@empty#1%
12 %     \empty%
13 %   \else%
14 %     \setkeys{lst}{%
15 %       #1%

```

```

16           ,literate=%
17           {<}{\HTMLentity{lt}}{4}%
18           {>}{\HTMLentity{gt}}{4}%
19           {\&}{\HTMLentity{amp}}{5}%
20       }%
21 %     \fi%
22 }

```

\lst@Init {*backslash-processing*} Done at the start of a listing.
 23 \renewcommand{\lst@Init}[1]{%

Perform the listings initialization:

```

24 \LWR@traceinfo{\lst@Init}%
25 \renewcommand*{\@captype}{lstlisting}%
26 \let\lst@aboveskip\z@\let\lst@beforeskip\z@%
27 \gdef\lst@boxpos{t}%
28 \let\lst@frame\@empty
29   \let\lst@frametshape\@empty
30   \let\lst@framershape\@empty
31   \let\lst@framebshape\@empty
32   \let\lst@framefshape\@empty
33 \lstframe@\lst@frameround ffff\relax%
34 \lst@multicols\@empty%
35 \LWR@origlst@Init{\#1}\relax%

```

Avoids extra horizontal space:

```

36 \def\lst@framelr{}%
37 \LWR@traceinfo{finished origlst@Init}%
38 \lst@ifdisplaystyle%

```

Creating a display.

Disable line numbers, produce the <pre>, then reenable line numbers.

```

39 \LWR@traceinfo{About to create verbatim.}%
40 \let\lsthk@EveryPar\relax%
41 \LWR@forcenewpage
42 \LWR@atbeginverbatim{2}{programlisting}%
43
44 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
45 \else%

```

Inline, so open a :

```

46 \ifbool{\LWR@verbtags}{\LWR@htmlltag{span class="inlineprogramlisting"}{}}{%
47 \fi%
48 }%
49

```

\lst@DeInit Done at the end of a listing.

```
50 \renewcommand*\lst@DeInit{%
51 \lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the </pre>, then reenable line numbers:

```
52 \let\lsthk@EveryPar\relax%
53 \LWR@afterendverbatim{0}%
54 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
55 \else%
```

Inline, so create the closing :

```
56 \ifbool{\LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
57 \fi%
```

Final listings deinit:

```
58 \LWR@origlst@DeInit%
59 }
```

\lst@MakeCaption {*t/b*}

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```
60 \def\lst@MakeCaption#1{%
61 \LWR@traceinfo{MAKING CAPTION at #1}%
62 \lst@ifdisplaystyle
63 \LWR@traceinfo{making a listings display caption}%
64 \ifx #1%
65 \ifx\lst@@caption\empty\expandafter\lst@HRefStepCounter \else
66 \expandafter\refstepcounter
67 \fi {lstlisting}%
68 \LWR@traceinfo{About to assign label: !\lst@label!}%
69 %
70 \label{\lst@label}\fi
71 \LWR@traceinfo{Finished assigning the label.}%
72 \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
73 \global\let\lst@name\lst@arg \global\let\lstname\lst@name
74 \lst@ifnolol\else
75 \ifx\lst@@caption\empty
76 \ifx\lst@caption\empty
77 \ifx\lst@intname\empty \else \def\lst@temp{ }%
78 \ifx\lst@intname\lst@temp \else
```

This code places a contents entry for a non-float. This would have to be modified for lwarp:

```
79 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
80 %
\addcontentsline{lol}{lstlisting}{\lst@name}
```

```

81          \fi\fi
82          \fi
83      \else

```

This would have to be modified for lwarp:

```

84 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
85             \addcontentsline{loll}{lstlisting}%
86 {\protect\numberline{\the\lstlisting}%
87 {\protect\ignorespaces \LWR@isolate{\lst@@caption} \protect\relax}}%
88         \fi
89         \fi
90     \fi
91     \ifx\lst@caption\empty\else
92 \LWR@traceinfo{lst@caption not empty-}%
93     \lst@ifSubstring #1\lst@captionpos
94         {\begingroup
95 \LWR@traceinfo{at the selected position}%

```

These space and box commands are not needed for HTML output:

```

96 %          \let\@vskip\vskip
97 %          \def\vskip{\afterassignment\lst@vskip \tempskipa}%
98 %          \def\lst@vskip{\nobreak\@vskip\tempskipa\nobreak}%
99 %          \par\@parboxrestore\normalsize\normalfont \% \noindent (AS)
100 %          \ifx #1\allowbreak \fi
101         \ifx\lst@title\empty

```

New lwarp code to create a caption:

```

102         \lst@makecaption\fnam@lstlisting{\ignorespaces \lst@caption}%
103     \else

```

New lwarp code to create a title:

```

104 %          \lst@maketitle\lst@title \% (AS)
105 \LWR@traceinfo{Making title: \lst@title}%
106 \begin{BlockClass}{lstlistingtitle}% lwarp
107 \lst@maketitle\lst@title% lwarp
108 \end{BlockClass}% lwarp
109         \fi
110 \LWR@traceinfo{About to assign label: !\lst@label!}%
111     \ifx\lst@label\empty\else
112 \leavevmode% gets rid of bad space factor error
113 \GetTitleStringExpand{\lst@caption}%
114 \edef\LWR@lntemp{\GetTitleStringResult}%
115 \edef@\currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
116 \label{\lst@label}\fi
117 \LWR@traceinfo{Finished assigning the label.}%

```

Not needed for lwarp:

```

118 %          \ifx #1b\allowbreak \fi
119         \endgroup}{}%

```

```

120      \fi
121 \LWR@traceinfo{end of making a listings display caption}%
122 \else
123 \LWR@traceinfo{INLINE}%
124 \fi
125 \LWR@traceinfo{DONE WITH CAPTION at #1}%
126 }

```

- line numbers** Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```

127 \lst@Key{numbers}{none}%
128   \let\lst@PlaceNumber\empty
129   \lstKV@SwitchCases{#1}%
130   {none&\%
131     left\def\lst@PlaceNumber\%

```

For now, lwarp places left line numbers inline. Ideally the entire line would be moved to the right, but conflicts with list indenting occurs.

```

132 %          \LWR@origllap{%
133             \LWR@orignormalfont%
134             \lst@numberstyle{\the\lstnumber}\kern\lst@numbersep%
135 %           }
136 }\\%
137   right\def\lst@PlaceNumber{\LWR@origrlap{\LWR@orignormalfont
138             \kern 6in \kern\lst@numbersep
139             \lst@numberstyle{\the\lstnumber}}\%
140 }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}

```

```
141 \end{warpHTML}
```

File 173 **lwarp-lltjext.sty**

§ 269 Package **lltjext**

(Emulates or patches code by THE LUATEX-JA PROJECT TEAM.)

Pkg lltjext lltjext is patched for use by lwarp.

for HTML output: 1 \LWR@ProvidesPackagePass{lltjext}

```

2 \protected\def\yoko{%
3   \directlua{luatexja.direction.set_list_direction(4, 'yoko')}%
4 }
5 \protected\def\tate{\yoko}

```

```

6 \protected\def\dtou{\yoko}
7 \protected\def\utod{\yoko}
8
9 \define@key[ltj]{japaram}{direction}{}
10
11 \yoko
12
13 \DeclareExpandableDocumentCommand{\rensushi}{s o m}{#3}
14
15 \DeclareDocumentCommand{\layoutfloat}{d() o m}{}
16
17 \DeclareDocumentCommand{\DeclareLayoutCaption}{m d<> d() o}{}
18
19 \LetLtxMacro\pcaption\caption
20
21 \DeclareDocumentCommand{\layoutcaption}{d<> d() o}{}
22
23 \let\captiondir\relax
24 \RenewDocumentEnvironment{LWR@HTML@minipage}{d<> 0{t} 0{} 0{t} m}
25     {\LWR@HTML@sub@minipage{#2}{#3}{#4}{#5}}
26     {\endLWR@HTML@sub@minipage}
27
28 \RenewDocumentCommand{\LWR@HTML@parbox}{d<> 0{t} 0{} 0{t} m +m}
29 {
30 \LWR@traceinfo{parbox of width #4}%
31 \begin{minipage}[#2][#3][#4]{#5}%
32 #6
33 \end{minipage}%
34 }
35
36 \RenewDocumentCommand{\pbox}{d<> 0{0pt} 0{c} m}{%
37 \booltrue{LWR@minipagewidth}%
38 \parbox{#2}{#4}%
39 }

```

File 174 **lwarf-longtable.sty**

§ 270 Package **longtable**

(Emulates or patches code by DAVID CARLISLE.)

Pkg **longtable** longtable is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{longtable}

⚠ Misplaced **\noalign**
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` `\kill` is ignored, place a `\kill` line inside

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

or place it inside `\warpingprintonly`.

⚠ `\lateximage` `\longtable` is not supported inside a `\lateximage`.

See:

<http://tex.stackexchange.com/questions/43006/why-is-input-not-expandable>

Env `\longtable` * [*horizontalignment*] {*colspec*} Emulates the `\longtable` environment.

Per the `caption` package, the starred version steps the counter per caption. The unstarred version steps the counter once at the beginning, but not at each caption.

Options [c], [l], and [r] are thrown away.

```

2 \newenvironment{longtable*}[2] []{%
3 \LWR@floatbegin{table}%
4 \setcaptiontype{\LTcaptype}%
5 \caption@setoptions{longtable}%
6 \caption@setoptions{@longtable}%
7 \caption@LT@setup%
8 \booltrue{\LWR@starredlongtable}%
9 \let\captionlistentry\LWR@LTcaptionlistentry%
10 \tabular{#2}%
11 }%
12 {\endtabular\LWR@floatend}%
13
14 \newenvironment{longtable}[2] []{%
15 \LWR@floatbegin{table}%
16 \setcaptiontype{\LTcaptype}%
17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{\LTcaptype}%
21 \let\captionlistentry\LWR@LTcaptionlistentry%
22 \tabular{#2}%

```

```
23 }
24 {\endtabular\LWR@floatend}
```

Provided for compatibility, but ignored:

```
25 \newcounter{LTchunksize}
26 \def\endhead{\LWR@tabularendofline}%
27 \def\endfirsthead{\LWR@tabularendofline}
28 \def\endfoot{\LWR@tabularendofline}
29 \def\endlastfoot{\LWR@tabularendofline}
30 \newcommand\tabularnewline{\LWR@tabularendofline}
31 \newcommand{\setlongtables}{}% Obsolete command, does nothing.
32 \newlength{\LTleft}
33 \newlength{\LTright}
34 \newlength{\LTpre}
35 \newlength{\LTpost}
36 \newlength{\LTcapwidth}

37 \LetLtxMacro{\LWR@origkill}{\kill}
38 \renewcommand*{\kill}{\LWR@tabularendofline}
39 \appto{\LWR@restoreorigformatting}{%
40 \LetLtxMacro{\kill}{\LWR@origkill}%
41 }
```

File 175 **l warp-lscape.sty**

§ 271 Package **lscape**

(Emulates or patches code by D. P. CARLISLE.)

Pkg **lscape** lscape is emulated.

for HTML output: Discard all options for l warp-lscape.

```
1 \LWR@ProvidesPackageDrop{lscape}
2 \newenvironment*{landscape}{}{}
```

File 176 **l warp-ltablex.sty**

§ 272 Package **ltablex**

(Emulates or patches code by ANIL K. GOEL.)

Pkg ltablex ltablex is emulated by l warp.

for HTML output: Relies on tabularx.

```

1 \RequirePackage{tabularx}
2
3 \LWR@ProvidesPackageDrop{ltablex}
4
5 \DeclareDocumentEnvironment{tabularx}{m o m}
6 {\longtable{#3}}
7 {\endlongtable}
8
9 \DeclareDocumentEnvironment{tabularx*}{m o m}
10 {\longtable{#3}}
11 {\endlongtable}
12
13 \newcommand*{\keepXColumns}(){}
14 \newcommand*{\convertXColumns}{}{}
```

File 177 l warp-ltcaption.sty

§ 273 Package **ltcaption**

(Emulates or patches code by AXEL SOMMERFELDT.)

Pkg ltcaption ltcaption is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{ltcaption}

\LTcaptype is already defined by l warp.

longtable* is already defined by l warp-longtable.

```

2 \newlength{\LTcapskip}
3 \newlength{\LTcapleft}
4 \newlength{\LTcapright}
5 \newcommand*{\LTcapmarginsfalse}{}{}
```

File 178 l warp-ltxgrid.sty

§ 274 Package **ltxgrid**

Pkg ltxgrid ltxgrid is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{ltxgrid}

```
2 \newcommand*{\onecolumngrid}{}  
3 \newcommand*{\twocolumngrid}{}  
4 \newcommand*{\removestuff}{}  
5 \newcommand*{\addstuff}[2]{}  
6 \newcommand*{\replacestuff}[2]{}  


---


```

File 179 **l warp-ltxtable.sty**

§ 275 Package **ltxtable**

Pkg ltxtable ltxtable is emulated.

⚠ **table numbering** The print version does not seem to honor `longtable*` from the caption package, while l warp does.

for HTML output: 1 \LWR@ProvidesPackageDrop{ltxtable}

```
\LTXtable {\langle width\rangle} {\langle file\rangle}  
2 \newcommand*{\LTXtable}[2]{%  
3 \input{#2}%  
4 }  


---


```

File 180 **l warp-lua-check-hyphen.sty**

§ 276 Package **lua-check-hyphen**

Pkg lua-check-hyphen lua-check-hyphen is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{lua-check-hyphen}

```
2 \newcommand*{\LuaCheckHyphen}[1]{}  


---


```

File 181 **l warp-lua-visual-debug.sty**

§ 277 Package **lua-visual-debug**

Pkg lua-visual-debug lua-visual-debug is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{lua-visual-debug}[2016/05/30]

File 182 **l warp-luacolor.sty**

§ 278 Package **luacolor**

Pkg luacolor luacolor is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{luacolor}
2 \newcommand{\luacolorProcessBox}[1]{}
```

File 183 **l warp-luatodonotes.sty**

§ 279 Package **luatodonotes**

(Emulates or patches code by FABIAN LIPP.)

Pkg luatodonotes luatodonotes is emulated.

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

for HTML output:

```
1 \LWR@ProvidesPackagePass{luatodonotes}
```

Nullify options:

```
2 \et@todonotes@additionalMarginEnabledfalse
3 \if@todonotes@disabled
4 \else
5
6 \newcommand{\ext@todo}{\todo}
7
8 \renewcommand{\l@todo}[2]{\hypertocfloat{1}{\todo}{\ldo}{#1}{#2}}
9 \let\LWRTODONOTES@orig@todototoc\todototoc
10
11 \renewcommand*{\todototoc}{%
12 \phantomsection%
13 \LWRTODONOTES@orig@todototoc%
14 }
```

```
15
16
17 \renewcommand{\@todonotes@drawMarginNoteWithLine}{%
18 \fcolorbox
19   {\@todonotes@currentbordercolor}
20   {\@todonotes@currentbackgroundcolor}
21   {\arabic{@todonotes@numberoftodonotes}}%
22 \marginpar{\@todonotes@drawMarginNote}
23 }
24
25 \renewcommand{\@todonotes@drawInlineNote}{%
26 \fcolorboxBlock%
27   {\@todonotes@currentbordercolor}%
28   {\@todonotes@currentbackgroundcolor}%
29   {%
30     \if@todonotes@authorgiven%
31       {\@todonotes@author:\,}%
32     \fi%
33     \@todonotes@text%
34   }%
35 }
36
37 \newcommand{\@todonotes@drawMarginNote}{%
38   \if@todonotes@authorgiven%
39     \@todonotes@author\par%
40   \fi%
41   \arabic{@todonotes@numberoftodonotes}: %
42   \fcolorbox{%
43     {\@todonotes@currentbordercolor}%
44     {\@todonotes@currentbackgroundcolor}%
45     {%
46       \@todonotes@sizecommand%
47       \@todonotes@text %
48     }%
49 }%
50
51 \renewcommand{\missingfigure}[2][]{%
52 \setkeys{todonotes}{#1}%
53 \addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
54 \fcolorboxBlock%
55   {\@todonotes@currentbordercolor}%
56   {\@todonotes@currentfigcolor}%
57   {%
58     \setlength{\fboxrule}{4pt}%
59     \fcolorbox{red}{white}{Missing figure} \quad #2%
60   }%
61 }
62
63 \LetLtxMacro{\LWRTODONOTES@orig}{\todocommon}%
64 \LetLtxMacro{\LWRTODONOTES@orig}{\todocommon}
```

```

65 \RenewDocumentCommand{\@todocommon}{m m}{%
66 \begingroup%
67 \renewcommand*\phantomsection{}%
68 \LWRTODONOTES@orig@todocommon{#1}{#2}%
69 \endgroup%
70 }%
71
72 \renewcommand{\@todoarea}[3][]{%
73     \todonotes@areaselectedtrue%
74     \todocommon{#1}{#2}%
75     \todonotes@textmark@highlight{#3}%
76     \zref@label!\todonotes@\arabic{\todonotes@numberoftodonotes}@end}%
77 }%
78
79
80 \DeclareDocumentCommand{\todonotes@textmark@highlight}{m}{%
81 \InlineClass[background:\LWR@origpound{}B3FFB3]{highlight}{#1}%
82 }%
83
84 \fi% \if\todonotes@disabled

```

File 184 **lwarf-magaz.sty**

§ 280 Package **magaz**

Pkg magaz magaz is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{magaz}

2 \newcommand\FirstLine[1]{%
3     \begingroup%
4     \FirstLineFont{%
5         \LWR@textcurrentcolor{%
6             \LWR@textcurrentfont{%
7                 #1%
8             }%
9         }%
10    }%
11    \endgroup%
12 }%
13
14 \providecommand\FirstLineFont{\scshape}

```

File 185 l warp-makeidx.sty

§ 281 Package **makeidx**

(Emulates or patches code by LATEX PROJECT TEAM.)

Pkg makeidx makeidx is patched for use by l warp.

for HTML output: 1 \LWR@ProvidesPackagePass{makeidx}

\@wrindex is redefined \AtBeginDocument by the l warp core.

\printindex

```
2 \preto\printindex{%
3   \LWR@orignewpage%
4   \LWR@startpars%
5 }
```

File 186 l warp-marginal.sty

§ 282 Package **marginal**

Pkg marginal marginal is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{marginal}

```
2 \newcommand*{\showlostmarginals}(){}
3 \newcommand*{\enlargefreelist}){}
4 \newcommand*{\onesidemarginals}{}{}
```

File 187 l warp-marginfit.sty

§ 283 Package **marginfit**

Pkg marginfit marginfit is ignored.

for HTML output:

Discard all options for l warp-marginfit:

```
1 \LWR@ProvidesPackageDrop{marginfit}
```

File 188 **l warp-marginfix.sty**

§ 284 Package **marginfix**

(Emulates or patches code by STEPHEN HICKS.)

Pkg **marginfix** Emulated.

for HTML output: Discard all options for l warp-marginfix:

```
1 \LWR@ProvidesPackageDrop{marginfix}

2 \newcommand*{\marginskip}[1]{}
3 \newcommand*{\clearmargin}{}
4 \newcommand*{\softclearmargin}{}
5 \newcommand*{\extendmargin}[1]{}
6 \newcommand*{\mparshift}[1]{}
7 \newdimen\marginheightadjustment
8 \newdimen\marginposadjustment
9 \newcommand*{\blockmargin}[1][]{}
10 \newcommand*{\unblockmargin}[1][]{}
11 \newcommand*{\marginphantom}[2][]{}
```

File 189 **l warp-marginnote.sty**

§ 285 Package **marginnote**

(Emulates or patches code by MARKUS KOHM.)

Pkg **marginnote** Emulated.

for HTML output: Discard all options for l warp-marginnote:

```
1 \LWR@ProvidesPackageDrop{marginnote}

2 \NewDocumentCommand{\marginnote}{+o +m o}{\marginpar{#2}{}}

3 \newcommand*{\marginnoteleftadjust}{}
4 \newcommand*{\marginnoterightadjust}{}
5 \newcommand*{\marginnotetextwidth}{}
```

```

6 \let\marginnotetextwidth\textwidth
7 \newcommand*\{\marginnoteadjust}{}
8 \newcommand*\{\marginfont}{}
9 \newcommand*\{\raggedleftmarginnote}{}
10 \newcommand*\{\raggedrightmarginnote}{}

```

File 190 **lwarf-mcaption.sty**

§ 286 Package **mcaption**

(Emulates or patches code by STEPHAN HENNIG.)

Pkg **mcaption** **mcaption** is nullified.

for HTML output: Discard all options for lwarf-mcaption:

```

1 \LWR@ProvidesPackageDrop{mcaption}

2 \newenvironment{margincap}{}{}
3 \newcommand*\{\margincapalign}{}
4 \newlength{\margincapsep}

```

File 191 **lwarf-mdframed.sty**

§ 287 Package **mdframed**

(Emulates or patches code by MARCO DANIEL, ELKE SCHUBERT.)

Pkg **mdframed** **mdframed** is loaded with options forced to **framemethod=none**.

§ 287.1 **Limitations**

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

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

font For title font, use

frametitlefont=\textbf,

instead of

```
frametitlefont=\bfseries,
```

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

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

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

css classes Environments created or encapsulated by `mdframed` are enclosed in a `<div>` of class `mdframed`, and also class `md<environmentname>` for new environments.

Frame titles are placed in a `<div>` of class `|mdframedtitle|`. Subtitles are in a `<div>` of class `|mdframedsubtitle|`, and likewise for subsubtitles.

Pre-existing hooks are used to patch extra functions before and after the frames.

§ 287.2 Package loading

for HTML output:

```
1 \RequirePackage{xcolor}%
2 %
3 \LWR@ProvidesPackageDrop{mdframed}
```

Do not require `Tikz` or `pstricks`:

```
4 \LWR@origRequirePackage[framemethod=none]{mdframed}
```

§ 287.3 Patches

Patch to remove PDF formatting and add HTML tags:

```
5 \AtBeginDocument{
6 \def\mdf@trivlist#1{%
7 \edef\mdf@temp{%
8 % \topsep=\the\topsep\relax%
9 % \partopsep=\the\partopsep\relax%
10 % \parsep=\the\parsep\relax%
11 }%
12 % \setlength{\topsep}{#1}%
13 % \topskip\z@%
14 % \partopsep\z@%
15 % \parsep\z@%
16 % \c@mmbrlistfalse%
17 % \c@trivlist%
18 % \labelwidth\z@%
19 % \leftmargin\z@%
```

```

20 \%   \itemindent\z@%
21  \let\@itemlabel\@empty%
22  \def\makelabel##1{##1}%
23 \%   \item\relax\mdf@temp\relax%
24 }
25
26 \renewcommand*{\endmdf@trivlist}{%
27 \LWR@traceinfo{\endmdf@trivlist}%
28 \% \endtrivlist%
29 \LWR@listend%
30 }
31 }% AtBeginDocument

```

§ 287.4 Initial setup

To handle css and paragraphs, patch code at start and end of environment and contents. \LWR@print@raggedright helps avoid hyphenation.

```

32 \mdfsetup{
33 startcode={\LWR@mdframedstart\LWR@print@raggedright},
34 endcode={\LWR@mdframedend},
35 startinnercode={\LWR@startpars\LWR@print@raggedright},
36 endinnercode={\LWR@stoppars},
37 }

```

§ 287.5 Color and length HTML conversion

\LWR@mdfprintcolor {*mdfcolorkey*}

Given the mdframed key, print the color.

```

38 \newcommand*{\LWR@mdfprintcolor}[1]{%
39 \convertcolorspec{named}{\@nameuse{mdf@##1}}{HTML}\LWR@tempcolor%
40 \LWR@origpound\LWR@tempcolor
41 }

```

\LWR@mdfprintlength {*mdflengthkey*}

Given the mdframed key, print the length.

```

42 \newcommand*{\LWR@mdfprintlength}[1]{%
43 \LWR@forceminwidth{\@nameuse{mdf@##1@length}}%
44 \LWR@printlengthf\LWR@atleastonept}%
45 }

```

§ 287.6 Environment encapsulation

\LWR@mdframedstart Actions before an mdframe starts.

Encapsulate a frame inside a <div> of the desired class.

```
46 \newcommand*{\LWR@mdframedstart}{%
47 \LWR@traceinfo{\LWR@mdframedstart start}%

```

Turn off paragraph handling during the generation of the encapsulating tags:

```
48 \LWR@stoppars%
```

Open a <div> and with custom `class` and custom `style`. A `BlockClass` environment is not used because this <div> is created by the `mdframed` `startcode` and `endcode` settings, which do not properly nest the <div> inside the `mdframed` environment.

```
49 \LWR@htmlltagc{div class="%"
50 mdframed%
51 \ifdefstring{\LWR@mdthisenv}{mdframed}{}{\LWR@mdthisenv}%
52 " \LWR@orignewline
53 style=" \LWR@orignewline
```

Convert and print the background color:

```
54 background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```
55 border: \LWR@mdfprintlength{linewidth} solid
56 \LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
57 border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
58 \ifbool{mdf@shadow}{%
59   box-shadow:
60   \LWR@mdfprintlength{shadowsize}
61   \LWR@mdfprintlength{shadowsize}
62   \LWR@mdfprintlength{shadowsize}
63   \LWR@mdfprintcolor{shadowcolor} ;
64 }
65 {box-shadow: none ;}
66 \LWR@orignewline
67 "}
68 % \LWR@htmldivclass{\LWR@mdthisenv}
```

`mdframed` environment may not work with the `HTML` versions of the following, so restore them to their originals while inside `mdframed`:

```
69 \LWR@select@print@hspace%
70 \renewcommand*{\rule}{\LWR@print@rule}
71 \LetLtxMacro\makebox\LWR@print@makebox%
```

```

72 \LWR@startpars%
73 \LWR@traceinfo{LWR@mdframedstart done}%
74 }

```

\LWR@mdframedend Actions after an mdframe ends.

After closing the <div>, globally restore to the default environment type:

```

75 \newcommand*{\LWR@mdframedend}{%
76 \LWR@traceinfo{LWR@mdframedend start}%

```

Close the custom <div>:

```
77 \LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
78 \gdef\LWR@mdthisenv{mdframed}
```

Resume paragraph handling:

```

79 \LWR@startpars%
80 \LWR@traceinfo{LWR@mdframedend done}%
81 }

```

§ 287.7 Mdframed environment

```

82 \renewenvironment{mdframed}[1] []{%
83   \color@begingroup%
84   \mdfsetup{userdefinedwidth=\linewidth,#1}%
85   \mdf@startcode%
86   \mdf@preenvsetting%
87   \ifdefempty{\mdf@firstframetitle}{}{%
88     {\let\mdf@frametitlesave\mdf@frametitle%
89      \let\mdf@frametitle\mdf@firstframetitle%}
89   }%
90   \ifvmode\nointerlineskip\fi%
91   \ifdefempty{\mdf@frametitle}{}{%
92     {\mdfframedtitleenv{\mdf@frametitle}%
93      \mdf@frametitle@use%}
94   }%
95   \mdf@trivlist{\mdf@skipabove@length}%%
96   \mdf@settings%
97   \mdf@clrbox{\mdf@splitbox@one}%
98 %   \mdf@startinnercode%
99 %   \mdf@startinnercode%
100 }%
101 {%
102 %   \mdf@ignorelastdescenders%
103   \par%
104 %   \unskip\ifvmode\nointerlineskip\hrule \z@\width\hspace\fi%%
105   \ifmdf@footnoteinside%
106     \def\mdf@reservedat{%
107       \mdf@footnoteoutput%

```

```

108 %           \mdf@endinnercode%
109 %           \endmdf@lrbox%
110 %           \ifdefempty{\mdf@frametitle}{}%
111 %               {\mdfframedtitleenv{\mdf@frametitle}\mdf@@frametitle@use}%
112 %               \detected@mdf@put@frame
113     }%
114   \else%
115     \def\mdf@reservedaf{%
116       \mdf@endinnercode%
117       \endmdf@lrbox%
118       \ifdefempty{\mdf@frametitle}{}%
119           {\mdfframedtitleenv{\mdf@frametitle}\mdf@@frametitle@use}%
120           \detected@mdf@put@frame%
121         \mdf@footnoteoutput%
122     }%
123   \fi%
124   \mdf@reservedaf%
125   \aftergroup\endmdf@trivlist%
126 \color@endgroup%
127 \mdf@endcode%
128 }

\mdf@footnoteoutput
129 \renewrobustcmd*\mdf@footnoteoutput{%
130   \LWR@printpendingmpfootnotes%
131 }

```

§ 287.8 Titles and subtitles

\mdfframedtitleenv {*title*}

Place the title inside a <div> of class mdframedtitle:

```

132 \newlength{\LWR@titleroundcorner}
133
134 \renewrobustcmd\mdfframedtitleenv[1]{%
135 \LWR@traceinfo{\LWR@mdframedtitleenv start}%

```

Open a <div> with a custom class and custom style:

```
136 \begin{BlockClass}[%
```

Convert and print the title background color:

```

137 background:
138 \LWR@mdfprintcolor{frametitlebackgroundcolor}
139 ; \LWR@newline

```

Convert and print the title rule:

```

140 \ifbool{\mdf@frametitlerule}{%
141   border-bottom:

```

```

142     \LWR@mdfprintlength{frametitlerulewidth}
143     solid
144     \LWR@mdfprintcolor{frametitlerulecolor}
145     ; \LWR@orignewline
146 }{}%

```

Finish the custom style and the opening <div> tag:

```
147 ]{\mdframedtitle}%
```

Print the title inside the <div>:

```
148 \mdf@frametitlefont{\LWR@textcurrentfont{\#1}}%
```

Close the <div>:

```

149 \end{BlockClass}%
150 \LWR@traceinfo{\LWR@mdframedtitleenv end}%
151 }

```

\LWR@mdfsubtitlecommon {\sub -or- subsub} [\options] {\title}

Common code for \LWR@mdfsubtitle and \LWR@mdfsubsubtitle.

Encapsulate the subtitle inside a <div> of class mdframedsubtitle:

```

152 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
153 {%
154   the following empty line is required
155   \LWR@traceinfo{\LWR@mdframedsubtitlecommon start}%
}

```

Open a <div> with a custom class and custom style:

```
156 \begin{BlockClass}[]
```

Convert and print the background color:

```

157 background:
158 \LWR@mdfprintcolor{\#1titlebackgroundcolor}
159 ; \LWR@orignewline

```

Convert and print the above line:

```

160 \ifbooleq{\mdf@#1titleaboveline}{%
161   border-top:
162   \LWR@mdfprintlength{\#1titleabovelinewidth}
163   solid
164   \LWR@mdfprintcolor{\#1titleabovelinecolor}
165   ; \LWR@orignewline
166 }{}%

```

Convert and print the below line:

```

167 \ifbooleq{\mdf@#1titlebelowline}{%
168   border-bottom:
169   \LWR@mdfprintlength{\#1titlebelowlinewidth}
170   solid
}

```

```

171      \LWR@mdfprintcolor{#1titlebelowlinecolor}
172      ; \LWR@orignewline
173 }{}%

```

Finish the custom style and the opening <div> tag:

```
174 ]{\mdframed#1title}%
```

Perform the original subtitle action:

```

175 \IfNoValueTF{#2}
176 { \nameuse{\LWR@origmdf#1title}{\csuse{mdf@#1titlefont}{\LWR@textcurrentfont{#3}}}}%
177 { \nameuse{\LWR@origmdf#1title}{[#2]{\csuse{mdf@#1titlefont}{\LWR@textcurrentfont{#3}}}}}}%

```

Close the <div>:

```

178 \end{BlockClass}%
179 \LWR@traceinfo{\LWR@mdframedsubtitlecommon end}%
180 }

```

```

\LWR@mdfsubtitle [⟨options⟩] {⟨title⟩}
181 \newcommand*{\LWR@mdfsubtitle}{%
182 \LWR@mdfsubtitlecommon{sub}%
183 }%
184 \let\mdfsubtitle\LWR@mdfsubtitle

```

```

\LWR@mdfsubsubtitle [⟨options⟩] {⟨title⟩}
185 \newcommand*{\LWR@mdfsubsubtitle}{%
186 \LWR@mdfsubtitlecommon{subsub}%
187 }%
188 \let\mdfsubsubtitle\LWR@mdfsubsubtitle

```

§ 287.9 New environments

\LWR@mdthisenv Stores the environment of the frame about to be created:

```
189 \newcommand*{\LWR@mdthisenv}{mdframed}
```

```
\newmdenv [⟨options⟩] {⟨env-name⟩}
```

Modified from the original to remember the environment.

```

190 \renewrobustcmd*\newmdenv[2] [] {%
191 \newenvironment{#2}%
192 {%
193 \mdfsetup{#1}%
194 \renewcommand*{\LWR@mdthisenv}{md#2}%
195 \begin{mdframed}%
196 }%
197 {\end{mdframed}}%
198 }

```

\surroundwithmdframed [*<options>*] [*<environment>*]

Modified from the original to remember the environment.

```
199 \renewrobustcmd*\{\surroundwithmdframed}[2] [] {%
200 \BeforeBeginEnvironment{#2}{%
201 \renewcommand*{\LWR@mdthisenv}{\md#2}%
202 \begin{mdframed}[#1]}%
203 \AfterEndEnvironment{#2}{\end{mdframed}}%
204 }
```

\mdtheorem [*mdframed-options*] [*<envname>*] [*<numberedlike>*] [*<caption>*] [*<within>*]

Modified from the original to remember the environment.

```
205 \DeclareDocumentCommand{\mdtheorem}{ O{} m o m o }{%
206 {\ifcsdef{#2}{%
207 {\LWR@PackageWarning{Environment #2 already exists\MessageBreak}}%
208 {%
209 \IfNoValueTF{#3}{%
210 {#3 not given -- number relationship}%
211 \IfNoValueTF{#5}{%
212 {#3+#5 not given}%
213 \definecounter{#2}%
214 \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
215 \newenvironment{#2}[1] [] {%
216 \refstepcounter{#2}%
217 \ifstrempty{##1}{%
218 {\let\@temptitle\relax}%
219 {%
220 \def\@temptitle{\LWR@theoremseparator}%
221 \LWR@thespace%
222 \LWR@titlefont% l warp
223 \LWR@textcurrentfont{##1}%
224 \mdf@thm@caption{#2}{\#4}{\csname the#2\endcsname{##1}}%
225 }%
226 \begin{mdframed}[#1,frametitle={\strut\#4\ \csname the#2\endcsname{##1}}%
227 {\@temptitle}]%
228 {\end{mdframed}}%
229 \newenvironment{#2*}[1] [] {%
230 \ifstrempty{##1}{\let\@temptitle\relax{\def\@temptitle{: \ ##1}}%
231 \begin{mdframed}[#1,frametitle={\strut\#4\@temptitle}]%
232 {\end{mdframed}}%
233 }%
234 {##5 given -- reset counter}%
235 \definecounter{#2}\@newctr{#2}{#5}%
236 \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
237 \expandafter\xdef\csname the#2\endcsname{%
238 \expandafter\noexpand\csname the#5\endcsname \@thmcountersep\%
239 \@thmcounter{#2}}%
```

```
240     \newenvironment{#2}[1] [] {%
241         \refstepcounter{#2}%
242         \ifstrempty{##1}%
243             {\let\@temptitle\relax}%
244             {%
245                 \def\@temptitle{\mdf@theoremseparator%
246                             \mdf@theoremspace%
247                             \mdf@theoremtitlefont%
248                             \LWR@textcurrentfont{##1}}% l warp
249                 \mdf@thm@caption{#2}{##4}{\csname the#2\endcsname}{##1}}%
250             }
251         \begin{mdframed} [#1,frametitle={\strut#4\ \csname the#2\endcsname%
252                                         \@temptitle}]]}%
253             {\end{mdframed}}%
254         \newenvironment{#2*}[1] [] {%
255             \ifstrempty{##1}%
256                 {\let\@temptitle\relax}%
257                 {%
258                     \def\@temptitle{\mdf@theoremseparator%
259                             \mdf@theoremspace%
260                             \mdf@theoremtitlefont%
261                             \LWR@textcurrentfont{##1}}% l warp
262                     \mdf@thm@caption{#2}{##4}{\csname the#2\endcsname}{##1}}%
263                 }
264             \begin{mdframed} [#1,frametitle={\strut#4\@temptitle}]]}%
265             {\end{mdframed}}%
266         }%
267     }%
268     {##3 given -- number relationship
269     \global\@namedef{the#2}{\@nameuse{the#3}}%
270     \newenvironment{#2}[1] [] {%
271         \refstepcounter{#3}%
272         \ifstrempty{##1}%
273             {\let\@temptitle\relax}%
274             {%
275                 \def\@temptitle{\mdf@theoremseparator%
276                             \mdf@theoremspace%
277                             \mdf@theoremtitlefont%
278                             \LWR@textcurrentfont{##1}}% l warp
279                 \mdf@thm@caption{#2}{##4}{\csname the#2\endcsname}{##1}}%
280             }
281         \begin{mdframed} [#1,frametitle={\strut#4\ \csname the#2\endcsname%
282                                         \@temptitle}]]}%
283             {\end{mdframed}}%
284         \newenvironment{#2*}[1] [] {%
285             \ifstrempty{##1}{\let\@temptitle\relax}{\def\@temptitle{: ##1}}%
286             \begin{mdframed} [#1,frametitle={\strut#4\@temptitle}]]}%
287             {\end{mdframed}}%
288         }%
289     \BeforeBeginEnvironment{#2}{\renewcommand*{\LWR@mdthisenv}{md#2}}% l warp
```

```

290     \BeforeBeginEnvironment{#2*}{\renewcommand*\LWR@mdthisenv}{md#2}}% l warp
291   }%
292 }

\newmdtheoremenv [⟨1: mdframed-options⟩] {⟨2: envname⟩} [⟨3: numberedlike⟩] {⟨4: caption⟩} [⟨5:
within⟩]
Modified from the original to remember the environment.

293 \DeclareDocumentCommand\newmdtheoremenv{O{} m o m o }{%
294   \ifboolexpr{ test {\IfNoValueTF {#3}} and test {\IfNoValueTF {#5}} }{%
295     {\newtheorem{#2}{#4}}%
296     {%
297       \IfValueT{#3}{\newtheorem{#2}[#3]{#4}}%
298       \IfValueT{#5}{\newtheorem{#2}{#4}[#5]}%
299     }%
300   \BeforeBeginEnvironment{#2}{%
301     \renewcommand*\LWR@mdthisenv}{md#2}}%
302   \begin{mdframed} [#1]}%
303 \AfterEndEnvironment{#2}{%
304   \end{mdframed}}%
305 }

```

File 192 **l warp-memhfixc.sty**

§ 288 Package **memhfixc**

Pkg **memhfixc** memhfixc is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{memhfixc}

File 193 **l warp-metalogo.sty**

§ 289 Package **metalogo**

(Emulates or patches code by ANDREW GILBERT MOSCHOU.)

Pkg **metalogo** metalogo is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{metalogo}

```

2 \newcommand\setlogokern[2]{}
3 \newcommand\setlogodrop[2][XeTeX]{}
4 \newcommand\setLaTeXa[1]{}
5 \newcommand\setLaTeXee[1]{}

```

```
6 \newcommand\seteverylogo[1]{}
7 \newcommand\everylogo[1]{}
```

File 194 **lwarf-mhchem.sty**

§ 290 Package **mhchem**

(Emulates or patches code by MARTIN HENSEL.)

Pkg **mhchem** **mhchem** is patched for use by **lwarf**.

mhchem expressions are converted to SVG math. Inline expressions use hashed filenames to allow reuse, and assume that any **mhchem** options are global.

⚠ **MATHJAX and mhchem** The MATHJAX **mhchem** extension is not yet used. If MATHJAX is used for math in the rest of the document, **lwarf** converts standalone **mhchem** expressions into SVG math images, but expressions inside math must be placed between **\displaymathother** and **\displaymathnormal**:

```
\displaymathother
\[ \ce{ ... } \]      ...    $ \ce { ... } $
\displaymathnormal
```

⚠ **nested math** When producing HTML output, **lwarf** does not support the use of nested dollar signs in **mhchem** expressions.

For some examples from the **mhchem** manual, change as follows:

$\ce{NaOH(aq,$\infty)}$	% old
$\ce{NaOH(aq,\infty)}$	% new
$\ce{Fe(CN)_{$\frac{6}{2}$}}$	% old
$\ce{Fe(CN)_{\frac{6}{2}}}$	% new
$\ce{NO_{x}}$	% old
$\ce{NO_x}$	% new
$\ce{NO_{x}}$	% old
$\ce{NO_x}$	% new
$\ce{[$cis$-$][PtCl2(NH3)2]}$	% old
$\ce{[\mathit{cis}]-[PtCl2(NH3)2]}$	% new

for HTML output: 1 \LWR@ProvidesPackagePass{mhchem}

The original definition of \ce:

```
2 \LetLtxMacro{\LWR@mhchem@origce}{\ce}
```

The new definition, called from the new \ce after math shift is set. The starred lateximage uses a hashed filename for the SVG image. The alt tag is set to the mhchem expression.

```
3 \newcommand{\LWR@mhchem@HTML@ce}[1]{%
4 \begin{lateximage}*[\\textbackslash{}ce\\{\LWR@HTMLsanitize{#1}\\}]%
5 \LWR@mhchem@origce{#1}%
6 \end{lateximage}%
7 \endgroup%
8 \addtocounter{\LWR@mhchem@cedepth}{-1}%
9 }
```

Only set math shift if outer depth:

```
10 \newcounter{\LWR@mhchem@cedepth}
11 \setcounter{\LWR@mhchem@cedepth}{0}
```

The new \ce. Sets math shift then continues.

```
12 \renewcommand{\ce}{%
13 \begingroup%
14 \ifnumequal{\value{\LWR@mhchem@cedepth}}{0}{%
15   \catcode`\$=3% math shift
16 }{%
17 \addtocounter{\LWR@mhchem@cedepth}{1}%
18 \LWR@mhchem@HTML@ce%
19 }}
```

The original definition of \cesplit:

```
20 \LetLtxMacro{\LWR@mhchem@origcesplit}{\cesplit}
```

The new definition, called from the new \cesplit after math shift is set. The starred lateximage uses a hashed filename for the SVG image. The alt tag is set to the mhchem expression.

```
21 \newcommand*{\LWR@mhchem@HTML@cesplit}[2]
22 {%
23 \begin{lateximage}*[\\textbackslash{}cesplit\\{\LWR@HTMLsanitize{#2}\\}]%
24 \LWR@mhchem@origcesplit{#1}{#2}%
25 \end{lateximage}%
26 \endgroup%
27 }
```

Only set math shift if outer depth:

```
28 \newcounter{LWR@mhchem@cesplitdepth}
29 \setcounter{LWR@mhchem@cesplitdepth}{0}
```

The new `\cesplit`. Sets math shift then continues.

```
30 \renewcommand{\cesplit}{%
31 \begingroup%
32 \ifnumequal{\value{LWR@mhchem@cesplitdepth}}{0}{%
33   \catcode`\$=3% math shift
34 }{}%
35 \addtocounter{LWR@mhchem@cesplitdepth}{1}%
36 \LWR@mhchem@HTML@cesplit%
37 }
```

Resore originals inside a `lateximage`:

```
38 \appto\LWR@restoreorigformatting{%
39 \LetLtxMacro{\ce}{\LWR@mhchem@origce}%
40 \LetLtxMacro{\cesplit}{\LWR@mhchem@origcesplit}%
41 }
```

File 195 `lwarp-microtype.sty`

§ 291 Package **microtype**

(Emulates or patches code by R SCHLICHT.)

Pkg `microtype` `microtype` is pre-loaded by `lwarp`. All user options and macros are ignored and disabled.

for HTML output: Discard all options for `lwarp-microtype`:

```
1 \LWR@ProvidesPackageDrop{microtype}

2 \DeclareDocumentCommand{\DeclareMicrotypeSet}{o m m} {}
3 \DeclareDocumentCommand{\UseMicrotypeSet}{o m} {}
4 \DeclareDocumentCommand{\DeclareMicrotypeSetDefault}{o m} {}
5 \DeclareDocumentCommand{\SetProtrusion}{o m m} {}
6 \DeclareDocumentCommand{\SetExpansion}{o m m} {}
7 \DeclareDocumentCommand{\SetTracking}{o m m} {}
8 \DeclareDocumentCommand{\SetExtraKerning}{o m m} {}
9 \DeclareDocumentCommand{\SetExtraSpacing}{o m m} {}
10 \DeclareDocumentCommand{\DisableLigatures}{o m} {}
11 \DeclareDocumentCommand{\DeclareCharacterInheritance}{o m m} {}
12 \DeclareDocumentCommand{\DeclareMicrotypeVariants}{m} {}
```

```

13 \DeclareDocumentCommand{\DeclareMicrotypeAlias}{m m}={}
14 \DeclareDocumentCommand{\LoadMicrotypeFile}{m}{}
15 \DeclareDocumentCommand{\DeclareMicrotypeBabelHook}{m m}={}
16 \DeclareDocumentCommand{\microtypesetup}{m}{}
17 \DeclareDocumentCommand{\microtypecontext}{m}{}
18 \DeclareDocumentCommand{\textmicrotypecontext}{m m}{#2}
19 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
20 \DeclareDocumentCommand{\lsstyle}{m}{}
21 \DeclareDocumentCommand{\textls}{o +m}{}
22 \DeclareDocumentCommand{\lslig}{m}{#1}
23 }
24 \def\DeclareMicrotypeSet#1{\@gobbletwo}
25 \def\DeclareMicrotypeVariants#1{\@gobble}
26 \@onlypreamble\DeclareMicrotypeSet
27 \@onlypreamble\UseMicrotypeSet
28 \@onlypreamble\DeclareMicrotypeSetDefault
29 \@onlypreamble\DisableLigatures
30 \@onlypreamble\DeclareMicrotypeVariants
31 \@onlypreamble\DeclareMicrotypeBabelHook

```

File 196 **lwarf-midfloat.sty**

§ 292 Package **midfloat**

(Emulates or patches code by SIGITAS TOLUŠIS.)

Pkg **midfloat** **midfloat** is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{midfloat}

2 \newenvironment{strip}[1][]{\{}{\}}
3 \newskip\stripsep

```

File 197 **lwarf-midpage.sty**

§ 293 Package **midpage**

Pkg **midpage** **midpage** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{midpage}

2 \newenvironment{midpage}
3 {\begin{BlockClass}{%

```

```
4     \LWR@print@mbox{margin-top:6ex} ; \LWR@print@mbox{margin-bottom:6ex}%
5 ]{\midpage}
6 {\end{BlockClass}}
```

File 198 **l warp-morefloats.sty**

§ 294 Package **morefloats**

Pkg **morefloats** morefloats is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{morefloats}

File 199 **l warp-moreverb.sty**

§ 295 Package **moreverb**

(Emulates or patches code by ROBIN FAIRBAIRNS.)

Pkg **moreverb** moreverb is supported with some patches.

for HTML output: 1 \begin{warpHTML}

```
2 \LWR@ProvidesPackagePass{moreverb}

3 \BeforeBeginEnvironment{verbatimtab}{%
4 \LWR@forcenewpage
5 \LWR@atbeginverbatim{3}{Verbatim}%
6 }
7 \AfterEndEnvironment{verbatimtab}{%
8 \LWR@afterendverbatim{1}%
9 }
10
11
12 \LetLtxMacro{\LWRMV@orig@verbatimtabinput}{\verbatimtabinput}
13
14 \renewcommand{\@verbatimtabinput}[2][]{%
15 \LWR@forcenewpage
16 \LWR@atbeginverbatim{3}{Verbatim}%
17 \LWRMV@orig@verbatimtabinput[#1]{#2}%
18 \LWR@afterendverbatim{1}%
19 }
20
21 \BeforeBeginEnvironment{listing}{%
22 \LWR@forcenewpage
```

```
23 \LWR@atbeginverbatim{3}{programlisting}%
24 }
25
26 \AfterEndEnvironment{listing}{%
27 \LWR@afterendverbatim{1}%
28 }
29
30 \BeforeBeginEnvironment{listingcont}{%
31 \LWR@forcenewpage
32 \LWR@atbeginverbatim{3}{programlisting}%
33 }
34
35 \AfterEndEnvironment{listingcont}{%
36 \LWR@afterendverbatim{1}%
37 }

38 \LetLtxMacro\LWRMV@@listinginput\@listinginput
39
40 \renewcommand{\@listinginput}[3][]{%
41 \LWR@forcenewpage
42 \LWR@atbeginverbatim{3}{programlisting}%
43 \LWRMV@@listinginput[#1]{#2}{#3}%
44 \LWR@afterendverbatim{1}%
45 }
46
47
48 \renewenvironment*{boxedverbatim}
49 {
50 \LWR@forcenewpage
51 \LWR@atbeginverbatim{3}{boxedverbatim}%
52 \verbatim%
53 }
54 {
55 \endverbatim%
56 \LWR@afterendverbatim{1}%
57 }

58 \end{warpHTML}
```

File 200 **lwarf-mparhack.sty**

§ 296 Package **mparhack**

Pkg **mparhack** Ignored.

for HTML output:

Discard all options for l warp-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```

File 201 **l warp-multicol.sty**

§ 297 Package **multicol**

(Emulates or patches code by FRANK MITTELBACH.)

Pkg **multicol** **multicol** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{multicol}[2015/09/13]

Multicols are converted into a 1–3 column display, browser-supported.

The optional multicols heading is placed inside a <div> of class **multicolsheading**.

The content is placed inside a <div> of class **multicols**.

```
2 \begin{warpHTML}
```

Env **multicol** * {<numcols>} [<heading>]
3 \NewDocumentEnvironment{multicol}{s m o}

HTML <div> class to contain everything:

```
4 {  
5 \LWR@forcenewpage  
6 \BlockClass{multicol}
```

Optional HTML <div> class for the heading:

```
7 \IfValueT{#3}{\begin{BlockClass}{multicolsheading}#3\end{BlockClass}}}
```

When done with the environment, close the <div>:

```
8 {\endBlockClass}
```

Emulated null functions which are not used in HTML:

```
9 \newcommand*{\columnbreak}{}  
10 \newcommand*{\RLmulticolcolumns}{}  
11 \newcommand*{\LRmulticolcolumns}{}  
12  
13 \newlength{\premulticols}  
14 \newlength{\postmulticols}  
15 \newlength{\multicolssep}
```

```

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 \NewDocumentCommand{\docolaction}{s o m m}{%
27     \IfValueTF{#2}{#2}{#3}%
28 }

29 \end{warpHTML}

```

File 202 **lwarf-multicollrule.sty**

§ 298 Package **multicollrule**

Pkg **multicollrule** **multicollrule** is ignored.

for HTML output:

```

1 \RequirePackage{multicol}
2
3 \LWR@ProvidesPackageDrop{multicollrule}

4 \newcommand*{\SetMCRule}[1]{}

```

File 203 **lwarf-multirow.sty**

§ 299 Package **multirow**

(Emulates or patches code by PIET VAN OOSTRUM, ØYSTEIN BACHE, JERRY LEICHTER.)

Pkg **multirow** **multirow** is emulated during HTML output, and used as-is while inside a **lateximage**.

In a **lateximage**, the original print-mode versions are temporarily restored by **\LWR@restoreorigformatting**.

See section [71.23](#) for the print-mode versions.

for HTML output:

Remove the placeholder macro which was used if multirow was not loaded:

```
1 \LetLtxMacro\multirow\relax
2 \LWR@ProvidesPackagePass{multirow}
```

\LWR@multirowborder Set to left or right to create a thick border for the cell, for use by bigdelim:

```
3 \newcommand{\LWR@multirowborder}{}%
```

§ 299.1 Multirow

```
\multirow [⟨vpos⟩] {⟨numrows⟩} [⟨bigstruts⟩] {⟨width⟩} [⟨fixup⟩] {⟨text⟩}
4 \NewDocumentCommand{\LWR@HTML@multirow}{O{c} m o m o +m}%
5 {%
6 \LWR@traceinfo{*** LWR@HTML@multirow #1 #2 #4}%
7 \LWR@maybenewtablerow%
8 \LWR@tabularleftedge%
```

Print the start of a new table data cell:

```
9 \LWR@htmntag{td rowspan="#2" %
```

The vertical alignment, if given:

```
10 \IfValueT{#1}{%
11 \ifstrequal{#1}{b}{style="\LWR@print@mbox{vertical-align:bottom}" }{}%
12 \ifstrequal{#1}{t}{style="\LWR@print@mbox{vertical-align:top}" }{}%
13 }%
```

The left/right border, if given:

```
14 \ifdefvoid{\LWR@multirowborder}{}{%
15 style="\LWR@print@mbox{border-\LWR@multirowborder:} 2px dotted black ; %
16 \LWR@print@mbox{padding-\LWR@multirowborder:} 2px" %
17 }%
```

A class adds the column spec and the rule:

```
18 class="td%"
```

Append this column's spec:

```
19 \LWR@getexpparray{LWR@tablecolspec}{\arabic{LWR@tableLaTeXcolindex}}%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag. Also add the vertical bar class.

```

20 \LWR@addcmidruletrim%
21 \LWR@addleftmostbartag%
22 \LWR@printbartag{\arabic{LWR@tableLaTeXcolindex}}%
23 "%

24 \LWR@tdstartstyles%
25 \LWR@addcmidrulewidth%
26 \LWR@addcdashline%
27 \LWR@addtabularrulecolors%
28 \LWR@tdendstyles%
29 }%

```

The column's < spec:

```
30 \LWR@getexparray{\LWR@colbeforespec}{\arabic{LWR@tableLaTeXcolindex}}%
```

While printing the text, redefine \\ to generate a new line

```

31 \begingroup\LetLtxMacro{\\"}{\LWR@endofline}#6\endgroup%
32 \LWR@stoppars%
33 \global\boolfalse{\LWR@intabularmetadata}%
34 \renewcommand{\LWR@multirowborder}{}%
35 \LWR@traceinfo{*** \LWR@HTML@multirow done}%
36 }%
37
38 \LWR@formatted{multirow}

```

§ 299.2 Combined multicolumn and multirow

⚠ **\multicolumn & \multirow** lwarp does not support directly combining `\multicolumn` and `\multirow`. Use `\multicolumnrow` instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}
```

The two arguments for `\multicolumn` come first, followed by the five arguments for `\multirow`, many of which are optional, followed by the contents.

⚠ **skipped cells** As per `\multirow`, skipped cells to the right of the `\multicolumnrow` statement are not included in the source code on the same line. On the following lines,

⚠ **empty cells** `\mcolrowcell` must be used for each cell of each column and each row to be skipped:

```

... & \multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text} & ...
... & \mcolrowcell & \mcolrowcell & ...
... & \mcolrowcell & \mcolrowcell & ...

```

vposn Note that recent versions of multirow include a new optional vposn argument.

```
\multicolumnrow {{<1:cols>}} {{<2:halign>}} {{<3:vpos>}} {{<4:numrows>}} {{<5:bigstruts>}} {{<6:width>}} {{<7:fixup>}}
{{<8:text>}}
```

\@ifpackageloaded{multirow} determines if v2.0 or later of multirow was used, which included the \ProvidesPackage macro.

The HTML version follows.

\AtBeginDocument because the print version had to see if multirow was loaded before determining how to define \LWR@print@multicolumnrow.

```

39 \AtBeginDocument{
40
41 \NewExpandableDocumentCommand{\LWR@HTML@multicolumnrow}{m m 0{} m 0{} m 0{} +m}{%

```

Figure out how many extra HTML columns to add for @ and ! columns:

```
42 \LWR@tabularhtmlcolumns{\arabic{\LWR@tableLaTeXcolindex}}{#1}
```

Create the multicolumn/multirow tag:

```

43 \begingroup%
44 \LetLtxMacro{\\"}{\LWR@endofline}%
45 \LWR@domulticolumn[#3][#4]{\arabic{\LWR@tabhtmlcoltotal}}{#2}{#8}%
46 \endgroup%
```

Move to the next L^AT_EX column:

```
47 \addtocounter{\LWR@tableLaTeXcolindex}{#1}%
48 \addtocounter{\LWR@tableLaTeXcolindex}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```

49 \global\booltrue{\LWR@skipatbang}%
50 }%
51
52 \LWR@expandableformatted{multicolumnrow}%
53
54 }% \AtBeginDocument
```

File 204 **l warp-multitoc.sty**

§ 300 Package **multitoc**

Pkg **multitoc** **multitoc** is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{multitoc}

2 \newcommand{\multicolumnmtoc}{2}
3 \newcommand{\multicolumnlot}{2}
4 \newcommand{\multicolumnlof}{2}
5 \newcommand*{\immediateaddtocontents}[2]{}

---


```

File 205 **l warp-musicography.sty**

§ 301 Package **musicography**

(Emulates or patches code by ANDREW A. CASHNER.)

Pkg **musicography** **musicography** is patched for use by l warp.

Images are used for the meter symbols, since the HTML fonts tend not to be the correct size. The HTML alt tag copies C and 3/2, etc. Hashes are used for the meter images, which are then reused as necessary.

- ⚠ Note that browser support for musical symbols may be buggy. Copy/paste into a text editor works well.

for HTML output:

```
1 \LWR@ProvidesPackagePass{musicography}[2018/05/21]

2 \RenewDocumentCommand{\musSymbol}{ O{\musFont} m m m m }{%
3 \begin{lateximage}%
4 {\#1\kern\#2\raisebox{\#3}{\#5}\kern\#4}%
5 \end{lateximage}%
6 }
7
8 \RenewDocumentCommand{\musStemmedNote}{ m }{%
9 \begin{lateximage}%
10 \musSymbol{0.05em}{0.5ex}{0.2em}{\#1\musStem}%
11 \end{lateximage}%
12 }
13
```

```
14 \RenewDocumentCommand{\musFlaggedNote}{ m m }{%
15 \begin{lateximage}%
16 \musSymbol{0.05em}{0.5ex}{Opt}{#1\musStem}%
17 \musSymbol{Opt}{Opt}{0.9em}{#2}%
18 \end{lateximage}%
19 }
20
21 \RenewDocumentCommand{\musDottedNote}{ m }{%
22 \begin{lateximage}%
23 #1\musDot%
24 \end{lateximage}%
25 }
26
27 \RenewDocumentCommand{\musMeter}{ m m }{%
28 \begin{lateximage}*[#1/#2] [#1#2]%
29 \musStack{#1}{#2}\kern0.05em%
30 \end{lateximage}%
31 }
32
33 \RenewDocumentCommand{\meterCplus}{ m }{%
34 \begin{lateximage}* [C#1]%
35     \meterC{}\kern-0.7pt#1%
36 \end{lateximage}%
37 }
38
39 \RenewDocumentCommand{\meterC}{ }{%
40 \begin{lateximage}* [C]%
41 \musSymbolMeter{\symbol{83}}%
42 \end{lateximage}%
43 }
44
45 \RenewDocumentCommand{\meterCutC}{ }{%
46 \begin{lateximage}* [C|]%
47 \musSymbolMeter{\symbol{82}}%
48 \end{lateximage}%
49 }
50
51 \RenewDocumentCommand{\meterCThreeTwo}{ }{%
52 \begin{lateximage}* [C3/2]%
53 \meterCplus{\musStack{3}{2}}%
54 \end{lateximage}%
55 }
56
57 \RenewDocumentCommand{\musFlat}{}{\{\HTMLUnicode{266D}\}}
58 \RenewDocumentCommand{\musDoubleFlat}{}{\{\HTMLUnicode{1D12B}\}}
59 \RenewDocumentCommand{\musSharp}{}{\{\HTMLUnicode{266F}\}}
60 \RenewDocumentCommand{\musDoubleSharp}{}{\{\HTMLUnicode{1D12A}\}}
61 \RenewDocumentCommand{\musNatural}{}{\{\HTMLUnicode{266E}\}}
62
63 \RenewDocumentCommand{\musWhole}{}{\{\HTMLUnicode{1D15D}\}}
```

```

64 \RenewDocumentCommand{\musHalf}{()}{\HTMLUnicode{1D15E}}
65 \RenewDocumentCommand{\musQuarter}{()}{\HTMLUnicode{1D15F}}
66 \RenewDocumentCommand{\musEighth}{()}{\HTMLUnicode{1D160}}
67 \RenewDocumentCommand{\musSixteenth}{()}{\HTMLUnicode{1D161}}
68
69 \RenewDocumentCommand{\musWholeDotted}{()}{\HTMLUnicode{1D15D}\HTMLUnicode{1D16D}}
70 \RenewDocumentCommand{\musHalfDotted}{()}{\HTMLUnicode{1D15E}\HTMLUnicode{1D16D}}
71 \RenewDocumentCommand{\musQuarterDotted}{()}{\HTMLUnicode{1D15F}\HTMLUnicode{1D16D}}
72 \RenewDocumentCommand{\musEighthDotted}{()}{\HTMLUnicode{1D160}\HTMLUnicode{1D16D}}
73 \RenewDocumentCommand{\musSixteenthDotted}{()}{\HTMLUnicode{1D161}\HTMLUnicode{1D16D}}

```

File 206 **l warp-nameref.sty**

§ 302 Package **nameref**

Pkg **nameref** nameref is emulated by l warp.

for HTML output: Discard all options for l warp-nameref:

```

1 \typeout{Using the l warp html version of package 'nameref', discarding options.}
2 \typeout{    Are not using ProvidesPackage, so that other packages}
3 \typeout{    do not attempt to patch l warp's version of 'nameref'.}
4 \DeclareOption*{}
5 \ProcessOptions\relax

```

File 207 **l warp-natbib.sty**

§ 303 Package **natbib**

(Emulates or patches code by PATRICK W. DALY.)

Pkg **natbib** natbib is patched for use by l warp.

for HTML output: 1 \LWR@ProvidesPackagePass{natbib}

Replace math < and > with \textless and \textgreater:

A macro to compare:

```
2 \newcommand{\LWRNB@NAT@open}{$<$}
```

To patch \NAT@open and \NAT@close

```
3 \newcommand{\LWRNB@patchnatbibopenclose}{
```

```

4 \ifdef\streq{\NAT@open}{\LWRNB@NAT@open}
5 {
6     \renewcommand{\NAT@open}{\textless}
7     \renewcommand{\NAT@close}{\textgreater}
8 }{ }
9 }
```

Do it now in case `angle` was selected as an option:

```
10 \LWRNB@patchnatbibopenclose
```

Also patch `\setcitestyle` to patch after settings are made:

```

11 \let\LWRNB@origsetcitestyle\setcitestyle
12
13 \renewcommand{\setcitestyle}[1]{%
14 \LWRNB@origsetcitestyle{#1}%
15 \LWRNB@patchnatbibopenclose%
16 }
```

File 208 **lwarf-nccfancyhdr.sty**

§ 304 Package **nccfancyhdr**

(Emulates or patches code by ALEXANDER I. ROZHENKO.)

Pkg nccfancyhdr nccfancyhdr is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{nccfancyhdr}

2 \newcommand*\headrulewidth{0pt}
3 \newcommand*\footrulewidth{0pt}
4 \newcommand{\headstrutheight}{0pt}
5 \newcommand{\footstrutheight}{0pt}
6 \newcommand*\headrule[1]{\hrule height #1}
7 \newcommand*\footrule[1]{\hrule height #1}
8
9 \newdimen\headwidth
10 \newcommand*\extendedheaders{0pt}
11 \newcommand*\normalheaders{0pt}
12
13 \newcommand*\fancyhead[2][]{\def\@fancyhead{#1}}
14 \newcommand*\fancyfoot[2][]{\def\@fancyfoot{#1}}
15 \newcommand*\fancyhf[2][]{\def\@fancyhf{#1}}
16 \newcommand*\fancypagestyle[2][]{\def\@fancypagestyle{#1}}
17 \newcommand*\lhead[2][]{\def\@lhead{#1}}
```

```

18 \newcommand*{\chead}[2] []
19 \newcommand*{\rhead}[2] []
20 \newcommand*{\lfoot}[2] []
21 \newcommand*{\cfoot}[2] []
22 \newcommand*{\rfoot}[2] []
23
24 \newcommand{\nouppercase}[1]{#1}
25
26 \NewDocumentCommand{\fancycenter}{o o m m m} {}
27
28 \NewDocumentCommand{\newpagestyle}{m o m} {}
29
30 \newcommand*{\iffloatpage}[2]{#2}
31 \newcommand*{\ifftopfloat}[2]{#2}
32 \newcommand*{\iffbotfloat}[2]{#2}

```

File 209 **lwarf-needspace.sty**

§ 305 Package **needspace**

(Emulates or patches code by PETER WILSON.)

Pkg **needspace** needspace is not used during HTML conversion.

for HTML output: Discard all options for lwarf-needspace:

```

1 \LWR@ProvidesPackageDrop{needspace}
2
3 \DeclareDocumentCommand{\needspace}{m} {}
4 \DeclareDocumentCommand{\Needspace}{s m} {}

```

File 210 **lwarf-nextpage.sty**

§ 306 Package **nextpage**

(Emulates or patches code by PETER WILSON.)

Pkg **nextpage** nextpage is nullified.

for HTML output: Discard all options for lwarf-nextpage.

```

1 \LWR@ProvidesPackageDrop{nextpage}
2
3 \DeclareDocumentCommand{\cleartoevenpage}{o} {}
3 \DeclareDocumentCommand{\movetoevenpage}{o} {}

```

```

4 \DeclareDocumentCommand{\cleartooddpage}{o} {}
5 \DeclareDocumentCommand{\movetooddpage}{o} {}

```

File 211 **l warp-nicefrac.sty**

§ 307 Package **nicefrac**

(Emulates or patches code by AXEL REICHERT.)

Pkg nicefrac nicefrac is patched for use by l warp.

for HTML output: 1 \LWR@ProvidesPackagePass{nicefrac}[1998/08/04]

```

2 \DeclareRobustCommand*\LWR@HTML@@UnitsNiceFrac}[3] [] {%
3   %% localize font selection
4   #1{%
5     \LWR@textcurrentfont{%
6       \InlineClass{numerator}{#2}%
7       /%
8       \InlineClass{denominator}{#3}%
9     }%
10   }%
11 }%
12 }
13
14 \LWR@formatted{@UnitsNiceFrac}
15
16 \DeclareRobustCommand*\LWR@HTML@@UnitsUglyFrac}[3] [] {%
17   %% localize font selection
18   #1{\LWR@textcurrentfont{#2/#3}}%
19 }%
20 }
21
22 \LWR@formatted{@UnitsUglyFrac}

```

For Mathjax:

23 \CustomizeMathJax{\newcommand{\nicefrac}[3] [] {#2/#3}}

File 212 **l warp-nonfloat.sty**

§ 308 Package **nonfloat**

(Emulates or patches code by KAI RASCHER.)

Pkg nonfloat nonfloat is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{nonfloat}

2 \LetLtxMacro{\topcaption}{\caption}
3 \newcommand{\figcaption}{\def\@capttype{figure}\caption}
4 \newcommand{\tabcaption}{\def\@capttype{table}\topcaption}
5 \newenvironment{narrow}[2]{}{}
```

File 213 **lwarf-nonumonpart.sty**

§ 309 Package **nonumonpart**

Pkg nonumonpart nonumonpart is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{nonumonpart}
```

File 214 **lwarf-nopageno.sty**

§ 310 Package **nopageno**

Pkg nopageno nopageno is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{nopageno}
```

File 215 **lwarf-notespages.sty**

§ 311 Package **notespages**

Pkg notespages notespages is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{notespages}

2 \newcommand*{\npnotesname}={}
3 \newcommand*{\npnotestext}={}
4 \newcommand*{\remainingtextheight}={}
5 \newdimen\remainingtextheight
6 \newcommand*{\notestitletext}={}
7 \newcommand*{\notesareatext}={}
8 \newcommand*{\npnplinfo}[1]{}
```

```

9 \newcommand*{\tracingnpmarks}){}
10 \newcommand*{\notespage}[1][]{}
11 \newcommand*{\notespages}[1][]{}
12 \newcommand*{\notesfill}[1][]{}
13 \newcommand*{\setnotespages}[1]({})
14 \newcommand*{\definenotesoption}[2]({})
15 \newcommand{\definenotesstyle}[2]({})
16 \newcommand{\definetitlestyle}[2]({})
17 \newcommand{\nppatchchapter}[1]({})
18 \newcommand{\npunpatchchapter}({})

```

File 216 **l warp-nowidow.sty**

§ 312 Package **nowidow**

(Emulates or patches code by RAPHAËL PINSON.)

Pkg **nowidow** nowidow is not used during HTML conversion.

Discard all options for l warp-nowidow:

for HTML output: 1 \LWR@ProvidesPackageDrop{nowidow}

```

\nowidow  [<lines>]
\setnowidow [<lines>]

2 \newcommand*{\nowidow}[1][]{}
3 \newcommand*{\setnowidow}[1]({})

\noclub  [<lines>]
\setnoclub [<lines>]

4 \newcommand*{\noclub}[1]({})
5 \newcommand*{\setnoclub}[1]({})

```

File 217 **l warp-ntheorem.sty**

§ 313 Package **ntheorem**

(Emulates or patches code by WOLFGANG MAY, ANDREAS SCHEDLER.)

Pkg **ntheorem** ntheorem is patched for use by l warp.

Table 15: Ntheorem package—css styling of theorems and proofs

Theorem: <div> of class theorembody<theoremstyle>

Theorem Header: of class theoremheader<style>

where <theoremstyle> is plain, break, etc.

§ 313.1 Limitations

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

§ 313.2 Options

Options `amsthm` or `standard` choose which set of theorems and proofs to initialize.

⚠ Disabled options The options `thmmarks` and `amsmath` are disabled, since they heavily modify the underlying math code. Theorem marks are emulated. The AMS-math modifications are not done.

Option `thref` is disabled because `cleveref` functions are used instead. `\thref` is emulated.

Option `hyperref` is disabled because l warp emulated `hyperref`.

for HTML output:

```

1 \DeclareOption{thref}{}  

2  

3  

4 \newbool{LWR@ntheoremmarks}  

5 \boolfalse{LWR@ntheoremmarks}  

6  

7 \DeclareOption{thmmarks} {  

8 \booltrue{LWR@ntheoremmarks}  

9 \newif\ifsetendmark\setendmarktrue  

10 }  

11  

12  

13 \newbool{LWR@ntheoremamsthm}  

14 \boolfalse{LWR@ntheoremamsthm}  

15

```

```

16 \DeclareOption{amsthm}{\booltrue{LWR@ntheoremamsthm}}
17
18
19 \DeclareOption{amsmath}={}
20 \DeclareOption{hyperref}={}
21
22 \LWR@ProvidesPackagePass{ntheorem}

```

§ 313.3 Remembering the theorem style

Storage for the style being used for new theorems.

```

23 \newcommand{\LWR@newtheoremstyle}[plain]

24 \AtBeginDocument{
25 @ifpackageloaded{cleveref}{
26 \gdef\@thm#1#2#3{%
27   \if@thmmarks
28     \stepcounter{end}\InTheoType{ctr}%
29   \fi
30   \renewcommand{\InTheoType}{#1}%
31   \if@thmmarks
32     \stepcounter{curr#1ctr}%
33     \setcounter{end#1ctr}{0}%
34   \fi
35   \refstepcounter[#1]{#2}% <<< cleveref modification
36   \theorem@prework
37   \LWR@forcenewpage% l warp
38   \BlockClass{theorembody#1}\LWR@thisthmstyle% l warp
39   \trivlist % latex's \trivlist, calling latex's \@trivlist unchanged
40   \ifuse@newframeskips % cf. latex.ltx for topsepadd: \@trivlist
41     \ifthm@inframe
42       \thm@topsep\theoreminframelpreskipamount
43       \thm@topsepadd\theoreminframepostskipamount
44     \else
45       \thm@topsep\theorempreskipamount
46       \thm@topsepadd\theorempostskipamount
47     \fi
48   \else% oldframeskips
49     \thm@topsep\theorempreskipamount
50     \thm@topsepadd\theorempostskipamount
51     \ifvmode\advance\thm@topsepadd\partopsep\fi
52   \fi
53   \atopsep\thm@topsep
54   \atopsepadd\thm@topsepadd
55   \advance\linewidth -\theorem@indent
56   \advance\linewidth -\theorem@rightindent
57   \advance\@totalleftmargin \theorem@indent
58   \parshape \one \@totalleftmargin \linewidth

```

```

59     \@ifnextchar[{`@ythm{#1}{#2}{#3}}{\@xthm{#1}{#2}{#3}}
60 }
61 }{%
62 \gdef\@thm#1#2#3{%
63   \if@thmmarks
64     \stepcounter{end}\InTheoType ctr}%
65   \fi
66   \renewcommand{\InTheoType}{#1}%
67   \if@thmmarks
68     \stepcounter{curr#1ctr}%
69     \setcounter{end#1ctr}{0}%
70   \fi
71   \refstepcounter{#2}%
72   \theorem@prework
73   \LWR@forcenewpage% l warp
74   \BlockClass{theorembody#1}\LWR@thisthmstyle% l warp
75 \trivlist % latex's \trivlist, calling latex's \trivlist unchanged
76 \ifuse@newframeskips % cf. latex.ltx for topsepadd: \trivlist
77   \ifthm@inframe
78     \thm@topsep\theoreminframelpskipamount
79     \thm@topsepadd\theoreminframepostskipamount
80   \else
81     \thm@topsep\theorempskipamount
82     \thm@topsepadd\theorempostskipamount
83   \fi
84   \else% oldframeskips
85     \thm@topsep\theorempskipamount
86     \thm@topsepadd \theorempostskipamount
87     \ifvmode\advance\thm@topsepadd\partopsep\fi
88   \fi
89   \atopsep\thm@topsep
90   \atopsepadd\thm@topsepadd
91   \advance\linewidth -\theorem@indent
92   \advance\linewidth -\theorem@rightindent
93   \advance\@totalleftmargin \theorem@indent
94   \parshape \one \atotalleftmargin \linewidth
95   \@ifnextchar[{`@ythm{#1}{#2}{#3}}{\@xthm{#1}{#2}{#3}}}
96 }
97 }
98 }% AtBeginDocument

```

Patched to remember the style being used for new theorems:

```

99 \gdef\theoremstyle#1{%
100   \@ifundefined{th@#1}{\@warning
101     {Unknown theoremstyle '#1'. Using 'plain'}%
102     \theorem@style{plain}
103     \renewcommand{\LWR@newtheoremstyle}{plain}}% l warp
104   }%

```

```

105      {
106          \theorem@style{#1}
107          \renewcommand{\LWR@newtheoremstyle}{#1}%
108      }
109 }

```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

110
111 \gdef\@xnthm#1#2[#3]{%
112     \ifthm@tempif
113         \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}%
114         \expandafter\@ifundefined{c@#1}%
115             {\@definecounter{#1}}{}%
116         \@newctr{#1}[#3]%
117         \expandafter\xdef\csname the#1\endcsname{%
118             \expandafter\noexpand\csname the#3\endcsname \@thmcOUNTERsep
119                 {\noexpand\csname\the\theoremnumbering\endcsname{#1}}}%
120         \expandafter\gdef\csname mkheader@#1\endcsname
121             {\csname setparms@#1\endcsname
122                 \@thm{#1}{#1}{#2}
123             }%
124         \global\@namedef{end#1}{\@endtheorem}
125         \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\@nameuse{\LWR@thmstyle#1}}}%
126     \fi
127 }
128
129 \gdef\@ynthm#1#2{%
130     \ifthm@tempif
131         \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}%
132         \expandafter\@ifundefined{c@#1}%
133             {\@definecounter{#1}}{}%
134         \expandafter\xdef\csname the#1\endcsname
135             {\noexpand\csname\the\theoremnumbering\endcsname{#1}}%
136         \expandafter\gdef\csname mkheader@#1\endcsname
137             {\csname setparms@#1\endcsname
138                 \@thm{#1}{#1}{#2}
139             }%
140         \global\@namedef{end#1}{\@endtheorem}
141         \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\@nameuse{\LWR@thmstyle#1}}}%
142     \fi
143 }
144
145 \gdef\@othm#1[#2]#3{%
146     \@ifundefined{c@#2}{\@nocounterr{#2}}%
147     {\ifthm@tempif
148         \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}%
149         \global\@namedef{the#1}{\@nameuse{the#2}}%

```

```

150     \expandafter\protected@xdef\csname num@addtheoremline#1\endcsname{%
151         \noexpand\@num@addtheoremline{#1}{#3}}%
152     \expandafter\protected@xdef\csname nonum@addtheoremline#1\endcsname{%
153         \noexpand\@nonum@addtheoremline{#1}{#3}}%
154     \theoremkeyword{#3}%
155     \expandafter\protected@xdef\csname #1Keyword\endcsname{%
156         {\the\theoremkeyword}}%
157     \expandafter\gdef\csname mkheader@#1\endcsname{%
158         {\csname setparms@#1\endcsname{%
159             \@thm{#1}{#2}{#3}}%
160         }}%
161     \global\@namedef{end#1}{\@endtheorem}%
162     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\@nameuse{LWR@thmstyle#1}}}\% l warp
163 \fi}
164 }

```

§ 313.4 **HTML cross-referencing**

Mimics a float by incrementing the float counter and generating an HTML anchor. These are used for list-of-theorem cross-references.

```

165 \newcommand{\LWR@inctheorem}{%
166 \addtocounter{LWR@thisautoid}{1}}%
167 \LWR@stoppars%
168 \LWR@htmntag{a id="\LWR@print@mbox{autoid-\arabic{LWR@thisautoid}}"}\LWR@htmntag{/a}%
169 \LWR@startpars%
170 }

```

§ 313.5 **\newtheoremstyle**

The following are patched for css.

These were in individual files thp.sty for plain, thmb.sty for margin break, etc. They are gathered together here.

Each theorem is encased in a BlockClass environment of class theorembody<style>.

Each header is encased in an \InlineClass of class theoremheader<style>.

```

171 \gdef\newtheoremstyle#1#2#3{%
172   \expandafter\@ifundefined{th@#1}{%
173     {\expandafter\gdef\csname th@#1\endcsname{%
174       \def\@begintheorem####1####2{%
175         \LWR@inctheorem% l warp
176         #2}%
177         \def\@opargbegintheorem####1####2####3{%
178           \LWR@inctheorem% l warp
179           #3}%
180     }}%

```

```
181 }%
182 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
183 }
```

§ 313.6 Standard styles

```
184 \renewtheoremstyle{plain}%
185   {\item[
186     \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
187   {\item[
188     \InlineClass{theoremheaderplain}{##1\ ##2\ (#3)\theorem@separator}]}
189
190 \renewtheoremstyle{break}%
191   {\item[
192     \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
193   ]}%
194   {\item[
195     \InlineClass{theoremheaderbreak}%
196     {##1\ ##2\ (#3)\theorem@separator}\newline
197   ]}
198
199 \renewtheoremstyle{change}%
200   {\item[
201     \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]}%
202   {\item[
203     \InlineClass{theoremheaderchange}{##2\ ##1\ (#3)\theorem@separator}]}
204
205 \renewtheoremstyle{changebreak}%
206   {\item[
207     \InlineClass{theoremheaderchangebreak}%
208     {##2\ ##1\theorem@separator}\newline
209   ]}%
210   {\item[
211     \InlineClass{theoremheaderchangebreak}%
212     {##2\ ##1\ (#3)\theorem@separator}\newline
213   ]}
214
215 \renewtheoremstyle{margin}%
216   {\item[
217     \InlineClass{theoremheadermargin}{##2 \qquad ##1\theorem@separator}
218   ]}%
219   {\item[
220     \InlineClass{theoremheadermargin}{##2 \qquad ##1\ (#3)\theorem@separator}
221   ]}
222
223 \renewtheoremstyle{marginbreak}%
224   {\item[
225     \InlineClass{theoremheadermarginbreak}%
226       {##2 \qquad ##1\theorem@separator}\newline
227   ]}
```

```

227      ]}%
228  {\item[
229    \InlineClass{theoremheadermarginbreak}%
230    {##2 \qquad ##1\ (###3)\theorem@separator}\newline
231  ]}
232
233 \renewtheoremstyle{nonumberplain}%
234  {\item[
235    \InlineClass{theoremheaderplain}{##1\theorem@separator}]}%
236  {\item[
237    \InlineClass{theoremheaderplain}{##1\ (###3)\theorem@separator}}%
238
239 \renewtheoremstyle{nonumberbreak}%
240  {\item[
241    \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline
242  ]}
243  {\item[
244    \InlineClass{theoremheaderbreak}{##1\ (###3)\theorem@separator}\newline
245  ]}
246
247 \renewtheoremstyle{empty}%
248  {\item[]}%
249  {\item[
250    \InlineClass{theoremheaderplain}{##3}]}
251
252 \renewtheoremstyle{emptybreak}%
253  {\item[]}%
254  {\item[
255    \InlineClass{theoremheaderplain}{##3}]\ \ \newline}

```

§ 313.7 Additional objects

The following manually adjust the css for the standard configuration objects which are not a purely plain style:

```
256 \ifbool{LWR@ntheoremamsthm}{}{%
```

Upright text via CSS:

```

257  \newtheoremstyle{plainupright}%
258  {\item[
259    \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
260  {\item[
261    \InlineClass{theoremheaderplain}{##1\ ##2\ (###3)\theorem@separator}}%

```

Upright text and small caps header via CSS:

```

262  \newtheoremstyle{nonumberplainuprightsc}%
263  {\item[
```

```

264      \InlineClass{theoremheadersc}{##1\theorem@separator}]]}%
265      {\item[%
266          \InlineClass{theoremheadersc}{##1\ (##3)\theorem@separator}]}
267 }% not amsthm

```

§ 313.8 Renewed standard configuration

The following standard configuration is renewed using the new css:

```

268 \ifbool{LWR@ntheoremamsthm}{}{%
269 \ifx\thm@usestd@\undefined
270 \else
271     \theoremnumbering{arabic}
272     \theoremstyle{plain}
273     \RequirePackage{latexsym}
274     \theoremsymbol{\Box}
275     \theorembodyfont{\itshape}
276     \theoremheaderfont{\normalfont\bfseries}
277     \theoremseparator{ }
278     \renewtheorem{Theorem}{Theorem}
279     \renewtheorem{theorem}{Theorem}
280     \renewtheorem{Satz}{Satz}
281     \renewtheorem{satz}{Satz}
282     \renewtheorem{Proposition}{Proposition}
283     \renewtheorem{proposition}{Proposition}
284     \renewtheorem{Lemma}{Lemma}
285     \renewtheorem{lemma}{Lemma}
286     \renewtheorem{Korollar}{Korollar}
287     \renewtheorem{korollar}{Korollar}
288     \renewtheorem{Corollary}{Corollary}
289     \renewtheorem{corollary}{Corollary}
290
291     \theoremstyle{plainupright}
292     \theorembodyfont{\upshape}
293     \theoremsymbol{\HTMLunicode{25A1}}% UTF-8 white box
294     \renewtheorem{Example}{Example}
295     \renewtheorem{example}{Example}
296     \renewtheorem{Beispiel}{Beispiel}
297     \renewtheorem{beispiel}{Beispiel}
298     \renewtheorem{Bemerkung}{Bemerkung}
299     \renewtheorem{bemerkung}{Bemerkung}
300     \renewtheorem{Anmerkung}{Anmerkung}
301     \renewtheorem{anmerkung}{Anmerkung}
302     \renewtheorem{Remark}{Remark}
303     \renewtheorem{remark}{Remark}
304     \renewtheorem{Definition}{Definition}
305     \renewtheorem{definition}{Definition}
306

```

```

307   \theoremstyle{nonumberplainuprightsc}
308   \theoremsymbol{\HTMLunicode{220E}}% UTF-8 end-of-proof
309   \renewtheorem{Proof}{Proof}
310   \renewtheorem{proof}{Proof}
311   \renewtheorem{Beweis}{Beweis}
312   \renewtheorem{beweis}{Beweis}
313   \qedsymbol{\HTMLunicode{220E}}% UTF-8 end-of-proof
314
315   \theoremsymbol{}
316 \fi
317 }% not amsthm

```

§ 313.9 amsthm option

Only if the `amsthm` option was given:

```

318 \ifbool{LWR@ntheoremamsthm}{
319
320 \gdef\th@plain{%
321   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
322   \def\@begintheorem##1##2{%
323     \LWR@inctheorem% l warp
324     \item[%
325       \InlineClass{theoremheaderplain}{##1\ ##2.}%
326     ]}%
327   \def\@opargbegintheorem##1##2##3{%
328     \LWR@inctheorem% l warp
329     \item[%
330       \InlineClass{theoremheaderplain}{##1\ ##2\ (###3).}%
331     ]}%
332
333 \gdef\th@nonumberplain{%
334   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
335   \def\@begintheorem##1##2{%
336     \LWR@inctheorem% l warp
337     \item[%
338       \InlineClass{theoremheaderplain}{##1.}%
339     ]}%
340   \def\@opargbegintheorem##1##2##3{%
341     \LWR@inctheorem% l warp
342     \item[%
343       \InlineClass{theoremheaderplain}{##1\ (###3).}%
344     ]}%
345
346 \gdef\th@definition{%
347   \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
348   \def\@begintheorem##1##2{%
349     \LWR@inctheorem% l warp
350     \item[%

```

```
351 \InlineClass{theoremheaderdefinition}{##1\ ##2.}
352     ]}%
353 \def\@opargbegintheorem##1##2##3{%
354     \LWR@intheorem% l warp
355     \item[
356 \InlineClass{theoremheaderdefinition}{##1\ ##2\ (###3).}
357     ]}%
358
359 \gdef\th@nonumberdefinition{%
360     \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
361     \def\@begintheorem##1##2{%
362         \LWR@intheorem% l warp
363         \item[
364 \InlineClass{theoremheaderdefinition}{##1.}
365         ]}%
366     \def\@opargbegintheorem##1##2##3{%
367         \LWR@intheorem% l warp
368         \item[
369 \InlineClass{theoremheaderdefinition}{##1\ (###3).}
370         ]}%
371
372 \gdef\th@remark{%
373     \def\theorem@headerfont{\itshape}\normalfont%
374     \def\@begintheorem##1##2{%
375         \LWR@intheorem% l warp
376         \item[
377 \InlineClass{theoremheaderremark}{##1\ ##2.}
378         ]}%
379     \def\@opargbegintheorem##1##2##3{%
380         \LWR@intheorem% l warp
381         \item[
382 \InlineClass{theoremheaderremark}{##1\ ##2\ (###3).}
383         ]}%
384
385 \gdef\th@nonumberremark{%
386     \def\theorem@headerfont{\itshape}\normalfont%
387     \def\@begintheorem##1##2{%
388         \LWR@intheorem% l warp
389         \item[
390 \InlineClass{theoremheaderremark}{##1.}
391         ]}%
392     \def\@opargbegintheorem##1##2##3{%
393         \LWR@intheorem% l warp
394         \item[
395 \InlineClass{theoremheaderremark}{##1\ (###3).}
396         ]}%
397
398 \gdef\th@proof{%
399     \def\theorem@headerfont{\normalfont\bfseries}\itshape%
400     \def\@begintheorem##1##2{%
```

```

401      \LWR@inctheorem% l warp
402      \item[
403 \InlineClass{theoremheaderproof}{##1.}
404      ]}%
405 \def\@opargbegingroup{\LWR@inctheorem% l warp
406      \LWR@inctheorem% l warp
407      \item[
408 \InlineClass{theoremheaderproof}{##1\ (###3).}
409      ]}%
410
411
412
413 \newcounter{proof}%
414 \if@thmmarks
415     \newcounter{currproofctr}%
416     \newcounter{endproofctr}%
417 \fi
418
419 \gdef\proofsymbol{\openbox}
420
421 \newcommand{\proofname}{Proof}
422
423 \newenvironment{proof}[1][\proofname]{
424     \th@proof
425     \def\theorem@headerfont{\itshape}%
426     \normalfont
427     \theoremsymbol{\HTMLUnic{220E}}% UTF-8 end-of-proof
428     \thm{proof}{proof}{#1}
429 }%
430 {\endtheorem}
431
432 }{}% amsthm option

```

§ 313.10 Ending a theorem

Patched for css:

```

433 \let\LWR@origendtheorem\@endtheorem
434 \renewcommand{\@endtheorem}{}%
435 \ifbool{LWR@ntheoremmarks}{}%
436     \ifsetendmark%
437         \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
438         \setendmarkfalse%
439     \fi%
440 }{}%
441 \LWR@origendtheorem% also does \endtrivlist
442 \ifbool{LWR@ntheoremmarks}{\global\setendmarktrue}{}%
443 \endBlockClass%
444 }

```

§ 313.11 \NoEndMark

```
445 \gdef\NoEndMark{\global\setendmarkfalse}
```

§ 313.12 List-of

Redefined to reuse the float mechanism to add list-of-theorem links:

```
\thm@thmline {\langle 1: printed type\rangle} {\langle 2: #\rangle} {\langle 3: optional\rangle} {\langle 4: page\rangle}
```

```
446 \renewcommand{\thm@@thmline@noname}[4]{%
447 \hypertocfloat{1}{theorem}{thm}{#2 #3}{}}%
448 }
449
450 \renewcommand{\thm@@thmline@name}[4]{%
451 \hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}}%
452 }
```

This was redefined by ntheorem when loaded, so it is now redefined for lwarp:

```
453 \def\thm@@thmline{\thm@@thmline@name}
```

Patch for css:

```
454 \def\listtheorems#1{%
455 \LWR@htmlelementclass{nav}{lothm}%
456 \begingroup
457 \c@tocdepth=-2%
458 \def\thm@list{\#1}\thm@processlist
459 \endgroup
460 \LWR@htmlelementclassend{nav}{lothm}%
461 }
```

§ 313.13 Symbols

Proof QED symbol:

```
462 \newcommand{\qed}{\qquad\the\qedsymbol}
463
464 \AtBeginDocument{
465 \@ifundefined{LWR@orig@openbox}{%
466 \LetLtxMacro{\LWR@orig@openbox}{\openbox}%
467 \LetLtxMacro{\LWR@orig@blacksquare}{\blacksquare}%
468 \LetLtxMacro{\LWR@orig@Box}{\Box}%
469
470 \def\openbox{\text{\HTMLunicode{25A1}}}% UTF-8 white box
471 \def\blacksquare{\text{\HTMLunicode{220E}}}% UTF-8 end-of-proof
472 \def\Box{\text{\HTMLunicode{25A1}}}% UTF-8 white box
473 }
```

```

474 \appto\LWR@restoreorigformatting{%
475 \LetLtxMacro\openbox\LWR@orig@openbox%
476 \LetLtxMacro\blacksquare\LWR@orig@blacksquare%
477 \LetLtxMacro\Box\LWR@orig@Box%
478 }% appto
479 }{}% @ifundefined
480 }% AtBeginDocument

```

§ 313.14 Cross-referencing

```

\thref {\langle label\rangle}

481 \newcommand*{\thref}[1]{\cref{#1}}

```

File 218 **l warp-octave.sty**

§ 314 Package **octave**

(Emulates or patches code by ANDREW A. CASHNER.)

Pkg **octave** octave is patched for use by l warp.

for HTML output: 1 \LWR@ProvidesPackagePass{octave}[2017/10/31]

Remove the leading 1pt kern:

```

2 \RenewDocumentCommand{\@PrintTicks}{ m }{%
3 \kern-1pt% l warp
4 \@TickNum = #1%
5 \loop
6 \@Tick{}%
7 \advance\@TickNum by -1
8 \ifnum\@TickNum > 0
9 \repeat
10 }

```

Use unicode for the prime character:

```
11 \RenewDocumentCommand{\@Tick}{}{\HTMLunicode{2032}}
```

Catch the inline font:

```

12 \RenewDocumentCommand{\pitch}{ m o m }{%
13 \if@OctaveNumber%
14 }%
15 \pitchfont%

```

```

16      \LWR@textcurrentfont{%
17          \MakeUppercase{#1}%
18          \IfValueTF{#2}{#2}{}
19          \textsubscript{#3}%
20      }%
21 }%
22 \else%
23 {%
24     \pitchfont{%
25         \LWR@textcurrentfont{%
26             \GetOctaveTick{#1}[#2][#3]%
27         }%
28     }%
29 }%
30 \fi%
31 }

```

The original was hard to adapt to lwarf's handling of &.

```

32 \StartDefiningTabulars
33 \renewcommand{\octavetable}{%
34 \begin{tabular}{ll}
35 \octaveprimes \pitch{C}{0} & \octavenumbers \pitch{C}{0} \\
36 \octaveprimes \pitch{C}{1} & \octavenumbers \pitch{C}{1} \\
37 \octaveprimes \pitch{C}{2} & \octavenumbers \pitch{C}{2} \\
38 \octaveprimes \pitch{C}{3} & \octavenumbers \pitch{C}{3} \\
39 \octaveprimes \pitch{C}{4} & \octavenumbers \pitch{C}{4} \\
40 \octaveprimes \pitch{C}{5} & \octavenumbers \pitch{C}{5} \\
41 \octaveprimes \pitch{C}{6} & \octavenumbers \pitch{C}{6} \\
42 \octaveprimes \pitch{C}{7} & \octavenumbers \pitch{C}{7} \\
43 \end{tabular}
44 }
45 \StopDefiningTabulars

```

File 219 **lwarf-overpic.sty**

§ 315 Package **overpic**

(Emulates or patches code by ROLF NIEPRASCHK.)

Pkg **overpic** overpic is patched for use by lwarf.

- ⚠ **scaling** The macros `\overpicfontsize` and `\overpicfontskip` are used during HTML generation. These are sent to `\fontsize` to adjust the font size for scaling differences between the print and HTML versions of the document. Renew these macros before using the `overpic` and `Overpic` environments.

See section 83.2 for the print-mode version of \overpicfontsize and \overpicfontskip.

for HTML output:

```

1 \LWR@ProvidesPackagePass{overpic}

2 \newcommand*\overpic[1]{\begin{overpic}[#1]\end{overpic}}
3 \newcommand*\overpic[2]{\begin{overpic}[#1]{#2}\end{overpic}}
4
5 \BeforeBeginEnvironment{overpic}{%
6   \begin{latentimage}%
7   \begin{overpic}[#1]{#2}%
8   \selectfont%
9 \end{overpic}%
10 \endlatentimage%
11 \AfterEndEnvironment{overpic}{\end{overpic}}%
12
13 \BeforeBeginEnvironment{Overpic}{%
14   \begin{latentimage}%
15   \begin{overpic}[#1]{#2}%
16   \selectfont%
17 \end{overpic}%
18 \endlatentimage%
19 \AfterEndEnvironment{Overpic}{\end{overpic}}%

```

File 220 **lwarf-pagegrid.sty**

§ 316 Package **pagegrid**

Pkg pagegrid pagegrid is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{pagegrid}

2 \newcommand*\pagegridsetup[1]{}

```

File 221 **lwarf-pagenote.sty**

§ 317 Package **pagenote**

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

for HTML output:

```

1 \DeclareOption{page}{}
2 \LWR@ProvidesPackagePass{pagenote}

```

File 222 **l warp-pagesel.sty**

§ 318 Package **pagesel**

Pkg pagesel pagesel is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{pagesel}

File 223 **l warp-paralist.sty**

§ 319 Package **paralist**

(Emulates or patches code by BERND SCHANDL.)

Pkg paralist paralist is supported with minor changes.

for HTML output: 1 \LWR@ProvidesPackagePass{paralist}

The compact environments are identical to the regular ones:

```
2 \LetLtxMacro\compactitem\itemize
3 \LetLtxMacro\compactenum\enumerate
4 \LetLtxMacro\compactdesc\description
5 \LetLtxMacro\endcompactitem\enditemize
6 \LetLtxMacro\endcompactenum\endenumerate
7 \LetLtxMacro\endcompactdesc\enddescription
```

For the inline environments, revert \item to its original print-mode version:

```
8 \AtBeginEnvironment{inparaitem}{\LetLtxMacro\item\lwr@origitem}
9 \AtBeginEnvironment{inparaenum}{\LetLtxMacro\item\lwr@origitem}
10 \AtBeginEnvironment{inparadesc}{\LetLtxMacro\item\lwr@origitem}
```

Manual formatting of the description labels:

```
11 \def\paradescriptionlabel#1{{\normalfont\textbf{#1}}}
```

File 224 **l warp-parnotes.sty**

§ 320 Package **parnotes**

(Emulates or patches code by CHELSEA HUGHES.)

Pkg parnotes parnotes is supported with some patches.

for HTML output:

```
1 \LWR@ProvidesPackagePass{parnotes}

2 \long\def\PN@parnote@real#1#2{%
3   \parnotemark{#1}%
4   % Unless this is the first parnote in \PN@text, add a separator first
5   \unless\ifx\PN@text\empty\g@addto@macro\PN@text{\parnoteintercmd}\fi
6   % Redefine \@currentlabel to the parnote label, so \label works
7   \g@addto@macro\PN@text{%
8     \phantomsection%
9     \def\@currentlabel{#1}%
10    \def\cref@currentlabel{%
11      [parnotemark] [\arabic{parnotemark}] []\theparnotemark%
12    }%
13  }%
14  \g@addto@macro\PN@text{%
15    \LWR@textcurrentfont{%
16      \parnotemark{#1}\nolinebreak\thinspace#2%
17    }%
18  }%
19 }
20
21 \def\PN@parnotes@real{%
22   % We call \par later, so this avoids recursion with \PN@parnotes@auto
23   \PN@inparnotestrue
24   \unless\ifvmode\par\fi
25   % Avoid page breaks between a paragraph and its parnotes
26   \nopagebreak\addvspace{\parnotevskip}%
27   \LWR@forcenewpage%
28   \begin{BlockClass}{footnotes}%
29   {\parnotefmt{\PN@text}\par}%
30   \end{BlockClass}%
31   \global\def\PN@text{}%
32   \addvspace{\parnotevskip}%
33   %
34   % These can be enabled or disabled by package options
35   %
36   \PN@disable@indent
```

```
37     \PN@reset@optional
38     \PN@inparnotesfalse
39 }
40
41 \AtBeginDocument{
42 \crefname{parnotemark}{paragraph note}{paragraph notes}
43 \Crefname{parnotemark}{Paragraph note}{Paragraph notes}
44 }
```

File 225 **l warp-parskip.sty**

§ 321 Package **parskip**

Pkg **parskip** **parskip** is ignored.

for HTML output: Discard all options for **l warp-parskip**.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

File 226 **l warp-pbox.sty**

§ 322 Package **pbox**

(Emulates or patches code by SIMON LAW.)

Pkg **pbox** **pbox** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{pbox}

```
2 \NewDocumentCommand{\pbox}{O{} O{} O{} m +m}%
3 \booltrue{\LWR@minipagefullwidth}%
4 \parbox[#1][#2][#3]{#4}{#5}%
5 }
6
7 \newcommand{\settominwidth}[3][\columnwidth]%
8 \settowidth{#2}{#3}%
9 }
10
11 \newcommand{\widthofpbox}[1]%
12 \widthof{#1}%
13 }
```

File 227 **lwarf-pdfcomment.sty**

§ 323 Package **pdfcomment**

Pkg **pdfcomment** pdfcomment is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{pdfcomment}[2016/06/13]

2 \newenvironment{pdfsidelinecomment}[2] {} {}
3 \newcommand{\pdfcomment}[2] {} {}
4 \newcommand{\pdfmargincomment}[2] {} {}
5 \newcommand{\pdfmarkupcomment}[3] {} {#2}
6 \newcommand{\pdffreetextcomment}[2] {} {}
7 \newcommand{\pdfsquarecomment}[2] {} {}
8 \newcommand{\pdfcirclecomment}[2] {} {}
9 \newcommand{\pdflinecomment}[2] {} {}
10 \newcommand{\pdftooltip}[3] {} {#2}
11 \newcommand{\pdfcommentsetup}[2] {} {}
12 \newcommand{\listofpdfcomments}[1] {} {}
13 \newcommand{\setliststyle}[1] {}
14 \newcommand{\defineliststyle}[2] {}
15 \newcommand{\defineavatar}[2] {}
16 \newcommand{\definestyle}[2] {}
```

File 228 **lwarf-pdflandscape.sty**

§ 324 Package **pdflandscape**

Pkg **pdflandscape** pdflandscape is ignored.

for HTML output: Discard all options for lwarf-pdflandscape:

```
1 \LWR@ProvidesPackageDrop{pdflandscape}
```

File 229 **lwarf-pdfmarginpar.sty**

§ 325 Package **pdfmarginpar**

Pkg **pdfmarginpar** pdfmarginpar is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{pdfmarginpar}[2011/08/05]

2 \newcommand{\pdfmarginpar}[2][]{}
3 \newcommand{\pdfmarginparset}[1]{}

```

File 230 **l warp-pdfpages.sty**

§ 326 Package **pdfpages**

(Emulates or patches code by ANDREAS MATTHIAS.)

Pkg pdfpages pdfpages is patched for use by l warp.

Option link and linkname work:

```
\hyperlink{<filename>.pdf.<pagenumber>}{some text}
\hyperlink{<linkname>.<pagenumber>}{some text}
```

Options which make no sense in HTML are disabled.

for HTML output:

```

1 \LWR@ProvidesPackagePass{pdfpages}
```

Disable option which have no meaning for HTML output:

```

2 \define@key{pdfpages}{fitpaper}[false]{}
3 \define@key{pdfpages}{landscape}[false]{}
4 \define@key{pdfpages}{openright}[false]{}
5 \define@key{pdfpages}{signature}{}
6 \define@key{pdfpages}{signature*}{}
7 \define@key{pdfpages}{booklet}[false]{}
8 \define@key{pdfpages}{rotateoversize}[false]{}
9 \define@key{pdfpages}{doublepages}[false]{}
10 \define@key{pdfpages}{doublepagestwist}[false]{}
11 \define@key{pdfpages}{doublepagestwistodd}[false]{}
12 \define@key{pdfpages}{doublepagestwist*}[false]{}
13 \define@key{pdfpages}{doublepagestwistodd*}[false]{}
14 \define@key{pdfpages}{duplicatepages}[2]{}
15 \define@key{pdfpages}{thread}[false]{}
16 \define@key{pdfpages}{threadname}{}
17 \define@key{pdfpages}{linkfit}{}
18 \define@key{pdfpages}{linktodoc}[false]{}
19 \define@key{pdfpages}{linktodocfit}{}
20 \define@key{pdfpages}{linkfilename}{}
21 \define@key{pdfpages}{survey}[false]{}
```

```
22 \define@key{pdfpages}{survey-nolink}[false]{}
23 \define@key{pdfpages}{newwindow}[false]{}
```

Use print mode while measuring the page numbers:

```
24 \xpretocmd{\AM@getpagecount}{\LWR@restoreorigformatting}{}{}
```

Emulate a bit of eso-pic:

```
25 \newif\ifESO@texcoord
26
27 \newcommand{\ESO@HookIIBG}{}
28
29 \renewcommand{\AM@AddToShipoutPicture}{\g@addto@macro{\ESO@HookIIBG}}
30
31 \renewcommand{\ClearShipoutPicture}{}
```

\LWR@esopic@newpage At each \newpage.

```
32 \newcommand*{\LWR@esopic@newpage}{%
```

Is there something to draw?

```
33 \ifdefvoid{\ESO@HookIIBG}{%
34 {}%
35 {}%
```

If the link option was specified, add a hyper taraget:

```
36     \ifAM@link%
37         \hypertarget{\AM@linkname.\AM@page}{}%
38     \fi%
```

Draw inside a picture environment of the size of a virtual page:

```
39     \begingroup%
40     \setlength{\unitlength}{1in}%
41     \begin{picture}(8,10.5)%
42     \ESO@HookIIBG%
43     \end{picture}%
44     \endgroup%
45     \global\let\ESO@HookIIBG\empty%
46 }
47 }
```

\AM@output Patched to use \LWR@esopic@newpage.

```
48 \xpatchcmd{\AM@output}
49   {\newpage}
50   {\LWR@esopic@newpage}
51   {}
```

```
52      {\LWR@patcherror{pdfpages}{AM@output-1}}
53
54 \xpatchcmd{\AM@output}
55   {\newpage}
56   {\LWR@esopic@newpage}
57   {}
58   {\LWR@patcherror{pdfpages}{AM@output-2}}
59
60 \xpatchcmd{\AM@output}
61   {\newpage}
62   {\LWR@esopic@newpage}
63   {}
64   {\LWR@patcherror{pdfpages}{AM@output-3}}
```

\includepdf Patched to set a reasonable paper size.

```
65 \xpretocmd{\includepdf}{%
66   \begingroup%
67   \setlength{\paperwidth}{8in}%
68   \setlength{\paperheight}{10.5in}%
69 }{}{%
70
71 \xapptocmd{\includepdf}{%
72   \endgroup%
73 }{}{}}
```

\includepdfmerge Patched to set a reasonable paper size.

```
74 \xpretocmd{\includepdfmerge}{%
75   \begingroup%
76   \setlength{\paperwidth}{8in}%
77   \setlength{\paperheight}{10.5in}%
78 }{}{%
79
80 \xapptocmd{\includepdfmerge}{%
81   \endgroup%
82 }{}{}}
```

\AM@hyper@begin@i Hyper links are created by \LWR@esopic@newpage, so don't create them here:

```
83 \renewcommand{\AM@hyper@begin@i}{}{}
```

File 231 **lwarf-pdfprivacy.sty**

§ 327 Package **pdfprivacy**

Pkg pdfprivacy pdfprivacy is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{pdfprivacy}

File 232 **lwarf-pdfrender.sty**

§ 328 Package **pdfrender**

Pkg pdfrender pdfrender is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{pdfrender}

2 \newcommand*{\pdfrender}[1]{}
3 \newcommand{\textpdfrender}[2]{#2}

File 233 **lwarf-pdfsync.sty**

§ 329 Package **pdftools**

(Emulates or patches code by J. LAURENS.)

Pkg pdftools Emulated.

for HTML output: Discard all options for lwarf-pdftools:

1 \LWR@ProvidesPackageDrop{pdftools}
2 \newcommand*{\pdftools}{}
3 \newcommand*{\pdftoolsstart}{}
4 \newcommand*{\pdftoolsstop}{}

File 234 **l warp-pdftricks.sty**

§ 330 Package **pdftricks**

(Emulates or patches code by C. V. RADHAKRISHNAN, C. V. RAJAGOPAL, ANTOINE CHAMBERT-LOIR.)

Pkg pdftricks pdftricks is patched for use by l warp.

- ⚠ convert image files The pdftricks image files <jobname>-fig*.pdf must be converted to .svg, or else a missing file error will occur. The image files must also be converted again whenever they change. To convert the images:

Enter ⇒ l warpmk pdftosvg <jobname>-fig*.pdf

for HTML output: 1 \LWR@ProvidesPackagePass{pdftricks}

Reuse the print-mode images:

2 \def\PDFTfigname{\BaseJobname-fig\the\psfig}

If the .pdf images have not yet been converted to .svg then an error about a missing file will occur. Warn the user to convert the images.

3 \PackageWarning{l warp-pdftricks}{
4 When the pdftricks images change,
5 remember to convert PDF images to SVG using 'l warpmk pdftosvg *-fig.pdf',
6 }
7
8 \AfterEndDocument{\typeout{***}}
9 \AfterEndDocument{\typeout{*** Note: If pdftricks images are not found, new, or updated,}}
10 \AfterEndDocument{\typeout{*** \space use 'l warpmk pdftosvg \BaseJobname-fig*.pdf'}}}
11 \AfterEndDocument{\typeout{***}}

File 235 **l warp-pdflx.sty**

§ 331 Package **pdflx**

Pkg pdflx pdflx is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{pdflx}

File 236 **l warp-pfnote.sty**

§ 332 Package **pfnote**

Pkg **pfnote** **pfnote** is emulated.

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

for HTML output: 1 \LWR@ProvidesPackageDrop{pfnote}

File 237 **l warp-phfqit.sty**

§ 333 Package **phfqit**

(Emulates or patches code by PHILIPPE FAIST.)

Pkg **phfqit** **phfqit** is patched for use by **l warp**.

for HTML output: 1 \LWR@ProvidesPackagePass{phfqit}

```
2 \LetLtxMacro{\LWR@origbitstring}{\bitstring}
3
4 \renewcommand{\bitstring}[1]{%
5   \InlineClass[%
6     text-decoration: overline underline ;
7   ]{\bitstring}{#1}%
8 } \phfqit@bitstring{#1}%
9 }
10
11 \appto{\LWR@restoreorigformatting}{%
12 \LetLtxMacro{\bitstring}{\LWR@origbitstring}%
13 }
```

File 238 **l warp-placeins.sty**

§ 334 Package **placeins**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg placeins placeins is not used during HTML conversion.

Discard all options for l warp-placeins:

for HTML output: 1 \LWR@ProvidesPackageDrop{placeins}
2 \newcommand*{\FloatBarrier}{}

File 239 l warp-plarydshln.sty

§ 335 Package plarydshln

Pkg plarydshln plarydshln is emulated by l warp-arydshln.

for HTML output: 1 \LWR@ProvidesPackageDrop{plarydshln}
2 \LWR@origRequirePackage{l warp-arydshln}

File 240 l warp-plex.sty

§ 336 Package plex

Pkg plex plex is preloaded by jtarticle and related classes.

for HTML output: 1 \LWR@loadbefore{plex}
2
3 \LWR@ProvidesPackagePass{plex}[2017/07/21]
4 \let\state\relax
5
6 \DeclareExpandableDocumentCommand{\rensuij}{s o m}{#3}
7
8 % \layoutfloat{width,height}[pos]#4
9 \DeclareDocumentCommand{\layoutfloat}{d() o m}{}
10
11 % \DeclareLayoutCaption{type} <dir>(width)[pos1pos2]
12 \DeclareDocumentCommand{\DeclareLayoutCaption}{m d> d() o}{}
13
14 \LetLtxMacro\pcaption\caption
15
16 % \layoutcaption<dir>(width)[pos]
17 \DeclareDocumentCommand{\layoutcaption}{d> d() o}{}
18
19 \let\captiondir\relax

Add the optional <t/y> direction:

```

20 \RenewDocumentEnvironment{LWR@HTML@minipage}{d<> 0{t} 0{} 0{t} m}
21     {\LWR@HTML@sub@minipage{#2}{#3}{#4}{#5}}
22     {\endLWR@HTML@sub@minipage}
23
24 \RenewDocumentCommand{\LWR@HTML@parbox}{d<> 0{t} 0{} 0{t} m +m}
25 {
26 \LWR@traceinfo{parbox of width #4}%
27 \begin{minipage}[#2][#3][#4]{#5}%
28 #6
29 \end{minipage}%
30 }
31
32 % \pbox <t/y> [width] [1/r] {contents}
33 \RenewDocumentCommand{\pbox}{d<> 0{0pt} 0{c} m}{%
34 \booltrue{LWR@minipagewillwidth}%
35 \parbox{#2}{#4}%
36 }

```

picture, as modified by pext, is encapsulated by the l warp core.

File 241 **l warp-plextarydshln.sty**

§ 337 Package **plextarydshln**

Pkg **plextarydshln** plexarydshln is emulated by l warp-arydshln.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{plextarydshln}

2 \LWR@origRequirePackage{l warp-arydshln}

```

File 242 **l warp-plextcolortbl.sty**

§ 338 Package **plextcolortbl**

Pkg **plextcolortbl** plexcolortbl is emulated by l warp-arydshln.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{plextcolortbl}

2 \LWR@origRequirePackage{l warp-colortbl}

```

File 243 **l warp-prelim2e.sty**

§ 339 Package **prelim2e**

(Emulates or patches code by MARTIN SCHRÖDER.)

Pkg **prelim2e** Emulated.

for HTML output: Discard all options for l warp-prelim2e:

```
1 \LWR@ProvidesPackageDrop{prelim2e}
2 \newcommand{\PrelimText}{}
3 \newcommand{\PrelimTextStyle}{}
4 \newcommand{\PrelimWords}{}
```

File 244 **l warp-prettyref.sty**

§ 340 Package **prettyref**

(Emulates or patches code by KEVIN S. RULAND.)

Pkg **prettyref** prettyref is patched for use by l warp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{prettyref}
2 \newreformat{fig}{Figure \ref{\#1}}
3 \newreformat{tab}{Table \ref{\#1}}
```

File 245 **l warp-preview.sty**

§ 341 Package **preview**

Pkg **preview** preview is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{preview}[2017/04/24]
2 \newenvironment{preview}{}{}
3 \newenvironment{nopreview}{}{}
```

```

4 \NewDocumentCommand{\PreviewMacro}{s o o +m}(){}
5 \NewDocumentCommand{\PreviewEnvironment}{s o o +m}){}
6 \newcommand{\PreviewSnarfEnvironment}[2][]{}
7 \NewDocumentCommand{\PreviewOpen}{s o}){}
8 \NewDocumentCommand{\PreviewClose}{s o}){}
9 \let\ifPreview\iffalse% \fi for syntax highlighting

```

File 246 **l warp-psfrag.sty**§ 342 Package **psfrag***(Emulates or patches code by MICHAEL C. GRANT, DAVID CARLISLE.)*

Pkg psfrag psfrag is patched for use by l warp.

⚠ use psfrags The psfrags environment is modified to use lateximage to encapsulate the image. Always use a psfrags environment to contain any local \psfrag macros and the associated \includegraphics or \epsfig calls. Outside of a psfrags environment, psfrags adjustments will not be seen by l warp.

⚠ Tip: Use a mono-spaced font for the tags in the EPS file.

for HTML output: 1 \LWR@ProvidesPackagePass{psfrag}[1998/04/11]

A lateximage captures the modified image from the document.

```

2 \BeforeBeginEnvironment{psfrags}{%
3   \begin{lateximage}[( -psfrags -- \packagediagramname )]%
4 }
5
6 \AfterEndEnvironment{psfrags}{\end{lateximage}}

```

File 247 **l warp-psfragx.sty**§ 343 Package **psfragx***(Emulates or patches code by PASCAL KOCKAERT.)*

Pkg psfragx psfragx is patched for use by l warp.

for HTML output: 1 \LWR@ProvidesPackagePass{psfragx}[2012/05/02]

A lateximage captures the modified image from the document.

```

2 \def\pfx@includegraphicx#1#2{%
3   \begin{lateximage}[(-psfragx--\packagediagramname)]%
4     \mbox{\pfx@overpix{\#1}{\#2}\endpfx@overpix}%
5   \end{lateximage}%
6 }
7
8 \def\@@@overpix[#1]<#2>[#3]#4{%
9   \begin{lateximage}[(-psfragx--\packagediagramname)]%
10    \pfx@overpix{\#1,ovpfgd={#2},ovpbgd={#3}}{\#4}%
11 }
12
13 \def\endoverpix{%
14   \endpfx@overpix%
15   \end{lateximage}%
16 }
```

File 248 **lwarf-pst-eps.sty**

§ 344 Package **pst-eps**

(Emulates or patches code by HERBERT VOSS.)

Pkg **pst-eps** **pst-eps** is patched for use by **lwarf**.

for HTML output:

```

1 \LWR@ProvidesPackagePass{pst-eps}

2 \renewenvironment{TeXtoEPS}{}{}
3 \renewcommand{\PSTtoEPS}[3][]{}
```

File 249 **lwarf-pstool.sty**

§ 345 Package **pstool**

(Emulates or patches code by ZEBB PRIME, WILL ROBERTSON.)

Pkg **pstool** **pstool** is patched for use by **lwarf**.

\graphicspath is ignored, and the file directory must be stated.

⚠ **path and filename** The filename must not have a file extension.

Use

Enter ⇒ **lwarfpmk html**

followed by

Enter ⇒ l warpmk l images

for HTML output: 1 \LWR@ProvidesPackagePass{pstool}

Each image is placed inside a `\teximage` to capture the results of `psfrag`.

```

2 \renewcommand{\pstool@alwaysprocess}[3][]{%
3   \begin{teximage}[(\pstool@~\packagediagramname)]%
4   \includegraphics{#2.pdf}%
5   \end{teximage}%
6 }
7 \LetLtxMacro{\pstool@neverprocess}{\pstool@alwaysprocess}
8 \LetLtxMacro{\pstool@maybeprocess}{\pstool@alwaysprocess}
9
10 \renewcommand{\pstool@@psfragfig}[4]{%
11   \begin{teximage}[(\pstool@~\packagediagramname)]%
12   \includegraphics{#2.pdf}%
13   \end{teximage}%
14 }
```

File 250 **l warp-pstricks.sty**

§ 346 Package **pstricks**

(Emulates or patches code by TIMOTHY VAN ZANDT.)

Pkg pstricks pstricks is patched for use by l warp.

⚠ **use pspicture** All pstricks content should be contained inside a `pspicture` environment.

for HTML output: 1 \LWR@ProvidesPackagePass{pstricks}

```

2 \BeforeBeginEnvironment{pspicture}{\begin{teximage}[(pspicture)]}
3 \AfterEndEnvironment{pspicture}{\end{teximage}}
```

File 251 **l warp-pxatbegshi.sty**

§ 347 Package **pxatbegshi**

Pkg pxatbegshi pxatbegshi is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{pxatbegshi}
2 \LWR@origRequirePackage{lwarf-atbegshi}

File 252 **lwarf-pxeveryshi.sty**

§ 348 Package **pxeveryshi**

Pkg pxeveryshi pxeveryshi is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{pxeveryshi}
2 \LWR@origRequirePackage{lwarf-everyshi}

File 253 **lwarf-pxftnright.sty**

§ 349 Package **pxftnright**

Pkg pxftnright pxftnright is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{pxftnright}
2 \LWR@origRequirePackage{lwarf-ftnright}

File 254 **lwarf-pxjahyper.sty**

§ 350 Package **pxjahyper**

Pkg pjaxahyper pjaxahyper is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{pxjahyper}

File 255 **lwarf-quotchap.sty**

§ 351 Package **quotchap**

(Emulates or patches code by KARSTEN TINNEFELD, JAN KLEVER.)

Pkg quotchap quotchap is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{quotchap}

2 \newcommand{\@quotchap}{}%
3 \newlength{\LWR@quotchapwidth}
4
5 \let\@printcites\relax
6
7 \newcommand*{\@iprintcites}{%

```

Place the quotes inside a <div> of class quotchap, of the maximum selected width:

```

8 \begin{BlockClass}[max-width: \LWR@printlength{\LWR@quotchapwidth}]{quotchap}
9 %\begin{minipage}{\LWR@quotchapwidth}
10 \@quotchap
11 %\end{minipage}
12 \end{BlockClass}

```

Deactivate the quote printing:

```

13 \global\let\@printcites\relax
14 }
15
16 \NewEnviron{savequote}[1][\linewidth]{%

```

Remember the width, adjusted for HTML, and make the length assignment global, per:

<https://tex.stackexchange.com/questions/300823/why-is-setlength-ineffective-inside-a-tabular-environment>

```

17 \setlength{\LWR@quotchapwidth}{#1*2}%
18 \global\LWR@quotchapwidth=\LWR@quotchapwidth%

```

Remember the body, and activate the quote printing:

```

19 \global\let\@quotchap BODY
20 \global\let\@printcites\@iprintcites%
21 }

```

The quotation author is placed inside a <div> of class qauthor:

```

22 \newcommand{\qauthor}[1]{\begin{BlockClass}{qauthor}{#1}\end{BlockClass}}

```

\qsetcnfont is ignored:

```

23 \newcommand{\qsetcnfont}[1]{}

```

File 256 **l warp-quoting.sty**

§ 352 Package **quoting**

(Emulates or patches code by THOMAS TITZ.)

Pkg **quoting** quoting is patched for use by l warp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{quoting}
2 \xpatchcmd{\quoting}{\quo@begintext}
3   {\begin{LWR@blocktextcurrentfont}\quo@begintext}
4   {}
5   {\LWR@patcherror{quoting}{quoting}}
6
7 \xpatchcmd{\endquoting}{\quo@endtext}
8   {\quo@endtext\end{LWR@blocktextcurrentfont}}
9   {}
10 {\LWR@patcherror{imakeidx}{endquoting}}
```

File 257 **l warp-ragged2e.sty**

§ 353 Package **ragged2e**

(Emulates or patches code by MARTIN SCHRÖDER.)

Pkg **ragged2e** ragged2e is not used during HTML conversion.

Discard all options for l warp-ragged2e:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{ragged2e}
2 \LetLtxMacro\Centering\centering
3 \LetLtxMacro\RaggedLeft\raggedleft
4 \LetLtxMacro\RaggedRight\raggedright
5 \newcommand*\justifying(){}
6 \newlength{\CenteringLeftskip}
7 \newlength{\RaggedLeftLeftskip}
8 \newlength{\RaggedRightLeftskip}
9 \newlength{\CenteringRightskip}
10 \newlength{\RaggedLeftRightskip}
11 \newlength{\RaggedRightRightskip}
12 \newlength{\CenteringParfillskip}
```

```

13 \newlength{\RaggedLeftParfillskip}
14 \newlength{\RaggedRightParfillskip}
15 \newlength{\JustifyingParfillskip}
16 \newlength{\CenteringParindent}
17 \newlength{\RaggedLeftParindent}
18 \newlength{\RaggedRightParindent}
19 \newlength{\JustifyingParindent}
20 \newenvironment*{Center}{\center}{\endcenter}
21 \newenvironment*{FlushLeft}{\flushleft}{\endflushleft}
22 \newenvironment*{FlushRight}{\flushright}{\endflushright}
23 \newenvironment*{justify}{\justifying}{\endjustifying}

```

File 258 **l warp-realscripts.sty**

§ 354 Package **realscripts**

(Emulates or patches code by WILL ROBERTSON.)

Pkg **realscripts** **realscripts** is emulated. See **l warp.css** for the of class **supsubscript**.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{realscripts}

2 \let\realsuperscript\textsuperscript
3 \let\realsubscript\textsubscript
4
5 \let\fakesuperscript\textsuperscript
6 \let\fakesubscript\textsubscript
7
8 \newlength{\subsupersep}
9
10 \newcommand*{\LWR@realscriptsalign}{}%
11
12 \newcommand*{\LWR@setrealscriptsalign}[1]{%
13 \renewcommand*{\LWR@realscriptsalign}{}%
14 \ifthenelse{\equal{#1}{c}}{%
15   \renewcommand{\LWR@realscriptsalign}{\LWR@print@mbox{text-align:center} ; }%
16 }{}%
17 \ifthenelse{\equal{#1}{r}}{%
18   \renewcommand{\LWR@realscriptsalign}{\LWR@print@mbox{text-align:right} ; }%
19 }{}%
20 }
21
22 \DeclareDocumentCommand \textsubsuperscript {s 0{1} mm} {%
23 \LWR@setrealscriptsalign{#2}%
24 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
25 \textsuperscript{#4}\textsubscript{#3}%
26 }%

```

```

27 }
28
29 \DeclareDocumentCommand \textsupersubscript {s O{l} mm} {%
30 \LWR@setrealscriptsalign{#2}%
31 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
32 \textsubscript{#4}\textsuperscript{#3}%
33 }%
34 }

```

File 259 **l warp-refcheck.sty**

§ 355 Package **refcheck**

Pkg **refcheck** refcheck is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{refcheck}[2013/02/14]

```

2 \def\showrefnames{}
3 \def\norefnames{}
4 \def\showcitenames{}
5 \def\nocitenames{}
6 \def\setonmsgs{}
7 \def\setoffmsgs{}
8 \def\checkunlbld{}
9 \def\ignoreunlbld{}
10 \newcommand*\refcheckxrdoc[2][]{}

```

File 260 **l warp-register.sty**

§ 356 Package **register**

(Emulates or patches code by MATTHEW LOVELL.)

Pkg **register** register is patched for use by l warp.

for HTML output: 1 \LWR@ProvidesPackagePass{register}[2018/11/10]

```

2 \xpatchcmd{\register}
3   {\centering}
4   {\begin{center}\begin{ lateximage }[-register-\~\packagediagramname]\end{ lateximage }\end{center}}
5   {}
6   {\LWR@patcherror{\register}{\register}}
7
8 \xpatchcmd{\endregister}

```

```

9      {\leftskip}
10     {%
11         \end{lateximage}\end{center}%
12         \leftskip%
13     }%
14     {}%
15     {\LWR@patcherror{register}{endregister}}
16
17 \expandafter\xapptocmd\csname register*\endcsname
18   {\begin{center}\begin{lateximage}[(-register-\~\packagediagramname)]}
19   {}
20   {\LWR@patcherror{register}{register*}}
21
22 \expandafter\xpatchcmd\csname endregister*\endcsname
23   {\leftskip}
24   {%
25       \end{lateximage}\end{center}%
26       \leftskip%
27   }%
28   {}%
29   {\LWR@patcherror{register}{endregister*}}
30
31 \setlength{\regWidth}{5in}

```

File 261 **lwarf-reysize.sty**

§ 357 Package **reysize**

(Emulates or patches code by DONALD ARSENEAU, BERNIE COSELL, MATT SWIFT.)

Pkg **reysize** `reysize` is patched for use by `lwarf`.

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

`\reysize`, `\larger`, `\smaller`, and `\textscale` are ignored.

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

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

for HTML output: 1 `\LWR@ProvidesPackagePass{reysize}`

```

2 \let\LWR@origreysize\reysize
3 \LetLtxMacro\LWR@origlarger\larger

```

```

4 \LetLtxMacro{\LWR@origsmaller}{\smaller}
5 \let\relsize\relscale
6 \LetLtxMacro{\LWR@origtextlarger}{\textlarger}
7 \LetLtxMacro{\LWR@origtextsmaller}{\textsmaller}
8 \let\textscale\textscale
9
10 \appto{\LWR@restoreorigformatting}{%
11 \let\relsize{\LWR@origrelsize}%
12 \LetLtxMacro{\larger}{\LWR@origlarger}%
13 \LetLtxMacro{\smaller}{\LWR@origsmaller}%
14 \let\relscale{\LWR@relscale}%
15 \LetLtxMacro{\textlarger}{\LWR@origtextlarger}%
16 \LetLtxMacro{\textsmaller}{\LWR@origtextsmaller}%
17 \let\textscale{\LWR@textscale}%
18 }
19
20 \newcounter{LWR@relsizetemp}
21
22 \renewcommand*{\relsize}[1]{}
23 \renewcommand*{\larger}[1][]{}
24 \renewcommand*{\smaller}[1][]{}
25 \renewcommand*{\relscale}[1]{}
26
27 \renewcommand*{\textlarger}[2][1]{%
28 \setcounter{LWR@relsizetemp}{100+ (#1*20)}%
29 \InlineClass{font-size:\arabic{LWR@relsizetemp}\%}{textlarger}{#2}%
30 }
31
32 \renewcommand*{\textsmaller}[2][1]{%
33 \setcounter{LWR@relsizetemp}{100- (#1*20)}%
34 \InlineClass{font-size:\arabic{LWR@relsizetemp}\%}{textsmaller}{#2}%
35 }
36
37 \renewcommand*{\textscale}[2]{%
38 \setcounter{LWR@relsizetemp}{100*\real{#1}}%
39 \InlineClass{font-size:\arabic{LWR@relsizetemp}\%}{textscale}{#2}%
40 }

```

File 262 lwarf-repeatindex.sty

§ 358 Package **repeatindex**

Pkg repeatindex repeatindex is emulated for lwarf.

lwarf must be used with a special style file:

`\usepackage[makeindex,makeindexStyle={lwarf_repeatindex}]{lwarf}`

where l warp_repeatindex.ist may be copied from the following modified version of l warp.ist:

```

preamble
"\begin{theindex}
 \providecommand*\lettergroupDefault[1]{}
 \providecommand*\lettergroup[1]{%
   \par\textbf{\#1}\par
   \nopagebreak
 }
"
headings_flag 1
heading_prefix "
 \lettergroup{"
heading_suffix "}"
delim_0 ", \hyperindexref{"
delim_1 ", \hyperindexref{"
delim_2 ", \hyperindexref{"
delim_n "}, \hyperindexref{"
delim_r "} -- \hyperindexref{"
delim_t "}"

item_0 "\n \item ["

```

(The modifications are the `delim_0` and `item_0` entries.)

for HTML output: 1 \LWR@ProvidesPackageDrop{repeatindex}[2001/10/13]

In the l warp core, `\LWR@indexitem` is modified to accept the optional `\item` argument.

```

2 \RequirePackage{makeidx}
3 \def\entryprefix{\itshape}
4 \def\entrypostfix{\dots}

```

File 263 **l warp-resizegather.sty**

§ 359 Package **resizegather**

Pkg `resizegather` `resizegather` is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{resizegather}

```
2 \newcommand*{\resizegathersetup}[1]{}
```

File 264 **l warp-rmpage.sty**

§ 360 Package **rmpage**

Pkg rmpage rmpage is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{rmpage}

File 265 **l warp-romanbar.sty**

§ 361 Package **romanbar**

(Emulates or patches code by H.-MARTIN MÜNCH.)

Pkg romanbar romanbar is patched for use by l warp.

An inline class with an overline and underline is used.

for HTML output: 1 \LWR@ProvidesPackagePass{romanbar}

```
2 \DeclareRobustCommand{\Roman@bar}[1]{% #1 is in Roman, i.e. MMXII
3   \InlineClass[%
4     text-decoration: overline underline ;
5   ]{\romanbar}{#1}%
6 }
```

File 266 **l warp-romanbarpagenumber.sty**

§ 362 Package **romanbarpagenumber**

Pkg romanbarpagenumber romanbarpagenumber is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{romanbarpagenumber}

File 267 **l warp-rotating.sty**

§ 363 Package **rotating**

(Emulates or patches code by ROBIN FAIRBAIRNS, SEBASTIAN RAHTZ, LEONOR BARROCA.)

Pkg **rotating** rotating is emulated.

All rotations are ignored in HTML output.

for HTML output: 1 \LWR@ProvidesPackageDrop{rotating}

```
2 \LetLtxMacro{\sidewaystable}{\table}
3 \let\endsidewaystable\endtable
4
5 \LetLtxMacro{\sidewaysfigure}{\figure}
6 \let\endsidewaysfigure\endfigure
7
8 \newenvironment*{sideways}{}{}
9 \newenvironment*{turn}[1]{}{}
10 \newenvironment*{rotate}[1]{}{}
11 \NewDocumentCommand{\turnbox}{m +m}{#2}
12 \let\rotcaption\caption
13 \let\@makerotcaption\@makecaption
```

File 268 **l warp-rotfloat.sty**

§ 364 Package **rotfloat**

(Emulates or patches code by AXEL SOMMERFELDT.)

Pkg **rotfloat** rotfloat is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{rotfloat}[2004/01/04]
2
3 \RequirePackage{float}

\newfloat {\langle 1: type\rangle} {\langle 2: placement\rangle} {\langle 3: ext\rangle} [{\langle 4: within\rangle}]

Emulates the \newfloat command from the float package. Sideways floats are \let to the same as regular floats.

“placement” is ignored.

```

4 \RenewDocumentCommand{\newfloat}{m m m o}{%
5 \IfValueTF{#4}{%
6 {%
7     \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}%
8 }%
9 {%
10    \DeclareFloatingEnvironment[fileext=#3]{#1}%
11 }%
12 \csletcs{sideways}{#1}%
13 \csletcs{endsideways}{#1}{end#1}%

```

Remember the float style:

```

14 \csedef{\LWR@floatstyle@#1}{\LWR@floatstyle}%
15 \csedef{\LWR@floatstyle@sideways#1}{\LWR@floatstyle}%

```

`newfloat` package automatically creates the `\listof` command for new floats, but `float` does not, so remove `\listof` here in case it is manually created later:

```

16 \cslet{\listof#1s}{\relax}%
17 \cslet{\listof#1es}{\relax}%
18 \cslet{\listofsideways#1s}{\relax}%
19 \cslet{\listofsideways#1es}{\relax}%
20 }

```

File 269 **lwarp-rviewport.sty**

§ 365 Package **rviewport**

Pkg `rviewport` `rviewport` is honored inside a `lateximage`, and otherwise ignored for HTML output.

If `rviewport` is important for an image, enclose the image inside a `lateximage` environment.

for HTML output: 1 `\LWR@ProvidesPackagePass[rviewport][2011/08/27]`

2 `\define@key{igraph}{rviewport}{}%`

File 270 **l warp-savetrees.sty**

§ 366 Package **savetrees**

Pkg **savetrees** Emulated.

for HTML output: Discard all options for l warp-savetrees:

```
1 \LWR@ProvidesPackageDrop{savetrees}
```

File 271 **l warp-scalefnt.sty**

§ 367 Package **scalefnt**

(Emulates or patches code by D. CARLISLE.)

Pkg **scalefnt** **scalefnt** is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{scalefnt}
```

```
2 \DeclareRobustCommand\scalefont[1]{}

---


```

File 272 **l warp-schemata.sty**

§ 368 Package **schemata**

(Emulates or patches code by CHARLES P. SCHAUM.)

Pkg **schemata** **schemata** is patched for use by l warp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{schemata}
```

```
2 \LetLtxMacro\LWR@schemata@origschema\schema
3 \LetLtxMacro\LWR@schemata@origSchema\Schema
4
5 \renewcommand{\schema}[3][open]{%
6 \begin{lateximage}%
7 \LWR@print@normalsize
8 \LWR@schemata@origschema[#1]{#2}{#3}%
9 \end{lateximage}%
```

```
10 }
11
12 \renewcommand{\Schema}[5][open]{%
13 \begin{lateximage}%
14 \LWR@print@normalsize
15 \LWR@schemata@origSchema[#1]{#2}{#3}{#4}{#5}%
16 \end{lateximage}%
17 }
```

File 273 **l warp-scrextend.sty**

§ 369 Package **scrextend**

Pkg scrextend scrextend is emulated.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. \DeclareDocumentCommand is used to overwrite the koma-script definitions.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{scrextend}

2 \DeclareDocumentCommand{\setkomafont}{m m}{}
3 \DeclareDocumentCommand{\addkomafont}{m m}{}
4 \DeclareDocumentCommand{\usekomafont}{m}{}
5
6 \DeclareDocumentCommand{\usefontofkomafont}{m}{}
7 \DeclareDocumentCommand{\useencodingofkomafont}{m}{}
8 \DeclareDocumentCommand{\usesizeofkomafont}{m}{}
9 \DeclareDocumentCommand{\usefamilyofkomafont}{m}{}
10 \DeclareDocumentCommand{\useseriesofkomafont}{m}{}
11 \DeclareDocumentCommand{\useshapeofkomafont}{m}{}

12
13 \AtBeginDocument{
14 \let\LWR@maketitle\maketitle
15 \DeclareDocumentCommand{\maketitle}{o}{\LWR@maketitle}
16 }
17
18 \DeclareDocumentCommand{\extratitle}{m}{}
19 \DeclareDocumentCommand{\titlehead}{m}{}
20 \DeclareDocumentCommand{\subject}{m}{}
21 \DeclareDocumentCommand{\publishers}{m}{\published{#1}}
22 \DeclareDocumentCommand{\uppertitleback}{m}{}
23 \DeclareDocumentCommand{\lowertitleback}{m}{}
24 \DeclareDocumentCommand{\dedication}{m}{}

25
26 \DeclareDocumentCommand{\ifthispageodd}{m m}{#1}
27
```

```
28 \DeclareDocumentCommand{\titlepagestyle}{}{}
29
30 \DeclareDocumentCommand{\cleardoublepageusingstyle}{m}{}
31 \DeclareDocumentCommand{\cleardoubleemptypage}{}{}
32 \DeclareDocumentCommand{\cleardoubleplainpage}{}{}
33 \DeclareDocumentCommand{\cleardoublestandardpage}{}{}
34 \DeclareDocumentCommand{\cleardoubleoddpage}{}{}
35 \DeclareDocumentCommand{\cleardoubleoddpageusingstyle}{m}{}
36 \DeclareDocumentCommand{\cleardoubleoddemptypage}{}{}
37 \DeclareDocumentCommand{\cleardoubleoddplainpage}{}{}
38 \DeclareDocumentCommand{\cleardoubleoddstandardpage}{}{}
39 \DeclareDocumentCommand{\cleardoubleevenpage}{}{}
40 \DeclareDocumentCommand{\cleardoubleevenpageusingstyle}{m}{}
41 \DeclareDocumentCommand{\cleardoubleevenemptypage}{}{}
42 \DeclareDocumentCommand{\cleardoubleevenplainpage}{}{}
43 \DeclareDocumentCommand{\cleardoubleevenstandardpage}{}{}
44
45 \DeclareDocumentCommand{\multiplefootnoteseparator}{}{%
46   \begingroup\let\thefootnotemark\multfootsep\@makefnmark\endgroup
47 }
48
49 \DeclareDocumentCommand{\multfootsep}{}{,}
50
51 \DeclareDocumentCommand{\footref}{m}{%
52   \begingroup
53     \unrestored@protected@xdef\@thefnmark{\ref{\#1}}%
54   \endgroup
55   \@footnotemark
56 }
57
58 \DeclareDocumentCommand{\deffootnote}{o m m m}{}
59 \DeclareDocumentCommand{\deffootnotemark}{m}{}
60 \DeclareDocumentCommand{\setfootnoterule}{o m}{}
61 \DeclareDocumentCommand{\raggedfootnote}{}{}
62
63 \DeclareDocumentCommand{\dictum}{o m}{%
64 \begin{LWR@BlockClassWP}{\LWR@print@mbox{text-align:right}}{}{\dictum}}
65   #2
66   \IfValueT{\#1}{%
67     {%
68       \ifbool{FormatWP}{%
69         {\begin{BlockClass}[\LWR@print@mbox{border-top:} 1px solid gray]{dictumauthor}}%
70         {\begin{BlockClass}{dictumauthor}}%
71         \dictumauthorformat{\#1}%
72         \end{BlockClass}}%
73     }%
74   \end{LWR@BlockClassWP}}
75 }
76
77 \DeclareDocumentCommand{\dictumwidth}{}{}
```

```
78 \DeclareDocumentCommand{\dictumauthorformat}{m}{(#1)}
79 \DeclareDocumentCommand{\dictumrule}{(){}}
80 \DeclareDocumentCommand{\raggeddictum}{(){}}
81 \DeclareDocumentCommand{\raggeddictumtext}{(){}}
82 \DeclareDocumentCommand{\raggeddictumauthor}{(){}}
83
84 \DeclareDocumentEnvironment{labeling}{o m}
85 {%
86 \def\sc@septext{#1}%
87 \list{}{.}%
88 \let\makelabel\labelinglabel%
89 }
90 {
91 \endlist
92 }
93
94 \DeclareDocumentCommand{\labelinglabel}{m}{%
95 #1 \qquad \sc@septext%
96 }
97
98 \let\addmargin\relax
99 \let\endaddmargin\relax
100 \cslet{addmargin*}{\relax}
101 \cslet{endaddmargin*}{\relax}
102
103 \NewDocumentEnvironment{addmargin}{s O{} m}
104 {
105 \setlength{\LWR@templengthtwo}{#3}
106 \ifblank{#2}
107 {
108   \begin{BlockClass}[
109     \LWR@print@mbox{margin-left:\LWR@printlength{\LWR@templengthtwo}} ;
110     \LWR@print@mbox{margin-right:\LWR@printlength{\LWR@templengthtwo}}
111   ]{addmargin}
112 }
113 {
114   \setlength{\LWR@templengthone}{#2}
115   \begin{BlockClass}[
116     \LWR@print@mbox{margin-left:\LWR@printlength{\LWR@templengthone}} ;
117     \LWR@print@mbox{margin-right:\LWR@printlength{\LWR@templengthtwo}}
118   ]{addmargin}
119 }
120 }
121 {\end{BlockClass}}
```

Ref to create a starred environment:

<https://tex.stackexchange.com/questions/45401/>
use-the-s-star-argument-with-newdocumentenvironment

```

122
123 \ExplSyntaxOn
124 \cs_new:cpx {addmargin*} {\addmargin*}
125 \cs_new_eq:cN {\endaddmargin*} \endaddmargin
126 \ExplSyntaxOff
127
128 \DeclareDocumentCommand{\marginline}{m}{\marginpar{#1}}

```

File 274 **lwarf-scrhack.sty**

§ 370 Package **scrhack**

Pkg **scrhack** **scrhack** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{scrhack}

File 275 **lwarf-scrlayer.sty**

§ 371 Package **scrlayer**

(Emulates or patches code by MARKUS KOHM.)

Pkg **scrlayer** **scrlayer** is emulated.

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

for HTML output: 1 \LWR@ProvidesPackageDrop{scrlayer}

```

2 \newcommand*{\DeclareSectionNumberDepth}[2]{}
3 \newcommand*{\DeclareLayer}[2][]{}
4 \newcommand*{\DeclareNewLayer}[2][]{}
5 \newcommand*{\ProvideLayer}[2][]{}
6 \newcommand*{\RedeclareLayer}[2][]{}
7 \newcommand*{\ModifyLayer}[2][]{}
8 \newcommand*{\layerhalign}{}%
9 \newcommand*{\layervalign}{}%
10 \newcommand*{\layerxoffset}{}%
11 \newcommand*{\layeryoffset}{}%
12 \newcommand*{\layerwidth}{}%
13 \newcommand*{\layerheight}{}%
14 \providecommand*{\LenToUnit}[1]{\strip@pt\dimexpr#1*\p@/\unitlength}
15 \newcommand*{\putUL}[1]{}%
16 \newcommand*{\putUR}[1]{}%
17 \newcommand*{\putLL}[1]{}%

```

```
18 \newcommand*\{\putLR}[1]{}
19 \newcommand*\{\putC}[1]{}
20 \newcommand*\{\GetLayerContents}[1]{}
21 \newcommand{\IfLayerExists}[3]{#3}
22 \newcommand*\{\DestroyLayer}[1]{}
23 \newcommand*\{\layercontentsmeasure\}{}{}
24 \newcommand*\{\currentpagestyle\}{}{}
25 \newcommand*\{\BeforeSelectAnyPageStyle}[1]{}
26 \newcommand*\{\AfterSelectAnyPageStyle}[1]{}
27 \newcommand*\{\DeclarePageStyleAlias}[2]{}
28 \newcommand*\{\DeclareNewPageStyleAlias}[2]{}
29 \newcommand*\{\ProvidePageStyleAlias}[2]{}
30 \newcommand*\{\RedeclarePageStyleAlias}[2]{}
31 \newcommand*\{\DestroyPageStyleAlias}[1]{}
32 \newcommand*\{\GetRealPageStyle}[1]{}
33 \newcommand*\{\DeclarePageStyleByLayers}[3]{[]}{}
34 \newcommand*\{\DeclareNewPageStyleByLayers}[3]{[]}{}
35 \newcommand*\{\ProvidePageStyleByLayers}[3]{[]}{}
36 \newcommand*\{\RedeclarePageStyleByLayers}[3]{[]}{}
37 \NewDocumentCommand{\ForEachLayerOfPageStyle}{s m m}(){}
38 \newcommand*\{\AddLayersToPageStyle}[2]{}
39 \newcommand*\{\AddLayersAtBeginOfPageStyle}[2]{}
40 \newcommand*\{\AddLayersAtEndOfPageStyle}[2]{}
41 \newcommand*\{\RemoveLayersFromPageStyle}[2]{}
42 \newcommand*\{\AddLayersToPageStyleBeforeLayer}[3]{}
43 \newcommand*\{\AddLayersToPageStyleAfterLayer}[3]{}
44 \newcommand*\{\UnifyLayersAtPageStyle}[1]{}
45 \newcommand*\{\ModifyLayerPageStyleOptions}[2]{}
46 \newcommand*\{\AddToLayerPageStyleOptions}[2]{}
47 \newcommand{\IfLayerPageStyleExists}[3]{#3}
48 \newcommand{\IfRealLayerPageStyleExists}[3]{#3}
49 \newcommand{\IfLayerAtPageStyle}[4]{#4}
50 \newcommand{\IfSomeLayerAtPageStyle}[4]{#4}
51 \newcommand{\IfLayersAtPageStyle}[4]{#4}
52 \newcommand*\{\DestroyRealLayerPageStyle}[1]{}
53 @ifundefined{fooheight}{\newlength\fooheight}{}
54 \DeclareDocumentCommand{\automark}{s o m}){}
55 \DeclareDocumentCommand{\manualmark}{}{)}
56 \DeclareDocumentCommand{\MakeMarkcase}{m}{#1}

57 \newcommand{\partmarkformat}{}{}
58 \if@chapter
59 \newcommand{\chaptermarkformat}{}{}
60 \fi
61 \newcommand{\sectionmarkformat}{}{}
62 \DeclareDocumentCommand{\GenericMarkFormat}{m}{}

63 \newcommand*\{@mkleft}[1]{}
64 \newcommand*\{@mkright}[1]{}
```

```

65 \newcommand*{\@mkdouble}{1}{}
66 \newcommand*{\@mkboth}{2}{}
67 \newcommand*{\scrlayerInitInterface}{1} []
68 \newcommand{\scrlayerAddToInterface}{3} []
69 \newcommand{\scrlayerAddCsToInterface}{3} []
70 \newcommand{\scrlayerOnAutoRemoveInterface}{2} []

```

File 276 **l warp-scrlayer-notecolumn.sty**

§ 372 Package **scrlayer-notecolumn**

(Emulates or patches code by MARKUS KOHM.)

Pkg scrlayer-notecolumn scrlayer-notecolumn is emulated.

⚠ Not fully tested! Please send bug reports!

for HTML output:

```

1 \LWR@ProvidesPackageDrop{scrlayer-notecolumn}

2 \newcommand*{\DeclareNoteColumn}{2} []
3 \newcommand*{\DeclareNewNoteColumn}{2} []
4 \newcommand*{\ProvideNoteColumn}{2} []
5 \newcommand*{\RedeclareNoteColumn}{2} []
6 \NewDocumentCommand{\makernote}{s o m}{\marginpar{#3}}
7 \newcommand*{\syncwithnotecolumn}{1} []
8 \newcommand*{\syncwithnotecolumns}{1} []
9 \newcommand*{\clearnotecolumn}{1} []
10 \newcommand*{\clearnotecolumns}{1} []

```

File 277 **l warp-scrlayer-scrpage.sty**

§ 373 Package **scrlayer-scrpage**

(Emulates or patches code by MARKUS KOHM.)

Pkg scrlayer-scrpage scrlayer-scrpage is emulated.

⚠ Not fully tested! Please send bug reports!

for HTML output:

```

1 \LWR@ProvidesPackageDrop{scrlayer-scrpage}

2 \@ifundefined{fooheight}{\newlength\fooheight}{}
3 \NewDocumentCommand{\lehead}{s o m}{}
4 \NewDocumentCommand{\cehead}{s o m}{}

```

```

5 \NewDocumentCommand{\rehead}{s o m}{}
6 \NewDocumentCommand{\lohead}{s o m}{}
7 \NewDocumentCommand{\cohead}{s o m}{}
8 \NewDocumentCommand{\rohead}{s o m}{}
9 \NewDocumentCommand{\lefoot}{s o m}{}
10 \NewDocumentCommand{\cefoot}{s o m}{}
11 \NewDocumentCommand{\refoot}{s o m}{}
12 \NewDocumentCommand{\lofoot}{s o m}{}
13 \NewDocumentCommand{\cofoot}{s o m}{}
14 \NewDocumentCommand{\rofoot}{s o m}{}
15 \NewDocumentCommand{\ohead}{s o m}{}
16 \NewDocumentCommand{\chead}{s o m}{}
17 \NewDocumentCommand{\ihead}{s o m}{}
18 \NewDocumentCommand{\ofoot}{s o m}{}
19 \NewDocumentCommand{\cfoot}{s o m}{}
20 \NewDocumentCommand{\ifoot}{s o m}{}
21 \DeclareDocumentCommand{\MakeMarkcase}{m}{#1}
22 \newcommand*\defpairofpagestyles[3][]{}
23 \newcommand*\newpairofpagestyles[3][]{}
24 \newcommand*\renewpairofpagestyles[3][]{}
25 \newcommand*\providepairofpagestyles[3][]{}

26 \newcommand*\clearmainofpairofpagestyles(){}
27 \newcommand*\clearplainofpairofpagestyles(){}
28 \newcommand*\clearpairofpagestyles(){}
29 \newcommand*\clearscrheadings(){}
30 \newcommand*\clearscrheadfoot(){}
31 \newcommand*\clearscrplain(){}

32 \NewDocumentCommand{\deftriplepagestyle}{m o o m m m m m}{}
33 \NewDocumentCommand{\newtriplepagestyle}{m o o m m m m m}{}
34 \NewDocumentCommand{\renewtriplepagestyle}{m o o m m m m m}{}
35 \NewDocumentCommand{\providetriplepagestyle}{m o o m m m m m}{}
36 \newcommand*\defpagestyle[3]({})
37 \newcommand*\newpagestyle[3]({})
38 \newcommand*\providepagestyle[3]({})
39 \newcommand*\renewpagestyle[3]({})

```

File 278 **lwarf-scrpage2.sty**

§ 374 Package **scrpage2**

(Emulates or patches code by MARKUS KOHM.)

Pkg scrpage2 scrpage2 is emulated.

⚠ Not fully tested! Please send bug reports!

for HTML output:

```
1 \LWR@ProvidesPackageDrop{scrrpage2}

2 \@ifundefined{footheight}{\newlength\footheight}{}
3 \NewDocumentCommand{\lehead}{o m}{}
4 \NewDocumentCommand{\cehead}{o m}{}
5 \NewDocumentCommand{\rehead}{o m}{}
6 \NewDocumentCommand{\lohead}{o m}{}
7 \NewDocumentCommand{\cohead}{o m}{}
8 \NewDocumentCommand{\rohead}{o m}{}
9 \NewDocumentCommand{\lefoot}{o m}{}
10 \NewDocumentCommand{\cefoot}{o m}{}
11 \NewDocumentCommand{\refoot}{o m}{}
12 \NewDocumentCommand{\lofoot}{o m}{}
13 \NewDocumentCommand{\cofoot}{o m}{}
14 \NewDocumentCommand{\rofoot}{o m}{}
15 \NewDocumentCommand{\ohead}{o m}{}
16 \NewDocumentCommand{\chead}{o m}{}
17 \NewDocumentCommand{\ihead}{o m}{}
18 \NewDocumentCommand{\ofoot}{o m}{}
19 \NewDocumentCommand{\cfoot}{o m}{}
20 \NewDocumentCommand{\ifoot}{o m}{}
21 \DeclareDocumentCommand{\automark}{o m}{}
22 \DeclareDocumentCommand{\manualmark}{(){}}
23 \DeclareDocumentCommand{\MakeMarkcase}{m}{#1}
24 \NewDocumentCommand{\deftripstyle}{m o o m m m m m m}{}
25 \NewDocumentCommand{\defpagestyle}{s m m m}{}
26 \NewDocumentCommand{\newpagestyle}{s m m m}{}
27 \NewDocumentCommand{\renewpagestyle}{s m m m}{}
28 \NewDocumentCommand{\providepagestyle}{s m m m}{}
29 \newcommand{\partmarkformat}{}
30 \if@chapter
31 \newcommand{\chaptermarkformat}{}
32 \fi
33 \newcommand{\sectionmarkformat}{}
34 \newcommand{\subsectionmarkformat}{}
35 \newcommand{\subsubsectionmarkformat}{}
36 \newcommand{\paragraphmarkformat}{}
37 \newcommand{\ subparagraphmarkformat}{}
38
39 \newcommand*{\clearscrheadings}{}
40 \newcommand*{\clearscrheadfoot}{}
41 \newcommand*{\clearscrplain}{}
```

File 279 **l warp-section.sty**

§ 375 Package **section**

Pkg **section** **section** is ignored.

(Emulates or patches code by OLIVER PRETZEL.)

for HTML output:

```
1 \LWR@ProvidesPackageDrop{section}

2 \ifx\chapter\undefined
3   \def\chsizet{\Large}\def\hdsizet{\huge}\else
4   \def\chsizet{\huge}\def\hdsizet{\Huge}
5 \fi
6 \let\ttsize\LARGE
7 \let\ausize\large
8 \let\dasize\large
9 \let\secsize\Large
10 \let\subsize\large
11 \let\hdpos\raggedright
12 \newcounter{hddepth}
13 \let\fpind\relax
14 \def\ttfnt{}
15 \def\hdfnt{}
16 \def\fefnt{}
17 \def\thfnt{}
18 \def\pgfnt{}
19 \def\hmkfnt{}
20 \let\mkcse\uppercase
21 \def\hddot{}
22 \def\cpdot{:}
23 \def\nmdot{}
24 \ifx\secindent\undefined
25   \newdimen\secindent
26   \newskip\secpreskp
27   \newskip\secpstskp
28   \newdimen\subindent
29   \newskip\subpreskp
30   \newskip\subpstskp
31   \newskip\parpstskp
32   \newcount\c@hddepth
33 \fi
```

File 280 **l warp-sectionbreak.sty**

§ 376 Package **sectionbreak**

(Emulates or patches code by MICHAL HOFTICH.)

Pkg sectionbreak sectionbreak is patched for use by l warp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{sectionbreak}

2 \renewcommand\asterism{\HTMLunicode{2042}}
3
4 \renewcommand\pre@sectionbreak{}
5 \renewcommand\post@sectionbreak{}
6
7 \renewcommand\print@sectionbreak[1]{%
8 \begin{center}
9 #1
10 \end{center}
11 }
12
```

File 281 **l warp-sectsty.sty**

§ 377 Package **sectsty**

(Emulates or patches code by ROWLAND McDONNELL.)

Pkg sectsty sectsty is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{sectsty}

2 \newcommand*\partfont           [1] {}
3 \newcommand*\partnumberfont     [1] {}
4 \newcommand*\parttitlefont      [1] {}
5 \newcommand*\chapterfont        [1] {}
6 \newcommand*\chapternumberfont  [1] {}
7 \newcommand*\chaptertitlefont   [1] {}
8 \newcommand*\sectionfont        [1] {}
9 \newcommand*\subsectionfont     [1] {}
10 \newcommand*\subsubsectionfont  [1] {}
11 \newcommand*\paragraphfont     [1] {}
```

```

12 \newcommand*{\subparagraphfont}[1] {}
13 \newcommand*{\minisecfont}[1] {}
14 \newcommand*{\allsectionsfont}[1] {}
15 \newcommand{\nohang}{}{}
```

\sectionrule is only to be used in *font commands, thus it is ignored.

```

16 \newcommand*{\sectionrule}[5] {}
17
18 \def\ulemheading#1#2{}
```

File 282 **lwarf-semantic-markup.sty**

§ 378 Package **semantic-markup**

(Emulates or patches code by ANDREW A. CASHNER.)

Pkg **semantic-markup** semantic-markup is patched for use by lwarf.



If using the endnotes option, add \theendnotes where desired.

for HTML output: 1 \LWR@ProvidesPackagePass{semantic-markup}[2018/05/21]

The endnotes must be printed by the user before the end of the document, since the end is after the HTML footer, etc.

```

2 \ifendnotes
3 \RenewDocumentCommand{\SetupEndnotes}{}{%
4     \let\footnote=\endnote
5 %     \AtEndDocument{\DoBeforeEndnotes{\EndnoteFont\theendnotes}}%
6 }
7 \fi
```

HTML unicode characters from musicography are used.

```

8 \RequirePackage{musicography}
9
10 \let\f1\musFlat
11 \let\sh\musSharp
12 \let\na\musNatural
```

The \musfig is placed inside a hashed image, with a simple alt tag.

```

13 \RequirePackage{amsmath}
14
15 \RenewDocumentCommand{\musfig}{ m m }{%
```

```

16      \LWR@subsingle dollar*%
17      {#1/#2}%
18      {musfig}%
19      {%
20          \LWR@origensuredmath{%
21              \genfrac{}{}{0pt}{1}{\text{#1}}{\text{#2}}%
22          }%
23      }%
24 }

```

The `\meter` is taken from musicography, and becomes a hashed image with a simple alt tag.

```

25 \RenewDocumentCommand{\meter}{ m m }{%
26     \musMeter{#1}{#2}%
27 }

```

File 283 **l warp-setspace.sty**

§ 379 Package **setspace**

(Emulates or patches code by ROBIN FAIRBAIRNS.)

Pkg **setspace** `setspace` is not used during HTML conversion.

Discard all options for `l warp-setspace`:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{setspace}
2
3 \newcommand*{\setstretch}[1]{}
4 \newcommand*{\SetSinglespace}[1]{}
5 \newcommand*{\singespacing}{}
6 \newcommand*{\onehalfspacing}{}
7 \newcommand*{\doublespacing}{}
8
9 \newenvironment*{singlespace}{}
10 {
11 \LWR@forcenewpage
12 \BlockClass{singlespace}
13 }
14 {\endBlockClass}
15
16 \newenvironment*{singlespace*}{}
17 {
18 \LWR@forcenewpage
19 \BlockClass{singlespace}
20 }

```

```
21 {\endBlockClass}
22
23 \newenvironment*{spacing}[1]{
24
25 }{
26
27 }
28
29 \newenvironment*{onehalfspace}%
30 {
31 \LWR@forcenewpage
32 \BlockClass{onehalfspace}
33 }
34 {\endBlockClass}
35
36 \newenvironment*{doublespace}%
37 {
38 \LWR@forcenewpage
39 \BlockClass{doublespace}
40 }
41 {\endBlockClass}
```

File 284 **l warp-shadow.sty**

§ 380 Package **shadow**

(Emulates or patches code by MAURO ORLANDINI.)

Pkg shadow shadow is emulated.

for HTML output: Discard all options for l warp-shadow:

```
1 \LWR@ProvidesPackageDrop{shadow}

2 \newdimen\sboxsep
3 \newdimen\sboxrule
4 \newdimen\sdim
5
6 \newcommand{\shabox}[1]{%
7 \InlineClass{shabox}{#1}%
8 }
```

File 285 **l warp-showidx.sty**

§ 381 Package **showidx**

Pkg **showidx** **showidx** is ignored.

for HTML output: Discard all options for l warp-showidx:

1 \LWR@ProvidesPackageDrop{showidx}

\@wrindex is redefined \AtBeginDocument by the l warp core.

File 286 **l warp-showkeys.sty**

§ 382 Package **showkeys**

(Emulates or patches code by DAVID CARLISLE, MORTEN HØGHOLM.)

Pkg **showkeys** **showkeys** is ignored.

for HTML output: Discard all options for l warp-showkeys:

1 \LWR@ProvidesPackageDrop{showkeys}

2 \NewDocumentCommand{\showkeys}{s}{}

File 287 **l warp-sidecap.sty**

§ 383 Package **sidecap**

(Emulates or patches code by ROLF NIEPRASCHK, HUBERT GÄSSLEIN.)

Pkg **sidecap** **sidecap** is emulated.

for HTML output: Discard all options for l warp-sidecap.

1 \LWR@ProvidesPackageDrop{sidecap}

See:

<http://tex.stackexchange.com/questions/45401/>

[use-the-s-star-argument-with-newdocumentenvironment](#)
 regarding the creation of starred environments with xparse.

```

2 \NewDocumentEnvironment{SCtable}{soo}
3 {\IfValueTF{#3}{\table[#3]}{\table}}
4 {\endtable}
5
6 \ExplSyntaxOn
7 \cs_new:cpx {SCtable*} {\SCtable*}
8 \cs_new_eq:cN {\endSCtable*} \endSCtable
9 \ExplSyntaxOff
10
11
12 \NewDocumentEnvironment{SCfigure}{soo}
13 {\IfValueTF{#3}{\figure[#3]}{\figure}}
14 {\endfigure}
15
16 \ExplSyntaxOn
17 \cs_new:cpx {SCfigure*} {\SCfigure*}
18 \cs_new_eq:cN {\endSCfigure*} \endSCfigure
19 \ExplSyntaxOff
20
21
22 \newenvironment*{wide}{}{}

```

File 288 **l warp-sidenotes.sty**

§ 384 Package **sidenotes**

(Emulates or patches code by ANDY THOMAS, OLIVER SCHEBAUM.)

Pkg **sidenotes** Patched for l warp.

for HTML output: Load the original package:

```
1 \LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with l warp:

```

\sidenote * [<entry>] [<offset>] {<text>}
2 \RenewDocumentCommand \sidenote {s o o +m}
3 {
4   \LWR@stoppars
5   \begingroup
6   \captionsetup{style=sidenote}
7   \IfBooleanTF{#1}

```

```

8   { % starred
9     \begin{BlockClass}[border:none ; box-shadow:none]{marginblock}
10    \caption*{#4}
11    \end{BlockClass}
12  }
13 { % unstarred
14 \IfNoValueOrEmptyTF{#2}
15   {\def\@sidenotes@sidecaption@tof{#4}}
16   {\def\@sidenotes@sidecaption@tof{#2}}
17   \begin{BlockClass}[border:none ; box-shadow:none]{marginblock}
18   \caption[\@sidenotes@sidecaption@tof]{#4}
19   \end{BlockClass}
20 }
21 \endgroup
22 \LWR@startpars
23 }
```

Borrowed from the lwarp version of keyfloat:

```

24 \NewDocumentEnvironment{KFLTsidenotes@marginfloat}{O{-1.2ex} m}
25 {%
26 \LWR@BlockClassWP{float:right; width:2in; margin:10pt}{}{marginblock}%
27 \captionsetup{type=#2}%
28 }
29 {%
30 \endLWR@BlockClassWP%
31 }
32
33 \RenewDocumentEnvironment{marginfigure}{o}
34  {\begin{KFLTsidenotes@marginfloat}{figure}}
35  {\end{KFLTsidenotes@marginfloat}}
36
37 \RenewDocumentEnvironment{margintable}{o}
38  {\begin{KFLTsidenotes@marginfloat}{table}}
39  {\end{KFLTsidenotes@marginfloat}}
```

The following were changed by sidenotes, and now are reset back to their lwarp-supported originals:

Restoring the definition from the L^AT_EX 2_E article.cls source:

```

40 \renewenvironment{figure*}
41           {@dblfloat{figure}}
42           {\end@dblfloat}
43
44 \renewenvironment{table*}
45           {@dblfloat{table}}
46           {\end@dblfloat}
```

File 289 **l warp-SIunits.sty**

§ 385 Package **SIunits**

(Emulates or patches code by MARCEL HELDOORN.)

Pkg **SIunits** SIunits is patched for use by l warp.

It is recommended to use `\unit` where possible, which combines the entire expression into a single `lateximage`, and adds the `alt` tag containing the L^AT_EX code, allowing for copy/paste. When units are used outside of the `\unit` macro, each unit macro will have its own `lateximage`, and each will have the `alt` tag set to “`<\mathimagedata>`”, which defaults to “`(math image)`”.

for HTML output: 1 \LWR@ProvidesPackagePass{SIunits}

Patched for copy/paste with the HTML alt tag:

```

2 \DeclareRobustCommand{\unit}[2]{%
3 \@inunitcommandtrue%    original
4 \LWR@subsingle$%    l warp
5 % alt tag
6     \textbackslash{}unit\{\LWR@HTMLsanitize{\#1}\}%
7     \{ \LWR@HTMLsanitize{\#2}\}% extra space
8 }%
9 {SIunits}% add'l hashing
10 {%
11     \LWR@origensuredmath%    l warp modification
12     \SI@fstyle{\#1\@qsk\period@active{\#2}}%    original
13     }%
14 }% contents
15 \@inunitcommandfalse%    original
16 }
```

File 290 **l warp-siunitx.sty**

§ 386 Package **siunitx**

(Emulates or patches code by JOSEPH WRIGHT.)

Pkg **siunitx** siunitx is patched for use by l warp.

fractions Due to `pdftotext` limitations, fraction output is replaced by symbol output for

per-mode and quotient-mode.

- ⚠ **math mode required** Some units will require that the expression be placed inside math mode.

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

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

```
for HTML output: 1 \RequirePackage{xcolor}%
2 % for \convertcolorspec
3 \LWR@ProvidesPackagePass{siunitx}

4 \AtBeginDocument{%
5   % in case textcomp was not loaded
6   \DeclareSIUnit\bohr{\textit{a}\textsubscript{0}}
7   \DeclareSIUnit\clight{\textit{c}\textsubscript{0}}
8   \DeclareSIUnit\elementarycharge{\textit{e}}
9   \DeclareSIUnit\electronmass{\textit{m}\textsubscript{e}}
10  \DeclareSIUnit\hartree{\textit{E}\textsubscript{h}}
11  \DeclareSIUnit\planckbar{\LWR@siunitx@textplanckbar}
12 }% AtBeginDocument
```

`\@ensuredmath` is not supported inside an `\hbox`, so it must temporarily be restored to its original. Similar for `\mbox`. SVG math is created explicitly when necessary, using `\LWR@subsingle$`.

```
12
13 \ExplSyntaxOn
14 %
```

Modified to set set HTML `\textcolor` if not black:

```
15 \cs_undefine:N \__siunitx_print_aux:
16 \cs_new_protected:Npn \__siunitx_print_aux:
17 {
18   \text
19   {
20     \__siunitx_ensure_ltr:n
21     {
22       \color@begingroup
23       \__siunitx_print_color:
24       \__siunitx_font_shape:
25       \__siunitx_font_weight:
26       \use:c
27       {
```

```
28          @@_ \l_siunitx_print_type_tl -
29          text \l_siunitx_font_family_tl :
30      }
31      \bool_if:NTF \l_siunitx_font_math_mode_bool
32      { \l_siunitx_print_math: }
33      {
34          \LWR@findcurrenttextcolor% lwarp
35          \ifdefstring{\LWR@tempcolor}{000000}% lwarp
36          {\l_siunitx_print_text:}% lwarp
37          {% lwarp
38              \LWR@textcurrentcolor{%
39                  \l_siunitx_print_text:
40              }%
41          }%
42      }
43      \color@endgroup
44  }
45 }
46 }
47
48 \cs_undefine:N \l_siunitx_set_math_fam:n
49 \cs_new_protected:Npn \l_siunitx_set_math_fam:n #1 {
50     \int_new:c { c_siunitx_math #1 _int }
51     \group_begin:% lwarp
52         \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
53         \LetLtxMacro\mbox\LWR@print@mbox% lwarp
54         \hbox_set:Nn \l_siunitx_tmp_box
55         {
56             \ensuremath
57             {
58                 \use:c { math #1 }
59             }
60             {
61                 \int_gset:cn { c_siunitx_math #1 _int } { \fam }
62             }
63         }
64     }
65     \group_end:% lwarp
66 }
67
68 \cs_undefine:N \l_siunitx_combined_output:n
69 \cs_new_protected:Npn \l_siunitx_combined_output:n #1 {
70     \group_begin:% lwarp
71     \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
72     \LetLtxMacro\mbox\LWR@print@mbox% lwarp
73     \bool_if:NTF \l_siunitx_number_parse_bool
74     {
75         \tl_clear:N \l_siunitx_number_out_tl
76         \bool_set_false:N \l_siunitx_number_compound_bool
77         \l_siunitx_number_output_parse:n {#1}
```

```

78      }
79      {
80      \__siunitx_unit_output_pre_print:

```

For parse-numbers=false:

```

81 \%      \__siunitx_print:nn { number } { \ensuremath {\#1} } }
82      \LWR@subsingle dollar{\% lwarp
83      \textbackslash ( \LWR@HTMLsanitize{\#1} \textbackslash ) \% lwarp
84      \{ siunitx \} \% 
85      \__siunitx_print:nn { number } { %
86      \LWR@origensuredmath{\#1} %
87      } %
88      } \% lwarp

89      \__siunitx_unit_output_print:
90      }
91      \group_end: \% lwarp
92 }
93 %

```

For quotients, the fraction code is replaced by the symbol code:

```

94 \cs_undefine:N \__siunitx_number_output_quotient_fraction:
95 \cs_new_protected:Npn \__siunitx_number_output_quotient_fraction: {
96   \bool_set_true:N \l__siunitx_number_compound_bool
97   \__siunitx_number_output_quotient_aux_i:
98   \tl_set_eq:NN \l__siunitx_number_out_tl
99   \l__siunitx_number_numerator_tl
100  \tl_put_right:NV \l__siunitx_number_out_tl \l__siunitx_output_quotient_tl
101  \tl_put_right:NV \l__siunitx_number_out_tl
102  \l__siunitx_number_denominator_tl
103  \__siunitx_number_output_single_aux:
104 }

```

For units, the fraction code is replaced by the symbol code:

```

105 \cs_undefine:N \__siunitx_unit_format_fraction_fraction:
106 \cs_new_protected:Npn \__siunitx_unit_format_fraction_fraction: {
107   \__siunitx_unit_format_fraction_symbol_aux:
108   \int_compare:nNnT { \l__siunitx_unit_denominator_int } > { 1 }
109   {
110     \bool_if:NT \l__siunitx_unit_denominator_bracket_bool
111     {
112       \tl_put_left:NV \l__siunitx_unit_denominator_tl \l__siunitx_bracket_open_tl
113       \tl_put_right:NV \l__siunitx_unit_denominator_tl \l__siunitx_bracket_close_tl
114     }
115   }
116   \tl_set_eq:NN \l__siunitx_unit_tl \l__siunitx_unit_numerator_tl

```

```
117 \tl_put_right:NV \l__siunitx_unit_tl \l__siunitx_per_symbol_tl
118 \tl_put_right:NV \l__siunitx_unit_tl \l__siunitx_unit_denominator_tl
119 }

120 \cs_undefine:N \__siunitx_angle_print_astronomy_aux:
121 \cs_new_protected:Npn \__siunitx_angle_print_astronomy_aux: {
122   \prop_get:NnNT \l__siunitx_number_out_prop { mantissa-integer }
123   \l__siunitx_tmpa_tl
124   { \__siunitx_print:nV { number } \l__siunitx_tmpa_tl }
125   \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
126   {\% lateximage
127     \hbox_set:Nn \l__siunitx_angle_marker_box
128     {
129       \__siunitx_print:nn { number } { { \l__siunitx_output_decimal_tl } }
130     }
131     \hbox_set:Nn \l__siunitx_angle_unit_box
132     {
133       \__siunitx_print:nV { unit } \l__siunitx_unit_tl
134       \skip_horizontal:n { -\scriptspace }
135     }
136   \__siunitx_angle_print_astronomy_aux:n { marker }
137   \__siunitx_angle_print_astronomy_aux:n { unit }
138   \hbox_set:Nn \l__siunitx_angle_marker_box
139   {
140     \box_use:N \l__siunitx_angle_marker_box
141     \box_use:N \l__siunitx_angle_unit_box
142   }
143 \dim_compare:nNnTF
144   { \l__siunitx_angle_marker_dim } > { \l__siunitx_angle_unit_dim }
145   { \__siunitx_angle_print_astronomy_marker: }
146   { \__siunitx_angle_print_astronomy_unit: }
147 }%
148 {\% not a lateximage
149   \__siunitx_print:nV { unit } \l__siunitx_unit_tl
150   \__siunitx_print:nn { number } { { \l__siunitx_output_decimal_tl } }
151 }%
152 {\% not a lateximage
153   \prop_get:NnNT \l__siunitx_number_out_prop { mantissa-decimal }
154   \l__siunitx_tmpa_tl
155   { \__siunitx_print:nV { number } \l__siunitx_tmpa_tl }
156 }

156 \RenewDocumentCommand \num { o m } {
157   \leavevmode
158   \group_begin: \% lwarp
159   \LetLtxMacro \ensuredmath \LWR@origensuredmath \% lwarp
160   \LetLtxMacro \mbox \LWR@print@mbox \% lwarp
161   \bool_set_false:N \l__siunitx_font_set_bool
162   \IfNoValueF {#1}
163   { \keys_set:nn { siunitx } {#1} }
```

```
164     \__siunitx_number_output:n {#2}
165     \group_end:% lwarp
166 }
167
168 \RenewDocumentCommand \numrange { o m m } {
169     \leavevmode
170     \group_begin:% lwarp
171         \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
172         \LetLtxMacro\mbox\LWR@print@mbox% lwarp
173         \bool_set_false:N \l__siunitx_font_set_bool
174         \IfNoValueF {#1}
175             { \keys_set:nn { siunitx } {#1} }
176             \__siunitx_range_numbers:nn {#2} {#3}
177     \group_end:% lwarp
178 }
179
180 \RenewDocumentCommand \ang { o > { \SplitArgument { 2 } { ; } } m } {
181     \group_begin:% lwarp
182         \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
183         \LetLtxMacro\mbox\LWR@print@mbox% lwarp
184         \IfNoValueF {#1}
185             { \keys_set:nn { siunitx } {#1} }
186             \__siunitx_angle_output:nnn #2
187     \group_end:% lwarp
188 }
189
190 \RenewDocumentCommand \si { o m } {
191     \leavevmode
192     \group_begin:% lwarp
193         \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
194         \LetLtxMacro\mbox\LWR@print@mbox% lwarp
195         \bool_set_false:N \l__siunitx_font_set_bool
196         \IfNoValueTF {#1}
197             { \__siunitx_unit_output:nn {#2} { } }
198             {
199                 \keys_set:nn { siunitx } {#1}
200                 \__siunitx_unit_output:nn {#2} {#1}
201             }
202     \group_end:% lwarp
203 }
204
205
206 \RenewDocumentCommand{\SIrange}{o m m m}
207 {%
208     \leavevmode
209     \group_begin:% lwarp
210         \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
211         \LetLtxMacro\mbox\LWR@print@mbox% lwarp
212         \bool_set_false:N \l__siunitx_font_set_bool
213         \IfNoValueTF {#1}
```

```

214      { \_siunitx_range_unit:nnnn {#4} { } {#2} {#3} }
215      {
216          \keys_set:nn { siunitx } {#1}
217          \_siunitx_range_unit:nnnn {#4} {#1} {#2} {#3}
218      }
219      \group_end: \% l warp
220 }
221
222 \ExplSyntaxOff

```

File 291 **l warp-soul.sty**

§ 387 Package **soul**

(Emulates or patches code by MELCHIOR FRANZ.)

Pkg **soul** Emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{soul}[2003/11/17]
2 \RequirePackage{xcolor}\% for \convertcolorspec

```

Storage for the colors to use:

```

3 \newcommand*\{\LWR@soululcolor\}{}
4
5 \newcommand*\{\LWR@soulstcolor\}{}
6
7 \% \definecolor{\LWR@soulhlcolordefault}{HTML}{F8E800}
8 \% \newcommand*\{\LWR@soulhlcolor\}{\LWR@soulhlcolordefault}
9 \newcommand*\{\LWR@soulhlcolor\}{}

```

\so {\langle text\rangle}

Basic markup with css:

```

10 \newcommand{\so}[1]\%
11 \InlineClass(letter-spacing:.2ex){letterspacing}{#1}\%
12 }

```

\caps {\langle text\rangle}

```

13 \newcommand{\caps}[1]\%
14     \InlineClass%
15         (font-variant:small-caps;letter-spacing:.1ex)\%
16         {capsspacing}{#1}\%
17 }

```

```
\LWR@soulcolor {<text>} {<color>} {<class>} {<colorstyle>} {<FormatWPstyle>}

Add colors if not empty:

18 \newcommand{\LWR@soulcolor}[5]{%
19 \ifcsempty{#2}{%
20 {%
21     \InlineClass{#5}{#3}{#1}{%
22 }{%
23 {%
24     \convertcolorspec{named}{\@nameuse{#2}}{HTML}\LWR@tempcolor{%
25         \LWR@htmlspanclass[#5;#4:\LWR@origpound\LWR@tempcolor]{#3}{#1}{%
26 }{%
27 }

28 \newcommand{\ul}[1]{%
29 \LWR@soulcolor{#1}{\LWR@soululcolor}{\uline}{text-decoration-color}{%
30     {text-decoration:underline; text-decoration-skip: auto;}{%
31 }{%
32
33 \newcommand{\st}[1]{%
34 \LWR@soulcolor{#1}{\LWR@soulstcolor}{\sout}{text-decoration-color}{%
35     {text-decoration:line-through}{%
36 }{%
37
38 \newcommand{\hl}[1]{%
39 \LWR@soulcolor{#1}{\LWR@soulhlc}{highlight}{background-color}{%
40     {background:\LWR@origpound{}F8E800}{%
41 }
```

Nullified:

```
42 \newcommand*\soulaccent}[1]{}{%
43 \newcommand*\soulregister}[2]{}{%
44 \newcommand\sloppyword}[1]{#1}{%
45 \newcommand*\sodef}[5]{\DeclareRobustCommand*#1[1]{\so{##1}}}{%
46 \newcommand*\resetso}{}{%
47 \newcommand*\capsdef}[5]{}{%
48 \newcommand*\capsreset}{}{%
49 \newcommand*\capssave}[1]{}{%
50 \newcommand*\capsselect}[1]{}{%
51 \newcommand*\setul}[2]{}{%
52 \newcommand*\resetul}{}{%
53 \newcommand*\setuldepth}[1]{}{%
54 \newcommand*\setuloverlap}[1]{}{}
```

Set colors:

```
55 \newcommand*\setulcolor}[1]{\renewcommand{\LWR@soululcolor}{#1}}
```

```
56 \newcommand*{\setstcolor}[1]{\renewcommand{\LWR@soulstcolor}{#1}}
57 \newcommand*{\sethlcolor}[1]{\renewcommand{\LWR@soulhlcolor}{#1}}
```

Long versions of the user-level macros:

```
58 \let\textso\soulstcolor
59 \let\textul\soulhlcolor
60 \let\texthl\soulhlcolor
61 \let\textcaps\soulstcolor
```

File 292 **l warp-soulpos.sty**

§ 388 Package **soulpos**

(Emulates or patches code by JAVIER BEZOS.)

Pkg **soulpos** **soulpos** is emulated.

for HTML output:

```
1 \RequirePackage{soul}
2 \RequirePackage{soulutf8}
3 \LWR@ProvidesPackageDrop{soulpos}

4 \NewDocumentCommand{\ulposdef}{m o m}{}%
5
6 \newdimen\ulwidth
7
8 \newcommand\ifulstarttype[1]{%
9 \expandafter\@secondoftwo%
10 }
11
12 \newcommand\ifulenctype[1]{%
13 \expandafter\@secondoftwo%
14 }
15
16 \newcommand{\ulstarttype}{0}
17 \newcommand{\ulenctype}{0}
18 \newcommand{\ulpostolerance}{0}%
```

File 293 **l warp-soulutf8.sty**

§ 389 Package **soulutf8**

Pkg **soulutf8** **soulutf8** is emulated.

lwarf's HTML output naturally supports UTF-8 encoding.

for HTML output: 1 \LWR@ProvidesPackageDrop{soulutf8}

File 294 **lwarf-splitidx.sty**

§ 390 Package **splitidx**

(Emulates or patches code by MARKUS KOHM.)

Pkg **splitidx** **splitidx** is patched for use by lwarf.

If the `latexmk` option is selected for lwarf, `latexmk` will compile the document but will *not* compile the indexes. `lwarpmk printindex` and `lwarpmk htmlindex` will still be required.

⚠ **\thepage** When using `\AtWriteToIndex` or `\AtNextWriteToIndex`, the user must not refer to `\thepage` during HTML output, as the concept of a page number is meaningless. Instead, do

```
\addtocounter{LWR@autoindex}{1}
\LWR@new@label{LWRindex-\arabic{LWR@autoindex}}
```

where the `\index`-like action occurs, and then refer to `\arabic{LWR@autoindex}` instead of `\thepage` where the reference should occur.

See section 469.17 in the lwarf-patch-memoir package for the `\@wrspindexhyp` macro as an example.

for HTML output: 1 \LWR@ProvidesPackagePass{splitidx}

```
2 \catcode`\_=12%
3 \xpatchcmd{\newindex}
4   {\jobname-#2.idx}
5   {\jobname-#2_html.idx}
6   {}
7   {\LWR@patcherror{splitidx}{\newindex}}
8 \catcode`\_=8%
```

Patched to use lwarf's automatic indexing counter instead of `\thepage`:

```
9 \renewcommand*{\@wrsindex}[2][]{%
10   \ifx\relax#1\relax
11   \if@splitidx
12     \@wrsindex[idx]{#2}%
13   \else
```

```

14      \def\@tempa{#2}%
15      \if@verbindex\@onellevel@sanitize\@tempa\fi
16      \@wrindex{\@tempa}%
17  \fi
18 \else
19      \def\@tempa{#2}%
20      \csname index@\#1@hook\endcsname
21%       \expandafter\ifx\csname @@wrsindex\endcsname\relax
22%         \addtocounter{LWR@autoindex}{1}%           lwarf
23%         \LWR@new@label{LWRindex-\arabic{LWR@autoindex}}%   lwarf
24%         \@@@wrsindex{#1}{\@tempa}{\thepage}%
25%         \@@@wrsindex{#1}{\@tempa}{\arabic{LWR@autoindex}}%
26%       \else
27%         \def\@tempb{\@@@wrsindex{#1}}%
28%         \expandafter\@tempb\@tempa||\\%
29%       \fi
30     \endgroup
31     \c@esphack
32  \fi
33 }

```

`lwarf` defines sectioning commands with `xparse`, so the below patches are done as temporary redefinitions instead of being `\let`.

```

34 \xpatchcmd{\printsubindex}
35   {\let\section\subsection}
36   {\renewcommand*\section{\subsection}}
37   {}
38   {\LWR@patcherror{splitidx}{printsubindex-section}}
39
40 \xpatchcmd{\printsubindex}
41   {\let\chapter\section}
42   {\renewcommand*\chapter{\section}}
43   {}
44   {\LWR@patcherror{splitidx}{printsubindex-chapter}}
45
46 \xpatchcmd{\printsubindex}
47   {\let\@makechapterhead\section}
48   {\def\@makechapterhead{\section}}
49   {}
50   {\LWR@patcherror{splitidx}{printsubindex-chapter}}

```

File 295 `lwarf-srcltx.sty`

§ 391 Package **srcltx**

Pkg `srcltx` `srcltx` is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{srccltx}[2006/11/12]

2 \newif\ifSRCOK \SRCOKfalse
3 \newcommand*\srcIncludeHook[1]{}
4 \newcommand*\srcInputHook[1]{}
5 \newcommand*\MainFile{}
6 \def\MainFile{\jobname.tex}
7 \newcommand*\CurrentInput{}
8 \gdef\CurrentInput{\MainFile}
9 \newcommand\Input{}
10 \let\Input\input

```

File 296 l warp-srctex.sty

§ 392 Package **srctex**

Pkg **srctex** srctex is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{srctex}[2006/11/12]
2 \LWR@origRequirePackage{l warp-srctex}

```

File 297 l warp-stabular.sty

§ 393 Package **stabular**

(Emulates or patches code by SIGITAS TOLUŠIS.)

Pkg **stabular** stabular is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{stabular}

```

Env **stabular** [$\langle vpos \rangle$] [$\langle colspec \rangle$]

```

2 \newenvironment{stabular}[2][c]
3 {
4 \begin{tabular}[\#1][\#2]
5 \renewcommand{\noalign}{\vphantom{#1}}
6 }
7 {\end{tabular}}

```

Env **stabular** [$\langle width \rangle$] [$\langle vpos \rangle$] [$\langle colspec \rangle$]

```

8 \NewDocumentEnvironment{stabular*}{m o m}

```

```

9 {
10 \begin{tabular}[#2]{#3}
11 \renewcommand{\noalign}{[1]{}}
12 }
13 {\end{tabular}}
```

File 298 **lwarf-stffloats.sty**

§ 394 Package **stfloats**

Pkg **stfloats** **stfloats** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{stfloats}

stfloats may have been preloaded by a **ltj*** class.

The following are provided in case they have not yet been defined:

```

2 \providecommand*\fnbelowfloat{}
3 \providecommand*\fnunderfloat{}
4 \providecommand*\setbaselinefloat{}
5 \providecommand*\setbaselinefixed{}
```

Nullified for HTML:

```

6 \renewcommand*\fnbelowfloat{}
7 \renewcommand*\fnunderfloat{}
8 \renewcommand*\setbaselinefloat{}
9 \renewcommand*\setbaselinefixed{}
```

File 299 **lwarf-subfig.sty**

§ 395 Package **subfig**

(Emulates or patches code by STEVEN DOUGLAS COCHRAN.)

Pkg **subfig** **subfig** is supported and patched by **lwarf**.

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

In the document source, use **\hfill** and **\hspace*** **subfig>inline** 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} lwarp

4 \IfValueTF{#2}{%
5     \LWR@setlatestname{#2}{%
6 }{%
7     \IfValueTF{#3}{%
8         \LWR@setlatestname{#3}{%
9     }{%
10 }{%
11 \LWR@stoppars% lwarp
12     \ifundefined{FBsc@max}{%
13         {\FB@readaux{\let\FBsubheight\relax}}{%
14     \tempcpta=\ne
15     \if@minipage
16         \tempcpta=z@
17     \else\ifdim \lastskip=z@ \else
18         \tempcpta=tw@
19     \fi\fi
20     \ifmaincaptiontop
21         \sf@top=\sf@nearskip
22         \sf@bottom=\sf@farskip
23     \else
24         \sf@top=\sf@farskip
25         \sf@bottom=\sf@nearskip
26     \fi
27     \leavevmode

28 \%     \setbox\tempboxa \hbox{#4}{%
29 \%     \tempdima=\wd\tempboxa
30 \%     \ifundefined{FBsc@max}{%
31 \%         {\global\advance\Xhsizet-\wd\tempboxa
32 \%             \dimen@\ht\tempboxa
33 \%             \advance\dimen@\dp\tempboxa
34 \%             \ifdim\dimen@>\FBso@max
35 \%                 \global\FBso@max\dimen@
36 \%             \fi}{%

```

Do not use boxes, which interfere with `lateximages`:

```
37%      \vtop%
38    \bgroup
39%      \vbox%
40    \bgroup
41      \ifcase\@tempcnta
42        \@minipagefalse
43        \or
44%          \vskip\sff@top
45        \or
46          \ifdim \lastskip=\z@ \else
47%            \tempskipb\sff@top\relax\zaddvskip
48          \fi
49        \fi
50        \sf@ifpositiontop{%
51          \ifx \empty#3\relax \else
52            \sf@subcaption{\#1}{\#2}{\#3}%
53%            \vskip\sff@capskip
54%            \vskip\sff@captionadj
55          \fi\egroup
56%          \hrule width0pt height0pt depth0pt
57          \LWR@startpars% lwarp
58% \box\@tempboxa
59          #4
60          \LWR@stoppars% lwarp
61        }%
62        \LWR@startpars% lwarp
63        \ifundefined{FBsc@max}%
64        {
65% \box\@tempboxa
66          #4
67        }%
68        {\ifx\FBsubheight\relax
69%          \box\@tempboxa
70          #4
71        \else
72%          \vbox to \FBsubheight{\FBafil\box\@tempboxa\FBbfil}%
73          #4
74        \fi}%
75        \LWR@stoppars% lwarp
76        \egroup
77        \ifx \empty#3\relax \else
78%          \vskip\sff@capskip
79%          \hrule width0pt height0pt depth0pt
80          \sf@subcaption{\#1}{\#2}{\#3}%
81          \fi
82        }%
83%        \vskip\sff@bottom
84      \egroup
85      \ifundefined{FBsc@max}{}%
86      {\addtocounter{FRobj}{-1}}%
```

```
87      \ifnum\c@FRobj=0\else
88          \subfloatrowsep
89      \fi}%
90  \ifmaincaptiontop\else
91      \global\advance\@nameuse{c@\@capttype}\m@ne
92  \fi
93 \end{minipage} \% lwarp
94 \LWR@startpars \% lwarp
95 \endgroup\ignorespaces%
96 }%  
  
\sf@subcaption {\langle 1 type\rangle} {\langle 2 LOF entry\rangle} {\langle 3 caption\rangle}
97 \long\def\sf@subcaption#1#2#3{%
98 \LWR@stopars \% lwarp
99 \ifx \relax#2\relax \else
100 \bgroup
101 \let\label=\gobble
102 \let\protect\string
103 \def\@subcaplabel{%
104 \caption@lstfmt{\@nameuse{p@\#1}}{\@nameuse{the\#1}}}}%
105 \sf@updatecaptionlist{\#1}{\#2}{\the\value{\@capttype}}{\the\value{\#1}}%
106 \egroup
107 \fi
108 \bgroup
109 \ifx \relax#3\relax
110 \let\captionlabelsep=\relax
111 \fi
112 \% \setbox0\vbox{%
113 \% \hb@xt@{\the\tempdima}{%
114 \% \hss
115 \% \parbox[t]{\the\tempdima}{%
116 \% \caption@make
117 \% {\@nameuse{sub\@capttype name}}%
118 \% {\@nameuse{thesub\@capttype}}%
119 \% {\#3}%
120 \% }%
121 \% }%
122 \% \hss
123 \% }
124 \% }%
125 \@ifundefined{FBsc@max}%
126 \% {\box0}%
127 {
128 \% \parbox[t]{\the\tempdima}{%
129 \LWR@traceinfo{sfsubcap B1} \% lwarp
130 \LWR@figcaption \% lwarp
131 \caption@make
132 {\@nameuse{sub\@capttype name}}%
```

```

133          {\@nameuse{thesub@\capttype}\}%
134          {\LWR@isolate{\#3}\}%
135          \endLWR@figcaption% lwarf
136 \LWR@traceinfo{sfsubcap B2}\% lwarf
137 \% }%
138 }%
139 {\dimen@\ht0%
140 \advance\dimen@\dp0%
141 \ifdim\dimen@>\FBsc@max
142     \global\FBsc@max\dimen@
143 \fi
144 \FB@readaux{\let\FBsubcheight\relax}%
145 \ifx\FBsubcheight\relax
146     \def\next{
147 \% \parbox[t]{\the\@tempdima}
148         }%
149     \else
150     \def\next{
151 \% \parbox[t][\FBsubcheight][t]{\the\@tempdima}
152         }%
153     \fi
154 \% \vbox{%
155 \%     \hb@xt@{\the\@tempdima}{%
156         \hss
157 \%         \next{%
158 \%             \caption@make
159 \LWR@traceinfo{sfsubcap C1}\% lwarf
160             {\@nameuse{sub@\capttype name}\}%
161             {\@nameuse{thesub@\capttype}\}%
162             {\#3}
163
164 \LWR@traceinfo{sfsubcap C1}\% lwarf
165 \% }%
166 \% \hss
167
168 \% }
169 \% }%
170 }%
171 \egroup
172 \LWR@startpars% lwarf
173 }

```

\subfloat@label Patches for \sf@sub@label:

```

174 \def\subfloat@label{%
175 \LWR@ensuredoingapar% lwarf
176     \@ifnextchar(% %) match left parenthesis
177     {\sf@sub@label}
178     {\sf@sub@label(Sub@\capttype\space

```

```

179          \@ifundefined{thechapter}{}{\@nameuse{thechapter}\space}%
180          \@nameuse{p@sub\@capttype}%
181          \@nameuse{thesub\@capttype}.)}

```

Patches for \subref.

```
\sf@subref {<label>}
```

The unstarred version uses a \ref link whose printed text comes from the sub@<label>:

```

182 \renewcommand{\sf@subref}[1]{%
183 \LWR@subnewref[#1]{sub@#1}%
184 }

```

```
\sf@@subref {<label>}
```

The starred version uses the printed sub@<label> which is stored as if it were a page number:

```
185 \renewcommand{\sf@@subref}[1]{\LWR@orig@pageref{sub@#1}}
```

Defining new subfloats. The l@sub<type> for each is redefined.

```
\@newsubfloat [{<keys/values>}]{<float name>}
```

```

186 \LetLtxMacro{\LWR@orig@newsubfloat}{\newsubfloat}
187
188 \def\@newsubfloat[#1]#2{%
189 \LWR@orig@newsubfloat[#1]{#2}%
190 \renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}%
191 }

```

Pre-defined for figures and tables:

```
\l@subfigure {<text>} {<pagenum>}
```

```
192 \renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2}}
```

```
\l@subtable {<text>} {<pagenum>}
```

```
193 \renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}
```

File 300 **lwarp-subfigure.sty**

§ 396 Package **subfigure**

Pkg **subfigure** subfigure is emulated by subfig.

```

for HTML output: 1 \LWR@ProvidesPackageDrop{subfigure}
2 \RequirePackage{subfig}

3 \LetLtxMacro{\subfigure}{\subfloat}
4 \LetLtxMacro{\subtable}{\subfloat}
5 \LetLtxMacro{\Subref}{\subref}
6 \@ifundefined{figuretopcaptrue}{\newif\iffiguretopcap}{}
7 \newif\iffiguretopcap
8 \newif\ifsubcaphang
9 \newif\ifsubcapcenter
10 \newif\ifsubcapcenterlast
11 \newif\ifsubcapnooneline
12 \newif\ifsubcapraggedright
13 \newskip\subfigtopskip
14 \newskip\subfigcapskip
15 \newdimen\subfigcapadj
16 \newskip\subfigbottomskip
17 \newdimen\subfigcapmargin
18 \newskip\subfiglabelskip
19 \newcommand*{\subcapsize}{}
20 \newcommand*{\subcaplabelfont}{}
21 \newcommand*{\subcapfont}{}

```

File 301 **lwarf-supertabular.sty**

§ 397 Package **supertabular**

(Emulates or patches code by JOHANNES BRAAMS, THEO JURRIENS.)

Pkg supertabular supertabular is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{supertabular}

⚠ Misplaced alignment tab character &

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

```

\StartDefiningTabulars
\tablefirsthead
...
\StopDefiningTabulars

```

See section 9.9.

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

```

2 \newcommand{\LWRST@firsthead}{}
3
4 \newcommand{\tablefirsthead}[1]{%

```

```
5      \long\gdef\LWRST@firsthead{#1}%
6 }
7
8 \newcommand{\tablehead}[1]{}
9 \newcommand{\tabletail}[1]{}
10
11 \newcommand{\LWRST@lasttail}{}
12
13 \newcommand{\tablelasttail}[1]{%
14     \long\gdef\LWRST@lasttail{#1}%
15 }

16 \newcommand{\tablecaption}[2][]{%
17     \long\gdef\LWRST@caption{%
18         \ifblank{#1}{%
19             {\caption{#2}}%
20             {\caption[#1]{#2}}%
21         }%
22     }%
23
24 \let\topcaption\tablecaption
25 \let\bottomcaption\tablecaption

26 \newcommand*\LWRST@caption{%
27
28 \newcommand*\shrinkheight[1]{%
29
30 \NewDocumentEnvironment{supertabular}{s o m}
31 {%
32 \LWR@traceinfo{supertabular}%
33 \begin{table}%
34 \LWRST@caption%
35 \begin{tabular}{#3}%
36 \TabularMacro\ifdefvoid{\LWRST@firsthead}{%
37 {\LWR@getmynexttoken}%
38 {\expandafter\LWR@getmynexttoken\LWRST@firsthead}%
39 }%
40 {%
41 \ifdefvoid{\LWRST@lasttail}{%
42 {}%
43 {%
44 \TabularMacro\ResumeTabular%
45 \LWRST@lasttail%
46 }%
47 \end{tabular}%
48 \end{table}%
49 \LWR@traceinfo{supertabular done}%
50 }%
51
```

```
52 \NewDocumentEnvironment{mpsupertabular}{s o m}
53 {\minipage{\linewidth}\supertabular{#3}}
54 {\endsupertabular\endminipage}
```

File 302 **l warp-syntonly.sty**

§ 398 Package **syntonly**

(Emulates or patches code by FRANK MITTELBACH, RAINER SCHÖPF.)

Pkg **syntonly** Emulated.

for HTML output: Discard all options for l warp-syntonly:

```
1 \LWR@ProvidesPackageDrop{syntonly}

2 \newif\ifsyntax@
3 \syntax@false
4
5 \newcommand*\syntonly{}{}
6
7 \onlypreamble\syntonly
```

File 303 **l warp-tables.sty**

§ 399 Package **tables**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg **tables** tables is emulated. \LWR@hline is used to handle the optional argument when tables is loaded.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{tables}

2 \newdimen\tablinesep
3 \newdimen\arraylinesep
4 \newdimen\extrarulesep
```

File 304 **l warp-tabularx.sty**

§ 400 Package **tabularx**

(Emulates or patches code by DAVID CARLISLE.)

Pkg **tabularx** tabularx is emulated by l warp.

for HTML output: Discard all options for l warp-tabularx:

```
1 \LWR@ProvidesPackageDrop{tabularx}

2 \DeclareDocumentEnvironment{tabularx}{m o m}
3 {\begin{array}{#3}}
4 {\end{array}}
5
6 \DeclareDocumentEnvironment{tabularx*}{m o m}
7 {\begin{array}{#3}}
8 {\end{array}}
```

File 305 **l warp-tabulary.sty**

§ 401 Package **tabulary**

(Emulates or patches code by DAVID CARLISLE.)

Pkg **tabulary** tabulary is emulated by l warp.

for HTML output: Discard all options for l warp-tabulary.

Column types L, C, R, and J are emulated by l warp core code.

```
1 \LWR@ProvidesPackageDrop{tabulary}

2 \NewDocumentEnvironment{tabulary}{m o m}
3 {\begin{array}{#3}}
4 {\end{array}}
5
6 \NewDocumentEnvironment{tabulary*}{m o m}
7 {\begin{array}{#3}}
8 {\end{array}}
9
10 \newdimen\tymin
```

```
11 \newdimen\tymax
12 \def\tyformat{}
```

File 306 **lwarf-tascmac.sty**

§ 402 Package **tascmac**

Pkg tascmac tascmac is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{tascmac}

2 \newenvironment*{boxnote}
3   {
4     \BlockClass[
5       padding: .5ex ;
6       border: 1px solid black ;
7       border-top: 1px dashed black ;
8     ]{boxnote}
9   }
10  {\endBlockClass}

11 \newenvironment*{screen}[1] []
12   {
13     \BlockClass[
14       padding: .5ex ;
15       border: 1px solid gray ;
16       border-radius: 8pt
17     ]{boxnote}
18   }
19  {\endBlockClass}

20 \newenvironment*{itembox}[2] []
21   {
22     \BlockClass[
23       padding: .5ex ;
24       border: 1px solid gray ;
25       border-radius: 8pt
26     ]{boxnote}
27     \InlineClass{itemboxtitle}{#2}\par
28   }
29  {\endBlockClass}

30 \newenvironment*{shadebox}
31   {
32     \BlockClass[
33       padding: .5ex ;
34       border: 1px solid black ;
```

```
38         box-shadow: 3px 3px 3px \#808080 ;
39     ]{boxnote}
40 }
41 {\endBlockClass}
42
43 \newcommand*{\mask}[2]{%
44     \InlineClass[background: lightgray]{mask}{#1}%
45 }
46
47 \newcommand*{\maskbox}[5]{%
48     \InlineClass[background: lightgray]{mask}{#5}%
49 }
50
51 \newcommand*{\Maskbox}[6]{%
52     \InlineClass[
53         background: lightgray ;
54         border: #5 solid black
55     ]{mask}{#6}%
56 }
57
58 \newcommand*{\keytop}[2][]{%
59     \InlineClass[%]
60         padding: .2ex ;
61         border: 1px solid black ;
62         border-radius: .7ex ;
63     ]{keytop}{#2}%
64 }
65
66 \def\yen{\HTMLunicode{00A5}}
67
68 \def\return{\HTMLunicode{23CE}}
69
70 \def\Return{\HTMLunicode{23CE}}
71
72 \def\ascii{ASCII Corporation}
73
74 \def\Ascii{ASCII Corporation}
75
76 \def\ASCII{ASCII Corporation}
```

File 307 **lwarf-textarea.sty**

§ 403 Package **textarea**

(Emulates or patches code by ALEXANDER I. ROZHENKO.)

Pkg **textarea** **textarea** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{textarea}

2 \newcommand\StartFromTextArea{}
3 \newcommand\StartFromHeaderArea{}
4 \newcommand*\RestoreTextArea{}
5 \newcommand*\ExpandTextArea[1][*]{}
6 \let\NCC@restoretexarea\empty

```

File 308 **l warp-textcomp.sty**

§ 404 Package **textcomp**

(Emulates or patches code by FRANK MITTELBACH, ROBIN FAIRBAIRNS, WERNER LEMBERG.)

Pkg **textcomp** **textcomp** is patched for use by **l warp**.

§ 404.1 Limitations

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

⚠ **missing symbols** Many **textcomp** symbols are not supported by many fonts. In the **css** try referencing fonts which are more complete, but expect to see gaps in coverage.

§ 404.2 Package loading

for HTML output:

```

1 \LWR@ProvidesPackagePass{textcomp}

```

§ 404.3 HTML symbols

For **HTML**, use **HTML** entities or direct **Unicode**, depending on the engine.

\AtBeginDocument improves support for **Lua^{LT}E** and **X^{LT}E**.

§ 404.3.1 pdf^{LT}E symbols

```

2 \AtBeginDocument{
3 \ifPDFTeX% pdflatex or dvi latex
4 \newcommand*{\LWR@HTML@textdegree}{\HTMLentity{deg}}
5 \newcommand*{\LWR@HTML@textcelsius}{\HTMLunicode{2103}}
6 \newcommand*{\LWR@HTML@textohm}{\HTMLunicode{2126}}
7 \newcommand*{\LWR@HTML@textmu}{\HTMLunicode{00B5}}
8 \newcommand*{\LWR@HTML@textlquill}{\HTMLunicode{2045}}
9 \newcommand*{\LWR@HTML@textrquill}{\HTMLunicode{2046}}

```

```

10 \newcommand*{\LWR@HTML@textcircledP}{\HTMLunicode{2117}}
11 \newcommand*{\LWR@HTML@texttwelveudash}{\HTMLunicode{2014}}% emdash
12 \newcommand*{\LWR@HTML@textthreequartersemdash}{\HTMLunicode{2014}}% emdash
13 \newcommand*{\LWR@HTML@textmho}{\HTMLunicode{2127}}
14 \newcommand*{\LWR@HTML@textnaira}{\HTMLunicode{20A6}}
15 \newcommand*{\LWR@HTML@textpeso}{\HTMLunicode{20B1}}
16 \newcommand*{\LWR@HTML@textrecipe}{\HTMLunicode{211E}}
17 \newcommand*{\LWR@HTML@textinterrobangdown}{\HTMLunicode{2E18}}
18 \newcommand*{\LWR@HTML@textperthousand}{\HTMLunicode{2030}}
19 \newcommand*{\LWR@HTML@textpertenthousand}{\HTMLunicode{2031}}
20 \newcommand*{\LWR@HTML@textbaht}{\HTMLunicode{0E3F}}
21 \newcommand*{\LWR@HTML@textdiscount}{\%}
22 \newcommand*{\LWR@HTML@textservicemark}{\HTMLunicode{2120}}
23 \else

```

§ 404.3.2 X_EL_AT_EX and L_UaL_AT_EX symbols

NOTE: Some of the following do not print well in the listing. Consult the .dtx or .sty file for the actual characters.

```

24 \newcommand*{\LWR@HTML@textdegree}{°}
25 \newcommand*{\LWR@HTML@textcelsius}{°C}
26 \newcommand*{\LWR@HTML@textohm}{Ω}
27 \newcommand*{\LWR@HTML@textmu}{μ}
28 \newcommand*{\LWR@HTML@textlquill}{{ℓ}}
29 \newcommand*{\LWR@HTML@textrquill}{{rℓ}}
30 \newcommand*{\LWR@HTML@textcircledP}{©}
31 \newcommand*{\LWR@HTML@texttwelveudash}{--}% emdash
32 \newcommand*{\LWR@HTML@textthreequartersemdash}{--}% emdash
33 \newcommand*{\LWR@HTML@textmho}{Ü}
34 \newcommand*{\LWR@HTML@textnaira}{₦}
35 \newcommand*{\LWR@HTML@textpeso}{₱}
36 \newcommand*{\LWR@HTML@textrecipe}{฿}
37 \newcommand*{\LWR@HTML@textinterrobangdown}{₧}
38 \newcommand*{\LWR@HTML@textperthousand}{.%}
39 \newcommand*{\LWR@HTML@textpertenthousand}{%.%}
40 \newcommand*{\LWR@HTML@textbaht}{฿}
41 \newcommand*{\LWR@HTML@textdiscount}{\%}
42 \newcommand*{\LWR@HTML@textservicemark}{}
43 \fi
44
45 \LWR@formatted{textdegree}
46 \LWR@formatted{textcelsius}
47 \LWR@formatted{textohm}
48 \LWR@formatted{textmu}
49 \LWR@formatted{textlquill}
50 \LWR@formatted{textrquill}
51 \LWR@formatted{textcircledP}
52 \LWR@formatted{texttwelveudash}

```

```

53 \LWR@formatted{textthreequartersemdash}
54 \LWR@formatted{textmho}
55 \LWR@formatted{textnaira}
56 \LWR@formatted{textpeso}
57 \LWR@formatted{textrecipe}
58 \LWR@formatted{textinterrobangdown}
59 \LWR@formatted{textperthousand}
60 \LWR@formatted{textpertenthousand}
61 \LWR@formatted{textbaht}
62 \LWR@formatted{textdiscount}
63 \LWR@formatted{textservicemark}

```

§ 404.4 HTML diacritics

For HTML, Unicode diacritical marks are used:

```

64 \newcommand*\{\LWR@HTML@capitalcedilla}[1]{\#1\HTMLunicode{0327}}
65 \newcommand*\{\LWR@HTML@capitalogonek}[1]{\#1\HTMLunicode{0328}}
66 \newcommand*\{\LWR@HTML@capitalgrave}[1]{\#1\HTMLunicode{0300}}
67 \newcommand*\{\LWR@HTML@capitalacute}[1]{\#1\HTMLunicode{0301}}
68 \newcommand*\{\LWR@HTML@capitalcircumflex}[1]{\#1\HTMLunicode{0302}}
69 \newcommand*\{\LWR@HTML@capitaltilde}[1]{\#1\HTMLunicode{0303}}
70 \newcommand*\{\LWR@HTML@capitaldieresis}[1]{\#1\HTMLunicode{0308}}
71 \newcommand*\{\LWR@HTML@capitalhungarumlaut}[1]{\#1\HTMLunicode{30B}}
72 \newcommand*\{\LWR@HTML@capitalring}[1]{\#1\HTMLunicode{30A}}
73 \newcommand*\{\LWR@HTML@capitalcaron}[1]{\#1\HTMLunicode{30C}}
74 \newcommand*\{\LWR@HTML@capitalbreve}[1]{\#1\HTMLunicode{306}}
75 \newcommand*\{\LWR@HTML@capitalmacron}[1]{\#1\HTMLunicode{304}}
76 \newcommand*\{\LWR@HTML@capitaldotaccent}[1]{\#1\HTMLunicode{307}}

```

\textcircled becomes a span with a rounded border. \providecommand is used to avoid conflict with xunicode.

```

77 \providecommand*\{\LWR@HTML@textcircled}[1]{%
78 \InlineClass[border: 1px solid \LWR@currenttextcolor]{textcircled}{\#1}%
79 }
80
81 \LWR@formatted{capitalcedilla}
82 \LWR@formatted{capitalogonek}
83 \LWR@formatted{capitalgrave}
84 \LWR@formatted{capitalacute}
85 \LWR@formatted{capitalcircumflex}
86 \LWR@formatted{capitaltilde}
87 \LWR@formatted{capitaldieresis}
88 \LWR@formatted{capitalhungarumlaut}
89 \LWR@formatted{capitalring}
90 \LWR@formatted{capitalcaron}
91 \LWR@formatted{capitalbreve}
92 \LWR@formatted{capitalmacron}

```

```

93 \LWR@formatted{capitaldotaccent}
94 \LWR@formatted{textcircled}
95
96 }% AtBeginDocument

```

File 309 **lwarf-textfit.sty**

§ 405 Package **textfit**

Pkg **textfit** **textfit** is emulated.

Text is placed into a of class **textfit**. Sizes are approximated, and also limited by browser min/max font-size settings.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{textfit}

2 \newsavebox{\LWR@textfitbox}
3
4 \newcommand*{\LWR@textfitscale}[2]{%
5 \setlength{\LWR@templengthone}{#1}%
6 \setlength{\LWR@templengthone}{%
7 1em*\ratio{\LWR@templengthone}{\LWR@templengthtwo}%
8 }%
9 \InlineClass[font-size:\LWR@printlength{\LWR@templengthone}]{textfit}{#2}%
10 }

11
12 \newcommand*{\scaletowidth}[2]{%
13 \sbox{\LWR@textfitbox}{#2}%
14 \settowidth{\LWR@templengthtwo}{\usebox{\LWR@textfitbox}}%
15 \LWR@textfitscale{#1}{#2}%
16 }
17
18 \newcommand*{\scaletoheight}[2]{%
19 \sbox{\LWR@textfitbox}{#2}%
20 \settoheight{\LWR@templengthtwo}{\usebox{\LWR@textfitbox}}%
21 \LWR@textfitscale{#1}{#2}%
22 }

```

File 310 **lwarf-textpos.sty**

§ 406 Package **textpos**

(Emulates or patches code by NORMAN GRAY.)

Pkg **textpos** **textpos** is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{textpos}

2 \NewDocumentEnvironment{textblock}{m r(){}{}}
3 \NewDocumentEnvironment{textblock*}{m o r(){}{}}
4 \newcommand*\{\\TPGrid}[3][]{}
5 \NewDocumentCommand{\TPMargin}{s o}{}
6 \newcommand*\{\\textblockcolour}[1] {}
7 \newcommand*\{\\textblockrulecolour}[1] {}
8 \newcommand*\{\\textblockcolor}[1] {}
9 \newcommand*\{\\textblockrulecolor}[1] {}
10 \newcommand*\{\\tekstblokkulur}[1] {}
11 \newcommand*\{\\tekstblokrulekulur}[1] {}
12 \newlength{\TPHorizModule}
13 \newlength{\TPVertModule}
14 \newlength{\TPboxrulesize}
15 \newcommand{\textblocklabel}[1]{}
16 \newcommand*\{\\showtextsize}={}
17 \newcommand{\textblockorigin}[2]{}

```

File 311 **l warp-theorem.sty**

§ 407 Package **theorem**

(Emulates or patches code by FRANK MITTELBACH.)

Pkg **theorem** **theorem** is patched for use by **l warp**.

Table 16: Theorem package — css styling of theorems and proofs

Theorem: <div> of class theorembody<theoremstyle>

Theorem Header: of class theoremheader

where <theoremstyle> is plain, break, etc.

for HTML output:

```

1 \LWR@ProvidesPackagePass{theorem}

```

§ 407.1 Remembering the theorem style

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
3 \gdef\theoremstyle#1{%
4   \@ifundefined{th@#1}{\@warning
5     {Unknown theoremstyle '#1'. Using 'plain'}%
6     \theorem@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}\l warp
8   }%
9   \%
10  \theorem@style{#1}%
11  \renewcommand{\LWR@newtheoremstyle}{#1}\l warp
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\xnithm#1#2[#3]{%
17   \expandafter\@ifdefinable\csname #1\endcsname
18   {%
19     \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}\l warp
20     \@definecounter{#1}\@newctr{#1}[#3]%
21     \expandafter\xdef\csname the#1\endcsname
22       {\expandafter\noexpand\csname the#3\endcsname
23         \@thmcOUNTERsep\@thmcOUNTER{#1}}%
24     \def@tempa{\global\@namedef{#1}}%
25     \expandafter\@tempa\expandafter{%
26       \csname th@\the\theorem@style
27         \expandafter\endcsname\the\theorem@bodyfont
28       \@thm{#1}{#2}}%
29     \global\expandafter\let\csname end#1\endcsname\@endtheorem
30     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\@nameuse{\LWR@thmstyle#1}}}\l warp
31   }%
32 }
33 \gdef\ynithm#1#2{%
34   \expandafter\@ifdefinable\csname #1\endcsname
35   {
36     \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}\l warp
37     \@definecounter{#1}%
38     \expandafter\xdef\csname the#1\endcsname{\@thmcOUNTER{#1}}%
39     \def@tempa{\global\@namedef{#1}}\expandafter\@tempa
```

```

40      \expandafter{\csname th@\the \theorem@style \expandafter
41      \endcsname \the\theorem@bodyfont \othm{#1}{#2}}%
42      \global \expandafter \let \csname end#1\endcsname \endtheorem
43      \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\@nameuse{LWR@thmstyle#1}}}% lwarp
44  }
45
46 \gdef\othm#1[#2]{%
47   \expandafter\ifx\csname c@#2\endcsname\relax
48   \@nocounterr{#2}%
49   \else
50   \expandafter\@if definable\csname #1\endcsname
51   {
52   \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
53   \expandafter \xdef \csname the#1\endcsname
54   {\expandafter\noexpand \csname the#2\endcsname}%
55   \def\@tempa{\global\@namedef{#1}}\expandafter \attempa
56   \expandafter{\csname th@\the \theorem@style \expandafter
57   \endcsname \the\theorem@bodyfont \othm{#2}{#3}}%
58   \global \expandafter \let \csname end#1\endcsname \endtheorem
59   \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\@nameuse{LWR@thmstyle#1}}}% lwarp
60   }%
61 \fi}

```

§ 407.2 **css patches**

The following are patched for css.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc. They are gathered together here.

Each theorem is encased in a `BlockClass` environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader`.

```

62 \gdef\th@plain{%
63   \def\@begintheorem##1##2{%
64     \item[
65       \InlineClass{theoremheader}{##1\ ##2}
66     ]
67   }%
68 \def\@opargbegintheorem##1##2##3{%
69   \item[
70     \InlineClass{theoremheader}{##1\ ##2\ (#3)}
71   ]
72 }
73 }
74
75 \gdef\th@break{%
76   \def\@begintheorem##1##2{%

```

```
77      \item[
78          \InlineClass{theoremheader}{##1\ ##2}\newline%
79      ]
80  }%
81 \def\@opargbegintheorem##1##2##3{%
82   \item[
83       \InlineClass{theoremheader}{##1\ ##2\ (###3)}\newline%
84   ]
85 }
86 }
87
88 \gdef\th@marginbreak{%
89   \def\@begintheorem##1##2{%
90     \item[
91         \InlineClass{theoremheader}{##2 \qquad ##1}\newline%
92     ]
93  }%
94 \def\@opargbegintheorem##1##2##3{%
95   \item[
96       \InlineClass{theoremheader}{##2 \qquad ##1\ %
97       (###3)}\newline%
98   ]
99 }
100 }
101
102 \gdef\th@changebreak{%
103   \def\@begintheorem##1##2{%
104     \item[
105         \InlineClass{theoremheader}{##2\ ##1}\newline%
106     ]
107  }%
108 \def\@opargbegintheorem##1##2##3{%
109   \item[
110       \InlineClass{theoremheader}{##2\ ##1\ %
111       (###3)}\newline%
112   ]
113 }
114 }
115
116 \gdef\th@change{%
117   \def\@begintheorem##1##2{%
118     \item[
119         \InlineClass{theoremheader}{##2\ ##1}%
120     ]
121  }%
122 \def\@opargbegintheorem##1##2##3{%
123   \item[
124       \InlineClass{theoremheader}{##2\ ##1\ (###3)}%
125   ]
126 }
```

```

127 }
128
129 \gdef\th@margin{%
130   \def\@begintheorem##1##2{%
131     \item[%
132       \InlineClass{theoremheader}{##2 \qquad ##1}%
133     ]%
134   }%
135 \def\@opargbegintheorem##1##2##3{%
136   \item[%
137     \InlineClass{theoremheader}{##2 \qquad ##1\ (#3)}%
138   ]%
139 }
140 }

```

Patched for css:

```

141 \gdef\@thm#1#2{\refstepcounter{#1}%
142 \LWR@forcenewpage% l warp
143   \BlockClass{theorembody}\LWR@thisthmstyle}% l warp
144   \trivlist
145   \topsep \theorempreskipamount           % used by first \item
146   \topsepadd \theorempostskipamount        % used by \endparenv
147   \ifnextchar [%%
148     {\@ythm{#1}{#2}}%
149     {\@begintheorem{#2}{\csname the#1\endcsname}\ignorespaces}}
150
151 \gdef\@endtheorem{%
152 \endtrivlist
153 \endBlockClass
154 }

```

File 312 **l warp-thinsp.sty**

§ 408 Package **thinsp**

Pkg thinsp thinsp is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{thinsp}

2 \AtBeginDocument{
3 \let\thinthinspace\relax% defined by some packages
4 \newcommand*\thinthinspace{\thinspace}
5 }
6
7 \newcommand*\stretchthinspace{\thinspace}
8 \newcommand*\stretchthinthinspace{\thinthinspace}

```

```
9 \newcommand*{\stretchnegthinspace}{\negthinspace}
```

File 313 **l warp-threadcol.sty**

§ 409 Package **threadcol**

Pkg **threadcol** **threadcol** is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{threadcol}
2 \newcommand{\setthreadname}[1]{}
```

File 314 **l warp-threeparttable.sty**

§ 410 Package **threeparttable**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg **threeparttable** **threeparttable** is emulated.

Table note are contained inside a css <div> of class **tnotes**. If **enumitem** is used, the note item labels are also individually highlighted with an additional css of class **tnoteitemheader**, otherwise they are plain text.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{threeparttable}
\LWR@printtablenote {\langle text\rangle}
Prints the table note item header inside a css class of tnoteitemheader.
2 \newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}
```

Env threeparttable [*alignment*] To emulate **threeparttable**:

```
3 \newenvironment*{threeparttable}[1][b]{}{}
```

Env tablenotes [*options*]

```
4 \newenvironment*{tablenotes}[1] []
5 {%
6 \LWR@forcenewpage
7 \BlockClass{tnotes}%
8 \ltx@ifpackageloaded{enumitem}{%
```

```

9 \setlist[description]{format=\LWR@printtablenote}%
10 }{}}%
11 \description%
12 }%
13 {%
14 \enddescription%
15 \endBlockClass%
16 }

\tnote {\langle text\rangle}
17 \newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}

```

File 315 **lwarf-thumb.sty**

§ 411 Package **thumb**

Pkg **thumb** **thumb** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{thumb}

2 \newcommand*\Overviewpage(){}
3 \newlength{\thumbheight}
4 \newlength{\thumbwidth}

```

File 316 **lwarf-thumbs.sty**

§ 412 Package **thumbs**

Pkg **thumbs** **thumbs** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{thumbs}

2 \newcommand{\addthumb}[4] {}
3 \newcommand{\addtitlethumb}[5] {}
4 \newcommand{\stopthumb} {}
5 \newcommand{\continuethumb} {}
6 \newcommand{\thumbsoverview}[1] {}
7 \newcommand{\thumbsoverviewback}[1] {}
8 \newcommand{\thumbsoverviewverso}[1] {}
9 \newcommand{\thumbsoverviewdouble}[1] {}
10 \newcommand{\thumbnewcolumn} {}
11 \newcommand{\addthumbsoverviewtocontents}[2] {}
12 \newcommand{\thumbsnophantom} {}

```

File 317 **l warp-tikz.sty**

§ 413 Package **tikz**

(Emulates or patches code by TILL TANTAU.)

Pkg **tikz** tikz is supported.

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

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

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

Accept all options for l warp-tikz:

```
1 \LWR@ProvidesPackagePass{tikz} [2015/08/07]
```

catcodes l warp changes the catcode of \$ for its own use. The Tikz babel library temporarily changes catcodes back to normal for Tikz's use. tikz v3.0.0 introduced the babel library which handles catcode changes. For older versions, l warp must change \$'s catcode itself.

Also see:

<https://tex.stackexchange.com/questions/16199/test-if-a-package-or-package-option-is-loaded>

```
2 \newbool{LWR@tikzbabel}
3
4 \@ifpackagelater{tikz}{2013/12/20}% Test for Tikz version v3.0.0
5 {\usetikzlibrary{babel}\booltrue{LWR@tikzbabel}}
6 {\boolfalse{LWR@tikzbabel}}
```

Env pgfpicture The `\pgfpicture` environment is enclosed inside a `\latextimage`. Enclose the low-level `\pgfpicture` in a `\latextimage`. This is also used by the higher-level `\tikz` and `\tikzpicture`.

```
7 \preto\pgfpicture{%
8   \begin{latextimage}%
9     \ifbool{LWR@tikzbabel}{% Test for Tikz version v3.0.0
10       {}%
```

```

11      {\catcode`\$=3}% dollar sign is math shift
12 }
13
14 \appto\endpgfpicture{\end{lateximage}}

```

Tikz is placed inside an svg image, so use the original meanings of the following:

```

15 \LetLtxMacro\pgfutil@minipage\LWR@print@minipage
16 \let\pgfutil@endminipage\endLWR@print@minipage
17
18 \let\pgfutil@raggedleft\LWR@print@raggedleft
19 \let\pgfutil@raggedright\LWR@print@raggedright
20
21 \def\pgfutil@font@tiny{\LWR@printtiny}
22 \def\pgfutil@font@scriptsize{\LWR@printscriptsize}
23 \def\pgfutil@font@footnotesize{\LWR@printfootnotesize}
24 \def\pgfutil@font@small{\LWR@print@small}
25 \def\pgfutil@font@normalsize{\LWR@print@normalsize}
26 \def\pgfutil@font@large{\LWR@printlarge}
27 \def\pgfutil@font@Large{\LWR@printLarge}
28 \def\pgfutil@font@huge{\LWR@printhuge}
29 \def\pgfutil@font@Huge{\LWR@printHuge}
30
31 \def\pgfutil@font@itshape{\LWR@origitshape}
32 \def\pgfutil@font@bfseries{\LWR@origbfseries}
33
34 \def\pgfutil@font@normalfont{\LWR@orignormalfont}

```

File 318 lwarf-titleps.sty

§ 414 Package **titleps**

(Emulates or patches code by JAVIER BEZOS.)

Pkg titleps titleps is loaded and used by lwarf during HTML output. All user options and macros are ignored and disabled.

Discard all options for lwarf-titleps:

for HTML output: 1 \LWR@ProvidesPackageDrop{titleps}

\pagestyle and \thispagestyle are already disabled in the lwarf code.

```

\newpagestyle  {{<name>} [<style>] {<commands>}}
2 \NewDocumentCommand{\newpagestyle}{m o m}{} 

```

```
\renewpagestyle {⟨name⟩} [⟨style⟩] {⟨commands⟩}
 3 \NewDocumentCommand{\renewpagestyle}{m o m}{}{}

\sethead [⟨el⟩] [⟨ec⟩] [⟨er⟩] {⟨ol⟩} {⟨oc⟩} {⟨or⟩}
 4 \NewDocumentCommand{\sethead}{o o o m m m}{}{}

\setfoot [⟨el⟩] [⟨ec⟩] [⟨er⟩] {⟨ol⟩} {⟨oc⟩} {⟨or⟩}
 5 \NewDocumentCommand{\setfoot}{o o o m m m}{}{}

\settlemarks * {⟨names⟩}
 6 \NewDocumentCommand{\settlemarks}{s m}{}{}

\headrule
 7 \newcommand*{\headrule}{}{}

\footrule
 8 \newcommand*{\footrule}{}{}

\setheadrule {⟨length⟩}
 9 \newcommand*{\setheadrule}[1]{}{}

\setfootrule {⟨length⟩}
10 \newcommand*{\setfootrule}[1]{}{}

\makeheadrule
11 \newcommand*{\makeheadrule}{}{}

\makefootrule
12 \newcommand*{\makefootrule}{}{}

\setmarkboth {⟨code⟩}
13 \newcommand{\setmarkboth}[1]{}{}

\widenhead
14 \NewDocumentCommand{\widenhead}{s o o m m}{}{}
```

```
\bottitlemarks
15 \newcommand*\bottitlemarks{}

\toptitlemarks
16 \newcommand*\toptitlemarks{}

\firstdtitlemarks
17 \newcommand*\firstdtitlemarks{}

\nextdtitlemarks
18 \newcommand*\nextdtitlemarks{}

\outertitlemarks
19 \newcommand*\outertitlemarks{}

\innertitlemarks
20 \newcommand*\innertitlemarks{}

\newtitlemark * {\langle name\rangle}
21 \NewDocumentCommand{\newtitlemark}{s m}{}

\pretitlemark * {\langle section\rangle} {\langle text\rangle}
22 \NewDocumentCommand{\pretitlemark}{s m m}{}

\ifsamemark {\langle group\rangle} {\langle command\rangle} {\langle true\rangle} {\langle false\rangle}
23 \newcommand{\ifsamemark}{[4]{}}
```



```
\setfloathead * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨extra⟩} [⟨which⟩]
24 \NewDocumentCommand{\setfloathead}{s o o o m m m m}{}
```



```
\setfloatfoot * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨extra⟩} [⟨which⟩]
25 \NewDocumentCommand{\setfloatfoot}{s o o o m m m m}{}
```



```
\nextfloathead * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨extra⟩} [⟨which⟩]
```

```

26 \NewDocumentCommand{\nextfloathead}{s o o o m m m m m}{}

\nextfloatfoot * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨extra⟩} [⟨which⟩]
27 \NewDocumentCommand{\nextfloatfoot}{s o o o m m m m m}{}

\newmarkset {⟨markset⟩}
28 \newcommand{\newmarkset}[1]{}

\newextramark * {⟨markset⟩} {⟨macro-name⟩}
29 \NewDocumentCommand{\newextramarkset}{s m m}{}

\botextramarks {⟨markset⟩}
30 \newcommand{\botextramarks}[1]{}

\topextramarks {⟨markset⟩}
31 \newcommand{\topextramarks}[1]{}

\firstextramarks {⟨markset⟩}
32 \newcommand{\firstextramarks}[1]{}

\nextextramarks {⟨markset⟩}
33 \newcommand{\nexttopextramarks}[1]{}

\outerextramarks {⟨markset⟩}
34 \newcommand{\outerextramarks}[1]{}

\innerextramarks {⟨markset⟩}
35 \newcommand{\innerextramarks}[1]{}

```

File 319 **lwarf-titleref.sty**

§ 415 Package **titleref**

Pkg titleref titleref is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{titleref}

```

2
3 \LetLtxMacro{\titleref}{\nameref}
4
5 \providecounter{LWR@currenttitle}
6
7 \newcommand*{\currenttitle}{%
8     \addtocounter{LWR@currenttitle}{1}%
9     \label{currenttitle}\arabic{LWR@currenttitle}%
10    \nameref{currenttitle}\arabic{LWR@currenttitle}%
11 }
12
13 \newcommand*{\theTitleReference}[2]{}

```

File 320 **l warp-titlesec.sty**

§ 416 Package **titlesec**

(Emulates or patches code by JAVIER BEZOS.)

Pkg **titlesec** titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for l warp-titlesec:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{titlesec}

\titleref {\langle label-format\rangle}
2 \newcommand*{\titleref}{[1]{}}

\titleformat* {\langle command\rangle} {\langle format\rangle}

\titleformat {\langle command\rangle} [{\langle shape\rangle}] {\langle format\rangle} {\langle label\rangle} {\langle sep\rangle} {\langle begfore\rangle} [{\langle after\rangle}]
3 \newcommand\titleformat{%
4     \@ifstar{\ttl@format@s}{%
5         {\ttl@format@i}}}
6 \newcommand{\ttl@format@s}{[1]{}}
7 \NewDocumentCommand{\ttl@format@i}{m o m m m o}{}

\chaptename
8 \@ifundefined{@chapapp}{\let\chapapp\chaptename}{}
9 \newcommand\chaptename{\@chapapp}

\titlespacing * {\langle command\rangle} {\langle left\rangle} {\langle before\rangle} {\langle after\rangle} [{\langle right\rangle}]

```

```
10 \NewDocumentCommand{\titlespacing}{s m m m o}{}  
  
\filright  
11 \newcommand*{\filright}{}  
  
\filcenter  
12 \newcommand*{\filcenter}{}  
  
\filleft  
13 \newcommand*{\filleft}{}  
  
\fillast  
14 \newcommand*{\fillast}{}  
  
\filinner  
15 \newcommand*{\filinner}{}  
  
\filouter  
16 \newcommand*{\filouter}{}  
  
\wordsep  
17 \newcommand\wordsep{\fontdimen\tw@\font \oplus  
18   \fontdimen\thr@@\font \ominus \fontdimen4\font}  
  
\titleline * [<align>] {<material>}  
19 \NewDocumentCommand{\titleline}{s o m}{}  
  
\titlerule [<height>]  
20 \providecommand*\titlerule{\ifstar{\t@tl@row}{\t@tl@rule}}  
21 \newcommand*{\t@tl@rule}[1][]{  
22 \newcommand*{\t@tl@row}[2][]{  
  
\iftitlemeasuring {\<true>} {\<false>}  
23 \newcommand{\iftitlemeasuring}[2]{#2}  
  
\assignpagestyle {\<command>} {\<pagestyle>}
```

```

24 \newcommand{\assignpagestyle}[2]{#2}

\titleclass {\name} [\startlevel] {\class} [\cmd]
25 \NewDocumentCommand{\titleclass}{m o m o}

```

File 321 **l warp-titletoc.sty**

§ 417 Package **titletoc**

(Emulates or patches code by JAVIER BEZOS.)

Pkg **titletoc** titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for l warp-titletoc:

```

for HTML output: 1 \LWR@ProvidesPackageDrop{titletoc}

\dottedcontents {\section} [\left] {\above} {\label} {\leader}
2 \NewDocumentCommand{\dottedcontents}{m o m m m}{}{}

\titlecontents * {\section} [\left] {\above} {\numbered} {\numberless} {\filler} [\below
or begin] [\separator] [\end]
3 \newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
4 \NewDocumentCommand{\ttl@tcstar}{m o m m m o o o}){}
5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m o}{}{}

\contentsmargin [{correction}] {\right}
6 \newcommand{\contentsmargin}[2][]{}

\thecontentslabel
7 \newcommand*{\thecontentslabel}{thecontentslabel}

\thecontentspage
8 \newcommand*{\thecontentspage}{thecontentspage}

\contentslabel [{format}] {\space}
9 \newcommand{\contentslabel}[2][]{\thecontentslabel}

```

```
\contentspage  [<format>]
 10 \newcommand{\contentspage}[1][]{\thecontentspage}

\contentspush  [{text}]
 11 \newcommand{\contentspush}[1]{}

\contentsuse   {{name} {text}}
 12 \newcommand{\contentsuse}[2]{}

\startcontents [{name}]
 13 \newcommand*{\startcontents}[1][]{}

\stopcontents  [{name}]
 14 \newcommand*{\stopcontents}[1][]{}

\resumecontents [{name}]
 15 \newcommand*{\resumecontents}[1][]{}

\printcontents [{name} {prefix} {start} {code}]
 16 \newcommand{\printcontents}[4][]{}

\startlist    [{name} {list}]
 17 \newcommand{\startlist}[2][]{}

\stoplist     [{name} {list}]
 18 \newcommand{\stoplist}[2][]{}

\resumelist   [{name} {list}]
 19 \newcommand{\resumelist}[2][]{}

\printlist    [{name} {list} {prefix} {code}]
 20 \newcommand{\printlist}[4][]{}
```

File 322 **l warp-titling.sty**

§ 418 Package **titling**

(Emulates or patches code by PETER WILSON.)

Pkg **titling**

package support l warp supports the native L^AT_EX titling commands, and also supports the packages authblk and titling. If both are used, authblk should be loaded before titling.
⚠ **load order**

\published and **\subtitle** If using the titling package, additional titlepage fields for \published and \subtitle may be added by using \AddSubtitlePublished in the preamble. See section 65.8.

The various titling footnote restyling commands have no effect.

Pass all options to l warp-titling:

for HTML output: 1 \LWR@ProvidesPackagePass{titling}

\@bsmtitleempty Patch \@bsmtitleempty:

2 \let\LWR@orig@bsmtitleempty\@bsmtitleempty
3 \renewcommand*{\@bsmtitleempty}{%
4 \LWR@orig@bsmtitleempty%
5 }

\keepthetitle Patch \keepthetitle:

6 \let\LWR@origkeepthetitle\keepthetitle
7 \renewcommand*{\keepthetitle}{%
8 \LWR@orig@keepthetitle%
9 }

\killtitle Patch \killtitle:

10 \let\LWR@origkilltitle\killtitle
11 \renewcommand*{\killtitle}{%
12 \LWR@orig@killtitle%
13 }

Env **titlingpage**

14 \renewenvironment*{titlingpage}{%
15 {%

Start an HTML titlepage div:

```
16 \LWR@printpendingfootnotes
17 \begin{titlepage}
```

Prepare for a custom version of `\maketitle` inside the `titlingpage`:

```
18 \LWR@maketitlesetup
19 \let\maketitle\LWR@titlingmaketitle
20 }
21 {
```

At the end of the environment, end the HTML titlepage div:

```
22 \end{titlepage}
23 }
```

Patch the pre/post title/author/date to add HTML tags, then initialize:

```
24
25 \pretitle{}
26 \posttitle{}
27
28 \preauthor{}
29 \postauthor{}
30
31 \predate{}
32 \postdate{}
```

`\LWR@maketitlesetup` Patches `\thanks` macros.

```
33 \renewcommand*\LWR@maketitlesetup{%
```

Redefine the footnote mark:

```
34 \def\makefnmark{\textsuperscript{\@thefnmark}}
```

```
\thefootnote \Rightarrow \nameuse{arabic}{footnote}, or
\thefootnote \Rightarrow \nameuse{fnsymbol}{footnote}
```

Redefine the footnote text:

```
35 \long\def\makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
36 \makethanksmark~%
      \makethanksmark \thanksfootmark \tamark \t@f@nmark \itshape a (or similar)
```

Print the text:

```
37 ##1%
38 }% \makefntext
39 }
```

```
\thanksfootmark
40 \renewcommand{\thanksfootmark}{%
41 % \hb@xt@{\thanksmarkwidth}{\hfil\normalfont%
42 % \thanksscript{%
43 % \thanksfootpre \tamark \thanksfootpost}%
44 % }%
45 % }%
46 }
```

`\maketitle` HTML mode. Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

```
47 \renewcommand*{\maketitle}{%
```

An HTML titlepage <div> is used for all classes.

```
48 \begin{titlepage}
```

Select which kind of footnote marks to use:

```
49 \@bsmarkseries
```

Set up special patches:

```
50 \LWR@\maketitlesetup
```

Typeset the title, etc:

```
51 \@maketitle
```

Immediately generate any \thanks footnotes:

52 \@thanks

Close the HTML titlepage div:

53 \end{titlepage}

Reset the footnote counter:

54 \@bscontmark
55 }

\@maketitle Typesets the title, etc. Patched for HTML.

```
56 \DeclareDocumentCommand{\@maketitle}{}
57     \maketitlehooka
58     {
59         \LWR@stoppars\LWR@htmlltag{\LWR@tagtitle}%
60         \@bspretitle \@title \@bsposttitle%
61         \LWR@htmlltag{\LWR@tagtitleend}\LWR@startpars%
62     }
63     \maketitlehookb
64     {
65         \begin{BlockClass}{author}
66         \renewcommand{\and}{%
67             \end{BlockClass}%
68             \begin{BlockClass}{oneauthor}%
69         }
70         \begin{BlockClass}{oneauthor}%
71         \@bspreauthor \@author \@bspostauthor%
72         \end{BlockClass}%
73         \end{BlockClass}%
74     }
75     \maketitlehookc
76     {
77         \begin{BlockClass}{titledate}%
78         \@bspredate \@date \@bspostdate%
79         \end{BlockClass}%
80     }
81     \maketitlehookd
82 }
```

\LWR@titlingmaketitle \maketitle for use inside an HTML titlingpage environment.

83 \renewcommand*{\LWR@titlingmaketitle}{%

Keep pending footnotes out of the title block:

84 \@thanks

Select which kind of footnote marks to use:

85 \bsmarkseries

Set up special patches:

86 \LWR@maketitlesetup

Typeset the title, etc:

87 \maketitle

Immediately generate any \thanks footnotes:

88 \@thanks

Reset the footnote counter:

89 \bscontmark
90 }

\thanksmarkseries {*series*}

Sets the type of footnote marks used by \thanks, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc.

91 \renewcommand{\thanksmarkseries}[1]{%
92 \def\bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
93 }

Set default titlepage thanks footnote marks. See section 65.7.

94 \ifclassloaded{memoir}{
95 \thanksmarkseries{arabic}
96 }{ % not memoir
97 \if@titlepage
98 \thanksmarkseries{arabic}
99 \else
100 \thanksmarkseries{fnsymbol}
101 \fi
102 }{ % not memoir

File 323 **l warp-tocbasic.sty**

§ 419 Package **tocbasic**

(Emulates or patches code by MARKUS KOHM.)

Pkg **tocbasic** tocbasic is patched for use by l warp.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. \DeclareDocumentCommand is used to overwrite the koma-script definitions.

for HTML output:

```
1 \LWR@ProvidesPackagePass{tocbasic}

2 \DeclareDocumentCommand{\usetocbasicnumberline}{o}{}{}
3 \DeclareDocumentCommand{\DeclareTOCStyleEntry}{o m m}{}{}
4 \DeclareDocumentCommand{\DeclareTOCEntryStyle}{m o m}{}{}
5 \DeclareDocumentCommand{\DefineTOCEntryOption}{m o m}{}{}
6 \DeclareDocumentCommand{\DefineTOCEntryBooleanOption}{m o m m m}{}{}
7 \DeclareDocumentCommand{\DefineTOCEntryCommandOption}{m o m m m}{}{}
8 \DeclareDocumentCommand{\DefineTOCEntryIfOption}{m o m m m}{}{}
9 \DeclareDocumentCommand{\DefineTOCEntryLengthOption}{m o m m m}{}{}
10 \DeclareDocumentCommand{\DefineTOCEntryNumberOption}{m o m m m}{}{}
11 \DeclareDocumentCommand{\CloneTOCEntryStyle}{m m}{}{}
12 \DeclareDocumentCommand{\TOCEntryStyleInitCode}{m m}{}{}
13 \DeclareDocumentCommand{\TOCEntryStyleStartInitCode}{m m}{}{}
```

File 324 **l warp-tocbibind.sty**

§ 420 Package **tocbibind**

(Emulates or patches code by PETER WILSON.)

Pkg **tocbibind** tocbibind is patched for use by l warp.

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

Pkg **makeidx** **Inline, with a manual toc entry:**

A commonly-used method to introduce an index in a L^AT_EX document:

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

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

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

Pkg **tocbibind** **Inline, with an automatic TOC entry:**

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

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

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

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

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

Other packages, such as **imakeidx**, may also have options for including the index in the Table of Contents.

for HTML output:

```

1 \let\simplechapterdelim\relax
2
3 \LWR@ProvidesPackagePass{tocbibind}

4 \renewenvironment{theindex}%
5 {%
6     \if@bibchapter
7         \if@donumindex
8             \chapter{\indexname}
9         \else
```

```

10      \if@dotocind
11          \chapter*{\indexname}
12          \addcontentsline{toc}{chapter}{\LWR@isolate{\indexname}}
13      \else
14          \chapter*{\indexname}
15      \fi
16  \fi
17 \else
18     \if@donumindex
19         \section{\indexname}
20     \else
21         \if@dotocind
22             \section*{\indexname}
23             \addcontentsline{toc}{\@tocextra}{\LWR@isolate{\indexname}}
24         \else
25             \section*{\indexname}
26         \fi
27     \fi
28 \fi
29 \let\item\LWR@indexitem%
30 \let\subitem\LWR@indexsubitem%
31 \let\subsubitem\LWR@indexsubsubitem%
32 }{}}

```

The following code is shared by anonchap.

```

33 \DeclareDocumentCommand{\simplechapter}{O{\emptyset}}{%
34     \def\@chapcntformat##1{%
35         #1\csname the##1\endcsname\simplechapterdelim\quad%
36     }%
37 }
38
39 \DeclareDocumentCommand{\restorechapter}{}{%
40 \let\@chapcntformat\@secntformat%
41 }

```

File 325 **lwarf-tocenter.sty**

§ 421 Package **tocenter**

Pkg **tocenter** **tocenter** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{tocenter}

2 \NewDocumentCommand{\ToCenter}{s o m m} {}
3 \NewDocumentCommand{\FromMargins}{s o m m m m} {}

```

File 326 **lwarf-tocloft.sty**

§ 422 Package **tocloft**

(Emulates or patches code by PETER WILSON.)

Pkg **tocloft** tocloft is emulated. Most user options and macros are ignored and disabled.
\newlistof and \cftchapterprecis are supported.

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

Discard all options for lwarf-tocloft:

for HTML output: 1 \LWR@ProvidesPackageDrop{tocloft}

\tocloftpagestyle {\langle style\rangle}
2 \newcommand{\tocloftpagestyle}[1]{}

\cftmarktoc
3 \newcommand*{\cftmarktoc}{}

\cfttoctitlefont
4 \newcommand*{\cfttoctitlefont}{}

\cftaftertoctitle
5 \newcommand*{\cftaftertoctitle}{}

6 \newlength{\cftbeforetoctitleskip}
7 \newlength{\cftaftertoctitleskip}

\cftmarklof
8 \newcommand*{\cftmarklof}{}

\cftloftitlefont

```
9 \newcommand*{\cftloftitlefont}{}  
  
\cftafterloftitle  
10 \newcommand*{\cftafterloftitle}{}  
  
11 \newlength{\cftbeforeloftitleskip}  
12 \newlength{\cftafterloftitleskip}  
  
\cftmarklot  
13 \newcommand*{\cftmarklot}{}  
  
\cftlottitlefont  
14 \newcommand*{\cftlottitlefont}{}  
  
\cftafterlottitle  
15 \newcommand*{\cftafterlottitle}{}  
  
16 \newlength{\cftbeforelottitleskip}  
17 \newlength{\cftafterlottitleskip}  
  
\cftdot  
18 \providecommand*{\cftdot}{.}  
  
\cftdotsep  
19 \providecommand*{\cftdotsep}{1}  
  
\cftnodots  
20 \providecommand*{\cftnodots}{5000}  
  
\cftdotfill {<sep>}  
21 \providecommand{\cftdotfill}[1]{}  
  
\cftsetpnumwidth {<length>}  
22 \DeclareDocumentCommand{\cftsetpnumwidth}{m}{}  
  
\cftsetrmarg {<length>}
```

```

23 \DeclareDocumentCommand{\cftsetrmarg}{m}{}{}

\cftpnumalign {⟨alignment⟩}
24 \DeclareDocumentCommand{\cftpnumalign}{m}{}{}

25 \LWR@providelength{\cftparskip}

```

The part-related items are also provided by memoir:

```

26 \LWR@providelength{\cftbeforepartskip}
27 \LWR@providelength{\cftpartindent}
28 \LWR@providelength{\cftpartnumwidth}
29 \providecommand*\cftpartfont(){}
30 \providecommand*\cftpartpresnum(){}
31 \providecommand*\cftpartaftersnum(){}
32 \providecommand*\cftpartaftersnumb(){}
33 \providecommand*\cftpartleader(){}
34 \providecommand*\cftpartdotsep{1}
35 \providecommand*\cftpartpagefont(){}
36 \providecommand*\cftpartafterpnum(){}

```

memoir uses the full name “chapter” instead of “chap”:

```

37 \LWR@providelength{\cftbeforechapskip}
38 \LWR@providelength{\cftchapindent}
39 \LWR@providelength{\cftchapnumwidth}
40 \newcommand*\cftchapfont(){}
41 \newcommand*\cftchappresnum(){}
42 \newcommand*\cftchapaftersnum(){}
43 \newcommand*\cftchapaftersnumb(){}
44 \newcommand*\cftchapleader(){}
45 \newcommand*\cftchapdotsep{1}
46 \newcommand*\cftchappagefont(){}
47 \newcommand*\cftchapafterpnum(){}

```

The following do not appear in memoir:

```

48 \LWR@providelength{\cftbeforesecskip}
49 \LWR@providelength{\cftsecindent}
50 \LWR@providelength{\cftsecnumwidth}
51 \newcommand*\cftsecfont(){}
52 \newcommand*\cftsecpresnum(){}
53 \newcommand*\cftsecaftersnum(){}
54 \newcommand*\cftsecaftersnumb(){}
55 \newcommand*\cftsecleader(){}
56 \newcommand*\cftsecdotsep{1}
57 \newcommand*\cftsecpagefont(){}
58 \newcommand*\cftsecafterpnum(){}

```

```
59 \LWR@providelength{\cftbeforesubsecskip}
60 \LWR@providelength{\cftsubsecindent}
61 \LWR@providelength{\cftsubsecnumwidth}
62 \newcommand*{\cftsubsecfont}={}
63 \newcommand*{\cftsubsecpresnum}={}
64 \newcommand*{\cftsubsecaftersnum}={}
65 \newcommand*{\cftsubsecaftersnumb}={}
66 \newcommand*{\cftsubsecleader}={}
67 \newcommand*{\cftsubsecdotsep}{1}
68 \newcommand*{\cftsubsecpagefont}={}
69 \newcommand*{\cftsubsecafterpnum}={}

70 \LWR@providelength{\cftbeforesubsubsecskip}
71 \LWR@providelength{\cftsubsubsecindent}
72 \LWR@providelength{\cftsubsubsecnumwidth}
73 \newcommand*{\cftsubsubsecfont}={}
74 \newcommand*{\cftsubsubsecpresnum}={}
75 \newcommand*{\cftsubsubsecaftersnum}={}
76 \newcommand*{\cftsubsubsecaftersnumb}={}
77 \newcommand*{\cftsubsubsecleader}={}
78 \newcommand*{\cftsubsubsecdotsep}{1}
79 \newcommand*{\cftsubsubsecpagefont}={}
80 \newcommand*{\cftsubsubsecafterpnum}={}

81 \LWR@providelength{\cftbeforeparaskip}
82 \LWR@providelength{\cftpaindent}
83 \LWR@providelength{\cftpnumwidth}
84 \newcommand*{\cftpafont}={}
85 \newcommand*{\cftpapresnum}={}
86 \newcommand*{\cftpaaftersnum}={}
87 \newcommand*{\cftpaaftersnumb}={}
88 \newcommand*{\cftpalaader}={}
89 \newcommand*{\cftpadotsep}{1}
90 \newcommand*{\cftpapagefont}={}
91 \newcommand*{\cftpaafterpnum}={}

92 \LWR@providelength{\cftbeforesubparaskip}
93 \LWR@providelength{\cftsubparaindent}
94 \LWR@providelength{\cftsubparanumwidth}
95 \newcommand*{\cftsubparafont}={}
96 \newcommand*{\cftsubparapresnum}={}
97 \newcommand*{\cftsubparaaftersnum}={}
98 \newcommand*{\cftsubparaaftersnumb}={}
99 \newcommand*{\cftsubparalaader}={}
100 \newcommand*{\cftsubparadotsep}{1}
101 \newcommand*{\cftsubparapagefont}={}
102 \newcommand*{\cftsubparaafterpnum}={}

103 \LWR@providelength{\cftbeforefigskip}
```

```
104 \LWR@providelength{\cftfigindent}
105 \LWR@providelength{\cftfignumwidth}
106 \newcommand*{\cftfigfont}={}
107 \newcommand*{\cftfigpresnum}={}
108 \newcommand*{\cftfigaftersnum}={}
109 \newcommand*{\cftfigaftersnumb}={}
110 \newcommand*{\cftfigleader}={}
111 \newcommand*{\cftfigdotsep}{1}
112 \newcommand*{\cftfigpagefont}={}
113 \newcommand*{\cftfigafterpnum}={}

114 \LWR@providelength{\cftbeforesubfigskip}
115 \LWR@providelength{\cftsubfigindent}
116 \LWR@providelength{\cftsubfignumwidth}
117 \newcommand*{\cftsubfigfont}={}
118 \newcommand*{\cftsubfigpresnum}={}
119 \newcommand*{\cftsubfigaftersnum}={}
120 \newcommand*{\cftsubfigaftersnumb}={}
121 \newcommand*{\cftsubfigleader}={}
122 \newcommand*{\cftsubfigdotsep}{1}
123 \newcommand*{\cftsubfigpagefont}={}
124 \newcommand*{\cftsubfigafterpnum}={}

125 \LWR@providelength{\cftbeforetabskip}
126 \LWR@providelength{\cfttabindent}
127 \LWR@providelength{\cfttabnumwidth}
128 \newcommand*{\cfttabfont}={}
129 \newcommand*{\cfttabpresnum}={}
130 \newcommand*{\cfttabaftersnum}={}
131 \newcommand*{\cfttabaftersnumb}={}
132 \newcommand*{\cfttableader}={}
133 \newcommand*{\cfttabdotsep}{1}
134 \newcommand*{\cfttabpagefont}={}
135 \newcommand*{\cfttabafterpnum}={}

136 \LWR@providelength{\cftbeforesubtabskip}
137 \LWR@providelength{\cftsubtabindent}
138 \LWR@providelength{\cftsubtabnumwidth}
139 \newcommand*{\cftsubtabfont}={}
140 \newcommand*{\cftsubtabpresnum}={}
141 \newcommand*{\cftsubtabaftersnum}={}
142 \newcommand*{\cftsubtabaftersnumb}={}
143 \newcommand*{\cftsubtableader}={}
144 \newcommand*{\cftsubtabdotsep}{1}
145 \newcommand*{\cftsubtabpagefont}={}
146 \newcommand*{\cftsubtabafterpnum}={}

147 \DeclareDocumentCommand{\cftsetindents}{m m m}{}

---


```

```

148 \newcommand{\pagenumbersoff}[1]{}
149 \newcommand{\pagenumberson}[1]{}

\newlistentry  [within] {counter} {ext} {level-1}
150 \DeclareDocumentCommand{\newlistentry}{o m m m}
151 {%
152 \LWR@traceinfo{newlistentry #2 #3 #4}%
153 \IfValueTF{#1}%
154 {%
155   \c@ifundefined{c@#2}{%
156     \newcounter{#2}[#1]%
157     \expandafter\edef\csname the#2\endcsname{%
158       \expandafter\noexpand\csname the#1\endcsname.\noexpand\arabic{#2}%
159     }%
160   }{}%
161 }%
162 {%
163   \c@ifundefined{c@#2}{%
164     \newcounter{#2}%
165   }{}%
166 }%
167 \c@namedef{l@#2}##1##2{%
168   \hypertocfloat{1}{#2}{#3}{##1}{##2}%
169   \def\cftwhatismyname{#2} from memoir
170 }%
171 \expandafter\newlength\csname cftbefore#2skip\endcsname%
172 \expandafter\newlength\csname cft#2indent\endcsname%
173 \expandafter\newlength\csname cft#2numwidth\endcsname%
174 \c@namedef{cft#2font}{}%
175 \c@namedef{cft#2presnum}{}%
176 \c@namedef{cft#2aftersnum}{}%
177 \c@namedef{cft#2aftersnumb}{}%
178 \c@namedef{cft#2leader}{}%
179 \c@namedef{cft#2dotsep}{1}%
180 \c@namedef{cft#2pagefont}{}%
181 \c@namedef{cft#2afterpnum}{}%
182 \c@namedef{toclevel@#2}{#4}%
183 \c@namedef{cft#2fillnum}##1{}%
184 \LWR@traceinfo{newlistentry done}%
185 }

\newlistof  [within] {type} {ext} {listofname}
Emulated through the \newfloat mechanism.

186 \DeclareDocumentCommand{\newlistof}{o m m m}
187 {%
188 \IfValueTF{#1}{%

```

```

189 {\newlistentry[#1]{#2}{#3}{0}}
190 {\newlistentry[#2]{#3}{0}}
191 \cnamedef{ext@#2}{#3}
192 \ifundefined{c@#3depth}{\newcounter{#3depth}}{}
193 \setcounter{#3depth}{1}
194 \cnamedef{cftmark#3}{}
195 \cnamedef{listof#2}{\listof{#2}{#4}}
196 \cnamedef{cftmake#3title}{}
197 \expandafter\newlength\csname cftbefore#3titleskip\endcsname
198 \expandafter\newlength\csname cftafter#3titleskip\endcsname
199 \cnamedef{cft#3titlefont}{}
200 \cnamedef{cftafter#3title}{}
201 \cnamedef{cft#3prehook}{}
202 \cnamedef{cft#3posthook}{}
203 }

\cftchapterprecis {text}
204 \newcommand{\cftchapterprecis}[1]{%
205   \cftchapterprecishere{#1}
206   \cftchapterprecistoc{#1}
207 \newcommand{\cftchapterprecishere}[1]{%
208   \begin{quote}\textit{#1}\end{quote}}
209 \newcommand{\cftchapterprecistoc}[1]{%
210   \addtocontents{toc}{%
211     \begin{protect}\begin{quote}#1\end{protect}\end{quote}}
212   }
213 }
214 }

```

File 327 **lwarf-tocstyle.sty**

§ 423 Package **tocstyle**

Pkg **tocstyle** **tocstyle** is ignored.

⚠ Not fully tested! Please send bug reports!

for HTML output:

```

1 \LWR@ProvidesPackageDrop{tocstyle}

2 \newcommand*{\usetocstyle}[2][]{}
3 \newcommand*{\deactivatetocstyle}[1][]{}
4 \newcommand*{\reactivatetocstyle}[1][]{}
5 \NewDocumentCommand{\settocfeature}{o o m m}{}
6 \NewDocumentCommand{\settocstylefeature}{o m m}{}
7 \NewDocumentCommand{\newtocstyle}{o o m m}{}
8 \newcommand*{\aliastoc}[2]{}
```

```
9 \newcommand*{\showtoc}[2][]{}
10 \newcommand{\iftochasdepth}[4]{}
```

File 328 **l warp-todo.sty**

§ 424 Package **todo**

(Emulates or patches code by FEDERICO GARCIA.)

Pkg todo todo is patched for use by l warp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{todo}

2 \renewcommand\todoitem[2]{%
3   \refstepcounter{todo}%
4   \item[%
5     \HTMLunicode{2610} \quad
6     \ref{todopage}:\thetodo%
7   ] : {\todoformat\ifx#1\todomark\else\textbf{#1}\fi}#2%
8   \label{todolbl}:\thetodo}%
9 }%
10
11 \renewcommand\doneitem[2]{%
12   \stepcounter{todo}%
13   \item[%
14     \HTMLunicode{2611} \quad
15     \ref{todopage}:\thetodo%
16   ] \nameuse{@done}{\the\c@todo}:
17   {\todoformat\ifx#1\todomark\else\textbf{#1}\fi}#2%
18 }
19
20 \xpatchcmd{\@displaytodo}
21   {\todoformat #1}{\todoformat \textbf{#1}}{}
22   {\PackageWarning{l warp-todo}{Unable to patch @displaytodo.}}
23
24 \xpatchcmd{\@displayfulltodo}
25   {\todoformat #1}{\todoformat \textbf{#1}}{}
26   {\PackageWarning{l warp-todo}{Unable to patch @displayfulltodo.}}
27
28 \patchcmd{\todoenv}{\itshape see text.}{\textit{see text.}}{}
29   {\PackageWarning{l warp-todo}{Unable to patch todoenv.}}
30
31 \patchcmd{\astodos}{\todoformat #1}{\todoformat \textbf{#1}}{}
32   {\PackageWarning{l warp-todo}{Unable to patch astodos.}}
33
34 \AtBeginDocument{
35 \crefname{todo}{todo}{todos}
```

```
36 \Crefname{todo}{Todo}{Todos}
37 }
```

File 329 **l warp-todonotes.sty**

§ 425 Package **todonotes**

(Emulates or patches code by HENRIK SKOV MIDTIBY.)

Pkg **todonotes** todonotes is emulated.

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

for HTML output:

```
1 \LWR@ProvidesPackagePass{todonotes}

2 \if@todonotes@disabled
3 \else
4
5 \newcommand{\ext@todo}{\tdo}
6
7 \renewcommand{\l@todo}[2]{\hypertocfloat{1}{todo}{\ldo}{#1}{#2}}

8 \let\LWRTODONOTES@orig@todototoc\todototoc
9
10 \renewcommand*{\todototoc}{%
11 \phantomsection%
12 \LWRTODONOTES@orig@todototoc%
13 }
14
15 \renewcommand{\@todonotes@drawMarginNoteWithLine}{%
16 \fcolorbox
17   {\@todonotes@currentbordercolor}
18   {\@todonotes@currentbackgroundcolor}
19   {\arabic{@todonotes@numberoftodonotes}}
20 \marginpar{\@todonotes@drawMarginNote}
21 }
22
23 \renewcommand{\@todonotes@drawInlineNote}{%
24 \fcolorboxBlock%
25   {\@todonotes@currentbordercolor}%
26   {\@todonotes@currentbackgroundcolor}%
27   {%
28     \if@todonotes@authorgiven%
29       {\@todonotes@author:\,}%

```

```
30      \fi%
31      \@todonotes@text%
32  }%
33 }%
34
35 \renewcommand{\@todonotes@drawMarginNote}{%
36     \if@todonotes@authorgiven%
37         \@todonotes@author\par%
38     \fi%
39     \arabic{@todonotes@numberoftodonotes}: %
40     \fcolorbox{%
41         {\@todonotes@currentbordercolor}%
42         {\@todonotes@currentbackgroundcolor}%
43     }{%
44         \@todonotes@sizecommand%
45         \@todonotes@text %
46     }%
47 }%
48
49 \renewcommand{\@todonotes@drawLineToRightMargin}{}
50
51 \renewcommand{\@todonotes@drawLineToLeftMargin}{}
52
53 \renewcommand{\missingfigure}[2][]{%
54 \setkeys{todonotes}{#1}%
55 \addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
56 \fcolorboxBlock{%
57     {\@todonotes@currentbordercolor}%
58     {\@todonotes@currentfigcolor}%
59 }{%
60     \setlength{\fboxrule}{4pt}%
61     \fcolorbox{red}{white}{Missing figure} \quad #2%
62 }
63 }
64
65 \LetLtxMacro{\LWRTODONOTES@orig}{\@todo}
66
67 \RenewDocumentCommand{\@todo}{o m}{%
68 \begingroup%
69 \renewcommand*{\phantomsection}{()}%
70 \IfValueTF{#1}{%
71     \LWRTODONOTES@orig[#1]{#2}%
72 }{%
73     \LWRTODONOTES@orig{#2}%
74 }
75 \endgroup%
76 }
77
78 \fi% \if@todonotes@disabled
```

File 330 **l warp-tram.sty**

§ 426 Package **tram**

Pkg **tram** **tram** is emulated.

- ⚠ **block only** The HTML emulation uses a `<div>`, which must not appear inside an HTML `` or an HTML paragraph. For this reason, the `tram` environment should only be used to contain paragraphs inside a `\parbox` or `minipage`. `tram` should not be used to mark up inline text.

To disable `tram`, allowing source compatibility with inline uses:

```
\begin{warpHTML}
\renewenvironment{tram}[1] [] {}{}
\end{warpHTML}
```

for HTML output: 1 `\LWR@ProvidesPackageDrop{tram}`

```
2 \newenvironment{tram}[1] []
3   {\BlockClass[background:#lightgray]{tram}}
4   {\endBlockClass}
```

File 331 **l warp-transparent.sty**

§ 427 Package **transparent**

(Emulates or patches code by HEIKO OBERDIEK.)

Pkg **transparent** Emulated. `\texttransparent` works for inline objects. `\transparent` only works for `\includegraphics`.

- ⚠ **Not X_ELa_TE_X!** Note that `transparent` does not work with X_ELa_TE_X.

for HTML output: Discard all options for l warp-transparent:

```
1 \LWR@ProvidesPackageDrop{transparent}

2 \newcommand*\transparent[1]{\edef\LWR@opacity{\#1}}
3
4 \newcommand*\texttransparent[2]{%
5 \begingroup%
```

```

6 \transparent{#1}%
7 \InlineClass[opacity: #1]{transparent}{#2}%
8 \endgroup%
9 }

```

File 332 **l warp-trimclip.sty**

§ 428 Package **trimclip**

Pkg **trimclip** **trimclip** is nullified.

for HTML output: 1 \LWR@ProvidesPackageDrop{trimclip}

The third argument, the text, is not touched. This allows `\bgroup / \egroup`, and verbatim content.

```

2 \csdef{trimbox}{\@ifstar\@gobble\@gobble}
3 \csletcs{trimbox*}{trimbox}
4 \def\endtrimbox{}
5 \csletcs{endtrimbox*}{endtrimbox}
6
7 \csletcs{clipbox}{trimbox}
8 \csletcs{clipbox*}{trimbox}
9 \csletcs{endclipbox}{endtrimbox}
10 \csletcs{endclipbox*}{endtrimbox}
11
12 \csletcs{marginbox}{trimbox}
13 \csletcs{marginbox*}{trimbox}
14 \csletcs{endmarginbox}{endtrimbox}
15 \csletcs{endmarginbox*}{endtrimbox}

```

File 333 **l warp-trivfloat.sty**

§ 429 Package **trivfloat**

(Emulates or patches code by JOSEPH WRIGHT.)

Pkg **trivfloat** **trivfloat** is forced to use the built-in **l warp** emulation for floats.

To create a new float type and change its name:

```

\trivfloat{example}
\renewcommand{\examplename}{Example Name}

```

```
\crefname{example}{example}{examples}
\Crefname{example}{Example}{Examples}
```

Discard all options for `lwarp-trivfloat`. This tells `trivfloat` not to use `floatrow` or `memoir`.

```
1 \LWR@ProvidesPackageDrop{trivfloat}
2 \LWR@origRequirePackage{trivfloat}
```

`\tfl@chapter@fix` Nullified at the beginning of the document. Is used by `trivfloat` to correct float chapter numbers, but is not needed for `lwarp`.

for HTML output:

```
3 \begin{warpHTML}
4 \AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}
5 \end{warpHTML}
```

§ 429.1 Combining `\newfloat`, `\trivfloat`, and `algorithmicx`

for HTML & PRINT:

For both print and HTML output:

⚠ When using `float`, `trivfloat`, or `algorithmicx` at the same time, be aware of conflicting file usage. `algorithmicx` uses `.loa`. `trivfloat` by default starts with `.loa` and goes up for additional floats, skipping `.lof` and `.lot`.

- ⚠ When using `\newfloat`, be sure to manually assign higher letters to the `\newfloat` files to avoid `.loa` used by `algorithmicx`, and any files used by `trivfloat`. Also avoid using `.lof` and `.lot`.
- ⚠ When using `\trivfloat`, you may force it to avoid conflicting with `algorithmicx` by starting `trivfloat`'s file extensions with `.lob`:

```
\makeatletter
\setcounter{tfl@float@cnt}{1} % start trivfloats with .lob
\makeatletter
```

```
7 \end{warpall}
```

File 334 **l warp-turnthepage.sty**

§ 430 Package **turnthepage**

Pkg **turnthepage** *turnthepage* is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{turnthepage}
2 \newcommand{\turnthepage}{}

---


```

File 335 **l warp-twoup.sty**

§ 431 Package **twoup**

Pkg **twoup** *twoup* is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{twoup}[2007/02/26]
2 \newcommand{\cleartolastpage}{}

---


```

File 336 **l warp-typearea.sty**

§ 432 Package **typearea**

(Emulates or patches code by MARKUS KOHM.)

Pkg **typearea** *typearea* is emulated.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. `\DeclareDocumentCommand` is used to overwrite the koma-script definitions.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{typearea}
2 \DeclareDocumentCommand{\typearea}{o m} {}
3 \DeclareDocumentCommand{\recalctypearea}{}{} {}
4 \@ifundefined{footheight}{\newlength\footheight} {}
5 \DeclareDocumentCommand{\areaset}{o m m} {}
6 \DeclareDocumentCommand{\activateareas}{}{}

---


```

```

7 \DeclareDocumentCommand{\storeareas}{m}{}%
8 \DeclareDocumentCommand{\BeforeRestoreareas}{s m}{}%
9 \DeclareDocumentCommand{\AfterRestoreareas}{s m}{}%
10 \DeclareDocumentCommand{\AfterCalculatingTypearea}{s m}{}%
11 \DeclareDocumentCommand{\AfterSettingArea}{s m}{}%

```

File 337 **lwarf-ulem.sty**

§ 433 Package **ulem**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg **ulem** Patched for use by lwarf.

for HTML output: Use the original package:

```
1 \LWR@ProvidesPackagePass{ulem}[2012/05/18]
```

Basic markup commands, using css:

```

2 \NewDocumentCommand{\LWR@HTML@uline}{+m}{}%
3     \InlineClass{%
4         (text-decoration:underline; text-decoration-skip: auto)%
5         {uline}{\LWR@isolate{#1}}%
6     }%
7 \LWR@formatted{uline}%
8
9 \NewDocumentCommand{\LWR@HTML@uuline}{+m}{}%
10    \InlineClass{%
11        (%
12            text-decoration:underline; text-decoration-skip: auto;%
13            text-decoration-style:double%
14        )%
15        {uuline}{\LWR@isolate{#1}}%
16    }%
17 \LWR@formatted{uuline}%
18
19 \NewDocumentCommand{\LWR@HTML@uwave}{+m}{}%
20    \InlineClass{%
21        (%
22            text-decoration:underline; text-decoration-skip: auto;%
23            text-decoration-style:wavy%
24        )%
25        {uwave}{\LWR@isolate{#1}}%
26    }%
27 \LWR@formatted{uwave}%
28

```

```
29 \NewDocumentCommand{\LWR@HTML@sout}{+m}{%
30     \InlineClass{%
31         (text-decoration:line-through)%
32         {sout}{\LWR@isolate{#1}}%
33     }%
34 \LWR@formatted{sout}%
35%
36 \NewDocumentCommand{\LWR@HTML@xout}{+m}{%
37     \InlineClass{%
38         (text-decoration:line-through)%
39         {xout}{\LWR@isolate{#1}}%
40     }%
41%
42 \NewDocumentCommand{\LWR@HTML@dashuline}{+m}{%
43     \InlineClass{%
44         (%
45             text-decoration:underline;%
46             text-decoration-skip: auto;%
47             text-decoration-style:dashed%
48         )%
49         {dashuline}{\LWR@isolate{#1}}%
50     }%
51 \LWR@formatted{xout}%
52%
53 \NewDocumentCommand{\LWR@HTML@dotuline}{+m}{%
54     \InlineClass{%
55         (%
56             text-decoration:underline;%
57             text-decoration-skip: auto;%
58             text-decoration-style: dotted%
59         )%
60         {dotuline}{\LWR@isolate{#1}}%
61     }%
62 \LWR@formatted{dotuline}
```

Nullified/emulated macros:

```
63 \NewDocumentCommand{\LWR@HTML@markoverwith}{m}{}
64 \LWR@formatted{markoverwith}%
65%
66 \NewDocumentCommand{\LWR@HTML@ULon}{+m}{\uline{#1}\egroup}%
67 \LWR@formatted{ULon}
```

File 338 **l warp-underscore.sty**

§ 434 Package **underscore**

Pkg **underscore** underscore is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{underscore}

File 339 **l warp-units.sty**

§ 435 Package **units**

(Emulates or patches code by AXEL REICHERT.)

Pkg **units** units is patched for use by l warp.

Values are not styled by css, and take the style of the surrounding HTML text.

Units are styled according to the print verison, so they will be forced to upright roman in HTML if the print version does so. It may be necessary to adjust the document's body css to match the print version.

for HTML output: 1 \LWR@ProvidesPackagePass{units}[1998/08/04]

```
2 \DeclareRobustCommand*\LWR@HTML@unit}[2] [] {%
3 \ifblank{#1}%
4   {\LWR@textcurrentfont{#2}}%
5   {%
6     #1%
7     \ifthenelse{\boolean{B@UnitsLoose}}{~}{\,,}%
8     \LWR@textcurrentfont{#2}%
9   }%
10 }%
11 \LWR@formatted{unit}

12 \DeclareRobustCommand*\LWR@HTML@unitfrac}[3] [] {%
13 \ifblank{#1}%
14   {%
15     \nicefrac{#2}{#3}%
16   }%
17   {%
18     #1%
```

```

19          \ifthenelse{\boolean{B@UnitsLoose}}{~}{\,}%
20          \nicefrac{#2}{#3}%
21      }%
22 }%
23
24 \LWR@formatted{unitfrac}

```

For Mathjax:

```

25 \CustomizeMathJax{\newcommand{\unit}[2][]{\#1 \#2}}
26 \CustomizeMathJax{\newcommand{\unitfrac}[3][]{\#1 \#2/\#3}}

```

File 340 **l warp-unitsdef.sty**

§ 436 Package **unitsdef**

(Emulates or patches code by PATRICK HAPPEL.)

Pkg **unitsdef** unitsdef is patched for use by l warp.

for HTML output:

```

1 \LWR@ProvidesPackagePass{unitsdef}

2 \renewcommand{\unitvaluesep}{,}
3
4 \renewcommand{\unittimes}{\@setunitsepfalse\HTMLunicode{22c5}\cdot}
5
6 \renewunit{\arcmin}{%
7   \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}{%
8     {\ensuremath{{}^{\prime}}}{%
9       {\HTMLunicode{2032}}\prime
10 }%
11
12 \renewunit{\arcsec}{%
13   \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}{%
14     {\ensuremath{{}^{\prime\prime}}}{%
15       {\HTMLunicode{2033}}\prime\prime
16 }%
17
18 \renewrobustcmd{\SI}[2]{%
19   \begingroup%
20     \let\unit@@xspace\relax%
21     \unitSIdf\selectfont%
22     \LWR@textcurrentfont{\#1\#2}\l warp
23   \endgroup%
24 }

```

File 341 **l warp-upref.sty**

§ 437 Package **upref**

Pkg upref Ignored.

for HTML output: Discard all options for l warp-upref:

```
1 \LWR@ProvidesPackageDrop{upref}
```

File 342 **l warp-url.sty**

§ 438 Package **url**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg url url is patched for use by l warp.

for HTML output: 1 \LWR@ProvidesPackagePass{url}

url uses math mode to print its string inside a group, so the original meaning of math is restored first.

```
2 \LetLtxMacro{\LWR@url}{\origUrl@FormatString}\Url@FormatString
3
4 \renewcommand*{\Url@FormatString}{%
5   \InlineClass{verbatim}{%
6     \LWR@restoreorigformatting%
7     \LWR@url{\origUrl@FormatString}%
8   }%
9 }
```

File 343 **l warp-uspace.sty**

§ 439 Package **uspace**

Pkg uspace uspace is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{uspace}

File 344 **lwarp-verse.sty**

§ 440 Package **verse**

(Emulates or patches code by PETER WILSON.)

Pkg **verse** **verse** is supported and patched by **lwarp**.

for HTML output: Pass all options for **lwarp-verse**:

```
1 \LWR@ProvidesPackagePass{verse}
```

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

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

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

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

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

Env **verse** The **verse** environment will be placed inside a **HTML <pre>**.

```
2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching verse.}
```

At the beginning of the `verse` environment:

```
4 \AtBeginEnvironment{verse}
5 {%
```

Use the original `list` environment inside a `<pre>` to attempt to preserve formatting.

```
6 \LWR@restoreoriglists%
```

Pkg <code>verse</code>	The <code>verse</code> or <code>memoir</code> packages can place stanza numbers to the left with their <code>\flagverse</code> command. Do not allow them to go into the left margin, which would cause <code>pdfcrop</code> to crop the entire page further to the left:
Cls <code>memoir</code>	
<code>\flagverse</code>	
Len <code>\vleftskip</code>	
7 \ifdef{\vleftskip}{% 8 \setlength{\vleftskip}{\HTMLvleftskip} 9 \setlength{\leftmargini}{\HTMLleftmargini} 10 }{}% 11 \LWR@forcenewpage 12 \LWR@atbeginverbatim{3}{verse}}% 13 }	

After the end of the `verse` environment, which places the `<pre>` tag at the regular left margin:

```
14 \AtEndEnvironment{verse}{%
15 \leavevmode%
16 \LWR@afterendverbatim{1}}%
17 }
```

Patch to place `poemtitle` inside an `HTML ` of class `poemtitle`:

```
18 \ifdef{\poemtitle}{%
19 \DeclareDocumentCommand{\vstypepoemtitle}{m}{%
20   \vspace{\beforepoemtitleskip}%
21   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
22   \vspace{\afterpoemtitleskip}%
23 }
24 }{}%
25
26 \LWR@traceinfo{Finished patching verse.}
27 }% AfterEndPreamble
```

File 345 **l warp-*versonotes*.sty**

§ 441 Package **versonotes**

(Emulates or patches code by NORMAN GRAY.)

Pkg **versonotes** versonotes is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{versonotes}

2 \newcommand{\versonote}[1]{\marginpar{#1}}
3 \newdimen\versotextwidth
4 \newdimen\versoleftmargin
5 \newcommand*\versolayout{}
```

File 346 **l warp-*vertbars*.sty**

§ 442 Package **vertbars**

(Emulates or patches code by PETER WILSON.)

Pkg **vertbars** vertbars is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{vertbars}

2 \newlength{\barwidth}
3 \setlength{\barwidth}{0.4pt}
4 \newlength{\barspace}
5 \setlength{\barspace}{1em}
6
7 \newenvironment{vertbar}{
8     \LWR@forcenewpage
9     \LWR@forceminwidth{\barwidth}
10    \begin{BlockClass}[
11        border-left: \LWR@printlength{\LWR@atleastonept} solid black ;
12        padding-left: \LWR@printlength{\barspace}%
13    ]{vertbar}
14 }{
15    \end{BlockClass}
16 }
```

File 347 **l warp-vmargin.sty**

§ 443 Package **vmargin**

Pkg **vmargin** **vmargin** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{vmargin}

2 \newcommand*{\LWRVM@customsize}[2]{}
3 \newcommand*{\setpapersize}[2][]{\ifstreq{\#2}{custom}{\LWRVM@customsize}{}}
4 \newcommand*{\setmargins}[8]{}
5 \newcommand*{\setmarginsrb}[8]{}
6 \newcommand*{\setmargnohf}[4]{}
7 \newcommand*{\setmargnohfrb}[4]{}
8 \newcommand*{\setmarg}[4]{}
9 \newcommand*{\setmargrb}[4]{}
10 \newlength{\PaperWidth}
11 \setlength{\PaperWidth}{8.5in}
12 \newlength{\PaperHeight}
13 \setlength{\PaperHeight}{11in}
14 \newif\ifLandscape

```

File 348 **l warp-vowel.sty**

§ 444 Package **vowel**

(Emulates or patches code by FUKUI REI.)

Pkg **vowel** **vowel** is patched for use by **l warp**.

This package has been tested with *pdflatex* and the Type 1 TIPA fonts using the following package load sequence:

```

\usepackage[T3,T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[noenc]{tipa}
\usepackage{vowel}

```

for HTML output:

```

1 \LWR@ProvidesPackagePass{vowel}

2 \renewenvironment{vowel}[1] []
3   {%

```

```

4      \begin{lateximage} [(-vowel--\packagediagramname)]%
5      \@vowel[#1]%
6    }
7  {%
8    \@@vowel%
9    \end{lateximage}%
10  }

```

File 349 **l warp-vpe.sty**

§ 445 Package **vpe**

Pkg vpe vpe is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{vpe}[2012/04/18]

File 350 **l warp-vwcol.sty**

§ 446 Package **vwcol**

(Emulates or patches code by WILL ROBERTSON.)

Pkg vwcol vwcol is patched for use with l warp.

The width option is ignored. All vwcol environments adjust to 1–3 equal-width columns, depending on the width of the browser window.

The remaining options are supported, except for lines and maxrecursion.

for HTML output: 1 \LWR@ProvidesPackagePass{vwcol}

Factored from \vwcol. Each is given a style tag to append to the final style.

```

\LWR@vwcol@addrule {<style tag>}

2 \newcommand*{\LWR@vwcol@addrule}[1]{%
3   \appto{\LWR@vwcolstyle}{%
4     #1: %
5     \LWR@printlength{\vwcol@rule} solid \LWR@origpound\LWR@vwcol@rulecolor ; %
6   }%
7 }

```



```

\LWR@vwcol@addrule {<style tag>}

```

```

8 \newcommand*{\LWR@vwcol@addgap}[1]{%
9     \appto{\LWR@vwcolstyle}{%
10        #1: %
11        \LWR@printlength{\vwcol@sep} ; %
12    }%
13 }

```

Env `vwcol` {*key/values*}

Redefine the environment to add a HTML style. The style is built depending on the required options.

```
14 \renewenvironment*{vwcol}[1] []{%
```

New paragraph, and process the options:

```
15 \par\noindent%
16 \vwcolsetup{#1}%

```

Begin with no style:

```
17 \newcommand*{\LWR@vwcolstyle}{}%
```

`presep` and `postsep` are created with HTML margins:

```

18 \if@vwcol@presep
19     \appto{\LWR@vwcolstyle}{margin-left: 1em ; padding-left: .5em ; }
20 \fi
21 \if@vwcol@postsep
22     \appto{\LWR@vwcolstyle}{margin-right: 1em ; padding-right: .5em ; }
23 \fi

```

`sep` becomes `column-gap`:

```

24 \ifdimgreater{\vwcol@sep}{1sp}{%
25     \LWR@vwcol@addgap{column-gap}
26     \LWR@vwcol@addgap{-moz-column-gap}
27     \LWR@vwcol@addgap{-webkit-column-gap}
28 }{}%

```

`rule` become `column-rule`, while `prerule` and `postrule` become HTML borders:

```

29 \convertcolorspec[named]{\vwcol@rulecol}{HTML}\LWR@vwcol@rulecolor%
30 \ifdimgreater{\vwcol@rule}{0pt}{%
31     \ifdimless{\vwcol@rule}{1pt}{%
32         \setlength{\vwcol@rule}{1pt}
33     }{}%
34     \LWR@vwcol@addrule{column-rule}
35     \LWR@vwcol@addrule{-moz-column-rule}
36     \LWR@vwcol@addrule{-webkit-column-rule}
37     \if@vwcol@prerule\LWR@vwcol@addrule{border-left}\fi
38     \if@vwcol@postrule\LWR@vwcol@addrule{border-right}\fi
39 }{}%

```

Each of the justify options becomes a `text-align`. Indentation is added where appropriate.

```

40 \ifdefequal{\vwcol@justify}{\RaggedRight}{
41     \appto{\LWR@vwcolstyle}{text-align: left ; }
42     \ifdimgreater{\vwcol@parindent}{0pt}{
43         \appto{\LWR@vwcolstyle}{%
44             text-indent: \LWR@printlength{\vwcol@parindent} ; %
45         }
46     }{}
47 }{}

48 \ifdefequal{\vwcol@justify}{\RaggedLeft}{
49     \appto{\LWR@vwcolstyle}{text-align: right ; }
50 }{}

51 \ifdefequal{\vwcol@justify}{\Centering}{
52     \appto{\LWR@vwcolstyle}{text-align: center ; }
53 }{}

54 \ifdefequal{\vwcol@justify}{\justifying}{
55     \appto{\LWR@vwcolstyle}{text-align: justify ; }
56     \ifdimgreater{\vwcol@parindent}{0pt}{
57         \appto{\LWR@vwcolstyle}{%
58             text-indent: \LWR@printlength{\vwcol@parindent} ; %
59         }
60     }{}
61 }{}

```

Create the `<div>` with the assembled style:

```

62 \BlockClass[\LWR@vwcolstyle]{multicols}
63 }

```

When the environment ends:

```

64 {
65 \endBlockClass
66 }

```

File 351 **lwarf-wallpaper.sty**

§ 447 Package **wallpaper**

(Emulates or patches code by MICHAEL H.F. WILKINSON.)

Pkg **wallpaper** **wallpaper** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{wallpaper}

```

2 \newcommand*{\CenterWallPaper}[2]{}
3 \newcommand*{\ThisCenterWallPaper}[2]{}
4 \newcommand*{\TileWallPaper}[3]{}
5 \newcommand*{\ThisTileWallPaper}[3]{}
6 \newcommand*{\TileSquareWallPaper}[2]{}
7 \newcommand*{\ThisTileSquareWallPaper}[2]{}
8 \newcommand*{\ULCornerWallPaper}[2]{}
9 \newcommand*{\ThisULCornerWallPaper}[2]{}
10 \newcommand*{\LLCornerWallPaper}[2]{}
11 \newcommand*{\ThisLLCornerWallPaper}[2]{}
12 \newcommand*{\URCornerWallPaper}[2]{}
13 \newcommand*{\ThisURCornerWallPaper}[2]{}
14 \newcommand*{\LRCornerWallPaper}[2]{}
15 \newcommand*{\ThisLRCornerWallPaper}[2]{}
16 \newcommand*{\ClearWallPaper}){}
17 \newlength{\wpXoffset}
18 \newlength{\wpYoffset}

```

File 352 **lwarf-watermark.sty**

§ 448 Package **Watermark**

(Emulates or patches code by ALEXANDER I. ROZHENKO.)

Pkg **watermark** watermark is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{watermark}

2 \newcommand{\watermark}[1]{}
3 \newcommand{\leftwatermark}[1]{}
4 \newcommand{\rightwatermark}[1]{}
5 \newcommand{\thiswatermark}[1]{}
6 \newcommand{\thispageheading}[1]{}

```

File 353 **lwarf-widows-and-orphans.sty**

§ 449 Package **widows-and-orphans**

Pkg **widows-and-orphans** widows-and-orphans is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{widows-and-orphans} [2018/09/01]

2 \NewDocumentCommand\WaOsetup{}{}
```

```
3 \NewDocumentCommand\WaOparameters{}{  
4 \NewDocumentCommand\WaOignorenext{}{}
```

File 354 l warp-wrapfig.sty

§ 450 Package **wrapfig**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg wrapfig wrapfig is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{wrapfig}  
  
2 \newcommand*\{\LWR@wrapposition}{}  
3  
4 \newcommand*\{\LWR@subwrapfigure}{[2]{%  
5 \renewcommand*\{\LWR@wrapposition}{}%  
6 \ifthenelse{  
7     \equal{\#1}{r}\OR\equal{\#1}{R}\OR%  
8     \equal{\#1}{o}\OR\equal{\#1}{O}%  
9 }%  
10 {\renewcommand*\{\LWR@wrapposition}{float:right}}%  
11 {\renewcommand*\{\LWR@wrapposition}{float:left}}%  
12 \setlength{\LWR@templengthone}{#2}%  
13 \LWR@BlockClassWP{  
14     width:\LWR@printlength{\LWR@templengthone}; \LWR@wrapposition; %  
15     margin:10pt%  
16 }%  
17 {  
18     width:\LWR@printlength{\LWR@templengthone}; \LWR@wrapposition%  
19 }%  
20 {marginblock}%  
21 }  
22  
23  
24 \NewDocumentEnvironment{wrapfigure}{o m o m}  
25 {  
26 \LWR@subwrapfigure{#2}{#4}%  
27 \captionsetup{type=figure}%  
28 }  
29 {  
30 \endLWR@BlockClassWP%  
31 }  
32  
33  
34 \NewDocumentEnvironment{wraptable}{o m o m}  
35 {
```

```
36 \LWR@subwrapfigure{#2}{#4}%
37 \captionsetup{type=table}%
38 }
39 {%
40 \endLWR@BlockClassWP%
41 }
42
43
44 \NewDocumentEnvironment{wrapfloat}{m o m o m}
45 {%
46 \LWR@subwrapfigure{#3}{#5}%
47 \captionsetup{type=#1}%
48 }
49 {%
50 \endLWR@BlockClassWP%
51 }
52
53 \newlength{\wrapoverhang}
```

File 355 **l warp-xbmks.sty**

§ 451 Package **xbmks**

Pkg **xbmks** *xbmks* is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{xbmks}[2018/07/04]

```
2 \newcommand{\xbmksetup}[1]{}
3 \NewDocumentCommand{\pdfbookmarkx}{o m o m}{}%
4 \NewDocumentCommand{\currentpdfbookmarkx}{m o m}{}%
5 \NewDocumentCommand{\subpdfbookmarkx}{m o m}{}%
6 \NewDocumentCommand{\belowpdfbookmarkx}{m o m}{}%
```

File 356 **l warp-xcolor.sty**

§ 452 Package **xcolor**

(Emulates or patches code by DR. UWE KERN.)

Pkg **xcolor** *xcolor* is supported by l warp.

§ 452.1 Limitations

\colorboxBlock and \fcolorboxBlock	\colorboxBlock and \fcolorboxBlock are provided for increased HTML compatibility, and they are identical to \colorbox and \fcolorbox in print mode. In HTML mode they place their contents into a <div> instead of a . These <div>s are set to display: inline-block so adjacent \colorboxBlocks appear side-by-side in HTML, although text is placed before or after each.
	Print-mode definitions for \colorboxBlock and \fcolorboxBlock are created by lwarp's core if xcolor is loaded.
background: none	\fcolorbox and \fcolorboxBlock allow a background color of none, in which case only the frame is drawn, which can be useful for HTML.
color support	Color definitions, models, and mixing are fully supported without any changes required.
colored tables	\rowcolors is supported, except that the optional argument is ignored so far.
colored text and boxes	\textcolor, \colorbox, and \fcolorbox are supported.
\color and \pagecolor	\color and \pagecolor are ignored. Use css or \textcolor where possible.

§ 452.2 xcolor definitions: location and timing

The lwarp core and its lwarp-xcolor package are tightly integrated to allow comparable results for print, HTML, and print inside an HTML `lateximage`. This requires a number of definitions and redefinitions depending on whether each of xcolor and `lateximage` is being used, and whether print or HTML is being generated. Some of these actions are one-time when xcolor is loaded, and others are temporary as `lateximage` is used.

When xcolor is loaded in print mode: No special actions are taken at the time that xcolor is loaded in print mode, but see \AtBeginDocument below.

When lwarp-xcolor is loaded in HTML mode: xcolor's original definitions are saved for later restoration. \LWR@restoreorigformatting is appended to restore these definitions for use inside a `lateximage`. New HTML-mode definitions are created for \textcolor, \pagecolor, \nopagecolor, \colorbox, \colorboxBlock, \fcolorbox, \fcolorboxBlock, and fcolorminipage.

\AtBeginDocument in print or HTML mode: See Section 84. If xcolor has been loaded, the print-mode \fcolorbox is modified to accept a background color of none, and additional definitions are created for lwarp's new macros print-mode macros \colorboxBlock, \fcolorboxBlock, and fcolorminipage. The HTML versions of these macros will already have been created by lwarp-xcolor if it has been loaded.

For use inside an HTML `lateximage`, `\LWR@restoreorigformatting` is appended to temporarily set these functions to their print-mode versions.

In a `lateximage` in HTML mode: `\LWR@restoreorigformatting` temporarily restores the print-mode definitions of xcolor's functions. See `\LWR@restoreorigformatting` on page 523.

`\color:`

Print: Used as-is.

HTML: Ignored by `pdftotext`, and will not appear.

HTML lateximage: Colors will appear in a `lateximage`.

`\textcolor:`

Print: Used as-is.

HTML: Redefined by `l warp-xcolor`, page 994.

HTML lateximage: Remembers and reuses the print version.

`\pagecolor:`

Print: Used as-is.

HTML: Ignored.

HTML lateximage: Colors will be picked up in a `lateximage`.

`\nopagecolor:`

Print: Used as-is.

HTML: Ignored.

HTML lateximage: Colors will be picked up in a `lateximage`.

`\colorbox:`

Print: Used as-is.

HTML: Redefined by `l warp-xcolor`, page 995.

HTML lateximage: Remembers and reuses the print version.

`\colorboxBlock:`

Print: Becomes `\colorbox`.

HTML: Newly defined by `l warp-xcolor` to use a `<div>`, page 995.

HTML lateximage: Remembers and reuses the print version `\colorbox`.

`\fcolorbox:`

Print: Modified to allow a background of `none`.

`\LWR@print@fcolorbox` at section 84

HTML: Redefined by l warp-xcolor, page 996.

HTML lateximage: Remembers and reuses the print version.

\fcolorboxBlock:

Print: Becomes \fcolorbox. Section 84

HTML: Newly defined by l warp-xcolor to use a <div>, page 997.

HTML lateximage: Remembers and reuses the print version \fcolorbox.

fcolorminipage:

Print: Newly defined in the l warp core.

LWR@print@fcolorminipage at section 84

HTML: Newly defined by l warp-xcolor, page 998.

HTML lateximage: Uses the print version.

\boxframe:

Print: Used as-is.

HTML: Redefined by l warp-xcolor, page 999.

HTML lateximage: Remembers and reuses the print version.

§ 452.3 Package loading

```
for HTML output: 1 \LWR@ProvidesPackagePass{xcolor}
                  2 \begin{warpHTML}
```

§ 452.4 Remembering and restoring original definitions

Remember the following print-mode actions to be restored when inside a lateximage environment:

```
3 \LetLtxMacro{\LWR@print@pagecolor}{\pagecolor}
4 \LetLtxMacro{\LWR@print@nopagecolor}{\nopagecolor}
```

\LWR@restoreorigformatting Inside a lateximage the following gets restored to their print-mode actions:

```
5 \appto{\LWR@restoreorigformatting}{%
6 \LetLtxMacro{\pagecolor}{\LWR@print@pagecolor}%
7 \LetLtxMacro{\nopagecolor}{\LWR@print@nopagecolor}%
8 }
```

§ 452.5 **HTML color style**

Sets \LWR@tempcolor to the current color.

```
\LWR@findcurrenttextcolor
 9 \renewcommand*\{\LWR@findcurrenttextcolor\}{%
10 \protect\colorlet{\LWR@current@color}{.}%
11 \protect\convertcolorspec{named}{\LWR@current@color}{HTML}\LWR@tempcolor%
12 }
```

Prints a color style for the current color.

```
\LWR@currenttextcolorstyle
13 \newcommand*\{\LWR@currenttextcolorstyle\}{%
14 \LWR@findcurrenttextcolor%
15 \ifdefstring{\LWR@tempcolor}{000000}%
16 {}%
17 {color: \LWR@origpound\LWR@tempcolor ; }%
18 }
```

\LWR@textcurrentcolor {*text*} Like \textcolor but uses the current \color instead.

```
19 \DeclareDocumentCommand{\LWR@textcurrentcolor}{m}{%
20 \begingroup%
21 \LWR@FBcancel%
22 \LWR@findcurrenttextcolor%
23 \InlineClass [color:\LWR@origpound\LWR@tempcolor]{textcolor}{%
24     \renewcommand*\{\LWR@currenttextcolor\}{\LWR@origpound\LWR@tempcolor}%
25     #1%
26 }%
27 \endgroup%
28 }
```

\LWR@colorstyle {*2: model*} {*3: color*}

For a color style, prints the color converted to HTML colors.

```
29 \NewDocumentCommand{\LWR@colorstyle}{m m}{%
30 \begingroup%
31 \LWR@FBcancel%
```

Use the xcolor package to convert to an HTML color space:

```
32 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
```

Print the converted color:

```
33 \LWR@origpound\LWR@tempcolor%
34 \endgroup%
35 }
```

\LWR@backgroundcolor [*model*] {*color*} {*text*}

Similar to `\textcolor`, but prints black text against a color background.

Converted into an HTML hex color span.

```

36 \NewDocumentCommand{\LWR@backgroundcolor}{O{named} m m}{%
37 \begingroup%
38 \LWR@FBcancel%
39 \InlineClass[background:\LWR@colorstyle{#1}{#2}]{backgroundcolor}{%
40 #3%
41 }%
42 \endgroup%
43 }

```

§ 452.6 **HTML border**

`\LWR@borderpadding` {`<colorstyle>`} {`<color>`} Prints the HTML attributes for a black border and padding.
`\LWR@forceminwidth` must be used first in order to set the border width.

```

44 \newcommand*\LWR@borderpadding[2]{%
45 border:\LWR@printlength{\LWR@atleastonept} solid \LWR@colorstyle{#1}{#2} ; %
46 padding:\LWR@printlength{\fboxsep}%
47 }

```

§ 452.7 **High-level macros**

`\textcolor` [`<model>`] {`<color>`} {`<text>`}

Converted into an HTML hex color span.

```

48 \NewDocumentCommand{\LWR@HTML@textcolor}{o m m}{%
49 \begingroup%

```

Set the PDF color, to be picked up by SVG math if possible.

The print-mode `\color` command cannot accept the named option with color mixing,
but it works with no option at all.

```

50 \IfValueTF{#1}{%
51   \color[#1]{#2}%
52 }{%
53   \color{#2}%
54 }%
55 \LWR@FBcancel%
56 \IfValueTF{#1}{%
57 \InlineClass[color:\LWR@colorstyle{#1}{#2}]{textcolor}{%
58 \renewcommand*\LWR@currenttextcolor{\LWR@origpound\LWR@tempcolor}%
59 #3%

```

```
60 }%
61 }{%
62 \InlineClass [color:\LWR@colorstyle{named}{#2}]{textcolor}{%
63 \renewcommand*\{\LWR@currenttextcolor\}{\LWR@origpound\LWR@tempcolor}%
64 #3%
65 }%
66 }%
67 \endgroup%
68 }
69
70 \LWR@formatted{textcolor}
```

\pagecolor [*model*] {*color*}

Ignored. Use css instead.

```
71 \renewcommand*\{\pagecolor}[2] [named] {}
```

\nopagecolor Ignored.

```
72 \renewcommand*\{\nopagecolor\}{}  
  
\colorbox [model] {color} {text}
```

Converted into an HTML hex background color .

```
73 \NewDocumentCommand{\LWR@HTML@colorbox}{O{named} m +m}{%
74 \begingroup%
75 \LWR@FBcancel%
76 \InlineClass[%
77 background:\LWR@colorstyle{#1}{#2} ; %
78 padding:\LWR@printlength{\fboxsep}%
79 ]{\colorbox}{#3}%
80 \endgroup%
81 }
82
83 \AtBeginDocument{
84 \LWR@formatted{colorbox}
85 }
```

\colorboxBlock [*model*] {*color*} {*text*}

Converted into an HTML hex background color <div>.

```
86 \NewDocumentCommand{\LWR@HTML@colorboxBlock}{O{named} m +m}{%
```

```

87 \begingroup%
88 \LWR@FBcancel%
89 \LWR@stopars%
90 \begin{BlockClass}[%  

91 background:\LWR@colorstyle{\#1}{\#2} ; %  

92 padding:\LWR@printlength{\fboxsep}%
93 ]{\colorboxBlock}
94 #3
95 \end{BlockClass}%
96 \endgroup%

```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```

97 \global\booltrue{\LWR@minipagethispar}%
98 }
99
100 \AtBeginDocument{
101 \LWR@formatted{\colorboxBlock}
102 }

```

\fcolorbox [⟨framemodel⟩] {⟨framecolor⟩} [⟨boxmodel⟩] {⟨boxcolor⟩} {⟨text⟩}

Converted into a framed HTML hex background color span.

A background color of none creates a colored frame without a background color.

```

103 \NewDocumentCommand{\LWR@HTML\fcolorbox}{O{named} m O{named} m +m}{%
104 \LWR@traceinfo{HTML fcolorbox #2 #4}%
105 \begingroup%
106 \LWR@FBcancel%
107 \LWR@forceminwidth{\fboxrule}%
108 \ifthenelse{\equal{\#4}{none}}{%
109 % no background color
110   \InlineClass[%
111     \LWR@borderpadding{\#1}{\#2}%
112   ]{\fcolorbox}{\#5}%
113 }{%
114 % yes background color
115   \InlineClass[%
116     \LWR@borderpadding{\#1}{\#2} ; %
117     background:\LWR@colorstyle{\#3}{\#4}%
118   ]{\fcolorbox}{\#5}%
119 }%
120 \endgroup%

```

```

121 }
122
123 \AtBeginDocument{
124 \LWR@formatted{fcolorbox}
125 }

\fcolorboxBlock  [<framemodel>] {<framecolor>} [<boxmodel>] {<boxcolor>} {<text>}
Converted into a framed HTML hex background color span.

A background color of none creates a colored frame without a background color.

126 \NewDocumentCommand{\LWR@HTML\fcolorboxBlock}{O{named} m O{named} m +m}{%
127 \LWR@traceinfo{HTML fcolorboxBlock #2 #4}%
128 \begingroup%
129 \LWR@FBcancel%
130 \LWR@forceminwidth{\fboxrule}%

131 \LWR@stopars%

132 \ifthenelse{\equal{#4}{none}}{%
133 {%
134     \begin{BlockClass}[%]
135         \LWR@borderpadding{#1}{#2}%
136     ]\fcolorboxBlock
137     #5
138     \end{BlockClass}%
139 }%
140 {%
141     \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
142     \begin{BlockClass}[%]
143         background:\LWR@origpound\LWR@tempcolortwo\ ; %
144         \LWR@borderpadding{#1}{#2}%
145     ]\fcolorboxBlock
146     #5
147     \end{BlockClass}%
148 }%
149 \endgroup%

```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```

150 \global\booltrue{\LWR@minipagethispar}%
151 \LWR@traceinfo{HTML fcolorboxBlock done}%
152 }
153
154 \AtBeginDocument{

```

```
155 \LWR@formatted{fcolorboxBlock}
156 }
```

Creates a framed HTML <div> around its contents.

A print-output version is defined in the l warp core: section 84

```
\LWR@subfcolorminipage {<framemode>} {<framecolor>} {<background tag>} {<height>}
157 \NewDocumentCommand{\LWR@subfcolorminipage}{m m m m}{%
158 \LWR@stoppars%
159 \begin{BlockClass}[%%
160 #3%
161 \LWR@borderpadding{#1}{#2} ; %
162 \IfValueT{#4}{height:\LWR@printlength{\LWR@tempheight} ; }%
163 width:\LWR@printlength{\LWR@tempwidth}%
164 ]{fcolorminipage}%
165 }

Env fcolorminipage [<1:framemode>] [<2:framecolor>] [<3:boxmodel>] [<4:boxcolor>] [<5:align>] [<6:height>]
[<7:inner-align>] [<8:width>]

166 \NewDocumentEnvironment{\LWR@HTML@fcolorminipage}{O{named} m O{named} m O{c} o o m}
167 {%
168 \LWR@FBcancel%
169 \setlength{\LWR@tempwidth}{#8}%
170 \IfValueT{#6}{\setlength{\LWR@tempheight}{#6}}%
171 \LWR@forceminwidth{\fboxrule}%
172 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
173 \ifthenelse{\equal{#4}{none}}{%
174 {\LWR@subfcolorminipage{#1}{#2}{}}{#6}}{%
175 {%
176   \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
177   \LWR@subfcolorminipage{#1}{#2}{background:\LWR@origpound\LWR@tempcolortwo\ ; }{#6}%
178 }%
179 }%
180 {%
181   \end{BlockClass}%
}
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
182   \global\booltrue{\LWR@minipagethispar}%
183 }
184
185 \AtBeginDocument{
```

```

186 \LWR@formattedeenv{fcolorminipage}
187 }

```

```
\boxframe {<width>} {<height>} {<depth>}
```

The depth is added to the height, but the box is not descended below by the depth.
\textcolor is honored.

```

188 \newcommand*{\LWR@HTML@boxframe}[3]{%
189 {%
190 \setlength{\LWR@tempwidth}{#1}%
191 \setlength{\LWR@tempheight}{#2}%
192 \addtolength{\LWR@tempheight}{#3}%
193 \LWR@forceminwidth{\fboxrule}%
194 \LWR@findcurrenttextcolor%
195 \InlineClass[% 
196 display:inline-block ; %
197 border:\LWR@printlength{\LWR@atleastonept} solid \LWR@currenttextcolor{} ; %
198 width:\LWR@printlength{\LWR@tempwidth} ; %
199 height:\LWR@printlength{\LWR@tempheight}%
200 ]{boxframe}{}%
201 }%
202 }
203
204 \LWR@formattedeenv{boxframe}

```

§ 452.8 Row colors

```

\rowc@l@rs [<cmds>] {<startrow>} {<odd color>} {<even color>}
205 \newcommand*{\LWR@xcolor@tempcolor}{}%
206
207 \def\rowc@l@rs[#1]#2#3#4{%
208 {
209 \global\rownum=1
210   \global\@rowcolorstrue
211   \@ifempty{#3}{%
212     {\def\@oddrowcolor{\@norowcolor}%
213     \convertcolorspec[named]{#3}{HTML}\LWR@xcolor@tempcolor%
214     \edef\@oddrowcolor{%
215       \csdef{\LWR@xcolorrowHTMLcolor}{\LWR@xcolor@tempcolor}%
216     }%
217   }%
218   }%
219   \@ifempty{#4}{%
220     {\def\@evenrowcolor{\@norowcolor}%
221     \convertcolorspec[named]{#4}{HTML}\LWR@xcolor@tempcolor%
222   }%

```

```

223     \edef\@evenrowcolor{%
224         \csdef{LWR@xcolorrowHTMLcolor}{\LWR@xcolortempcolor}%
225     }%
226     }%
227     \if@rowcmd
228         \def\@rowcolors
229     {%
230     #1%
231         \if@rowcolors
232     \noalign{%
233             \relax\ifnum\rownum<#2\@norowcolor\else
234                 \ifodd\rownum\@oddrowcolor\else\@evenrowcolor\fi\fi%
235     }%
236         \fi%
237     }%
238     \else
239         \def\@rowcolors
240     {%
241         \if@rowcolors
242             \ifnum\rownum<#2%
243         \noalign{%
244             \@norowcolor
245     }%
246         \else
247     #1%
248         \noalign{%
249             \ifodd\rownum\@oddrowcolor\else\@evenrowcolor\fi%
250     }%
251         \fi
252         \fi%
253     }%
254     \fi
255     \ignorespaces%
256 }

```

`\@norowcolor` Turns off color for this row.

```

257 \def\@norowcolor{%
258 \renewcommand{\LWR@xcolorrowHTMLcolor}{}%
259 }

```

`\@rowcolors` Executed at the end of each row.

```

260 \def\@rowcolors{%
261     \noalign{%
262         \global\advance\rownum\@ne%
263     }%
264     \@rowcolors%
265 }

```

```
266 \end{warpHTML}
```

File 357 **lwarp-xexchangebar.sty**

§ 453 Package **xexchangebar**

Pkg **xexchangebar** **xexchangebar** is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{xexchangebar}[2017/08/03]
2 \LWR@origRequirePackage{lwarp-changebar}
```

File 358 **lwarp-xellipsis.sty**

§ 454 Package **xellipsis**

(Emulates or patches code by DONALD P. GOODMAN III.)

Pkg **xellipsis** **xellipsis** is patched for use by **lwarp**.

When non-zero, each of the spaces is converted to an HTML thin unbreakable space.

for HTML output:

```
1 \LWR@ProvidesPackagePass{xellipsis}

2 \newcommand*{\LWR@xellipsespace}[1]{%
3 \ifdim#1=0pt\else%
4     \ifdim#1<\fontdimen2\font%
5         ,%
6     \else%
7         ~%
8     \fi%
9 \fi%
10 }
11
12 \def\xelip{%
13 \mbox{%
14     \LWR@xellipsespace{\xeliprebef}%
15     \xeliprechar%
16     \LWR@xellipsespace{\xelipreaf}%
17     \LWR@xellipsespace{\xelipbef}%
18     \xelipchar%
19     \xel@loopi = 1%
20     \loop\ifnum\xelipnum>\xel@loopi%
21         \advance\xel@loopi by1%
22         \LWR@xellipsespace{\xelipgap}%
23     \repeat
24 }
```

```

23      \xelipchar%
24      \repeat%
25      \LWR@xellipsespace{\xelipaft}%
26      \LWR@xellipsespace{\xelippostbef}%
27      \xelippostchar%
28      \LWR@xellipsespace{\xelippoststaft}%
29 }%
30 }%

```

File 359 **l warp-xetexko-vertical.sty**

§ 455 Package **xetexko-vertical**

(Emulates or patches code by DOHYUN KIM.)

Pkg **xetexko-vertical** xetexko-vertical is patched for use by l warp.

for HTML output:

```

1 \LWR@loadbefore{xetexko-vertical}
2
3 \LWR@ProvidesPackagePass{xetexko-vertical}[2018/04/06]

4 \renewcommand{\verticallytypesetting}{}
5 \renewenvironment{vertical}[1]{\BlockClass{verticalrl}}{\endBlockClass}
6 \renewenvironment{horizontal}[1]{\BlockClass{horizontaltb}}{\endBlockClass}
7 \renewcommand{\vertlatin}[1]{#1}

```

File 360 **l warp-xfrac.sty**

§ 456 Package **xfrac**

(Emulates or patches code by THE LATEX3 PROJECT.)

Pkg **xfrac** Supported by adding xfrac instances.

for HTML output:

```
1 \LWR@ProvidesPackagePass{xfrac}
```

⚠ font size In the user's document preamble, l warp should be loaded after font-related setup. During HTML conversion, this font is used by l warp to generate its initial PDF output containing HTML tags, later to be converted by *pdftotext* to a plain text file. While the text may be in any font which *pdftotext* can read, the math is directly converted into SVG images using this same user-selected font. xfrac below is set for the Latin Modern (lmr) font. If another font is used, it may be desirable to redefine \xfracHTMLfontsize with a different em size.

\sfrac [⟨instance⟩] {⟨num⟩} [⟨sep⟩] {⟨denom⟩}

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. \scalebox is made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a `lateximage`, no adjustments are necessary.

for HTML & PRINT: 2 \begin{warpall}

\xfracHTMLfontsize User-redefinable macro which controls the font size of the fraction.

```
3 \newcommand*\{\xfracHTMLfontsize\}{.6em}

4 \end{warpall}
```

for HTML output: 5 \begin{warpHTML}

instances Instances of xfrac for various font choices:

Produce css for a small raised numerator and a small denominator.

Scaling is turned off so that `pdftotext` correctly reads the result.

```
6 \DeclareInstance{xfrac}{default}{text}{

7   numerator-format = {%
8     \begingroup%
9     \RenewDocumentCommand{\scalebox}{m o m}{##3}%
10    \InlineClass{numerator}{#1}\%,%
11    \endgroup%
12  },
13  denominator-format = {%
14    \begingroup%
15    \RenewDocumentCommand{\scalebox}{m o m}{##3}%
16    \InlineClass{denominator}{#1}\%
17    \endgroup%
18  },
```

For `pdftotext`, do not scale the text:

```
19   scaling = false
20 }
21
22 \DeclareInstance{xfrac}{lmr}{text}{

23   numerator-format = {%
24     \begingroup%
25     \RenewDocumentCommand{\scalebox}{m o m}{##3}%
26   },
```

```
26     \InlineClass{numerator}{#1}\,%
27     \endgroup%
28 },
29 denominator-format = {%
30     \begingroup%
31     \RenewDocumentCommand{\scalebox}{m o m}{{\#3}%
32     \InlineClass{denominator}{#1}%
33     \endgroup%
34 },
```

For *pdftotext*, do not scale the text:

```
35     scaling = false
36 }
37
38 \DeclareInstance{xfrac}{lmss}{text}{
39     numerator-format = {%
40         \begingroup%
41         \RenewDocumentCommand{\scalebox}{m o m}{{\#3}%
42         \InlineClass{numerator}{#1}\,%
43         \endgroup%
44 },
45     denominator-format = {%
46         \begingroup%
47         \RenewDocumentCommand{\scalebox}{m o m}{{\#3}%
48         \InlineClass{denominator}{#1}%
49         \endgroup%
50 },
```

For *pdftotext*, do not scale the text:

```
51     scaling = false
52 }
53
54 \DeclareInstance{xfrac}{lmtt}{text}{
55     numerator-format = {%
56         \begingroup%
57         \RenewDocumentCommand{\scalebox}{m o m}{{\#3}%
58         \InlineClass{numerator}{#1}\,%
59         \endgroup%
60 },
61     denominator-format = {%
62         \begingroup%
63         \RenewDocumentCommand{\scalebox}{m o m}{{\#3}%
64         \InlineClass{denominator}{#1}%
65         \endgroup%
66 },
```

For *pdftotext*, do not scale the text:

```
67      scaling = false  
68 }  
  
69 \end{warpHTML}
```

File 361 **l warp-xltabular.sty**

§ 457 Package **xltabular**

(Emulates or patches code by ROLF NIEPRASCHK, HERBERT VOSS.)

Pkg xltabular xltabular is emulated by l warp.

for HTML output:

- ⚠ **table numbering** At present, an xltabular without a caption or with only a \caption* may be mis-numbered in HTML, so it may be necessary to place at the end of the table:

```
\warpHTMLonly{\addtocounter{table}{-1}}  
  
1 \RequirePackage{tabularx}  
2  
3 \LWR@ProvidesPackageDrop{xltabular}  
4  
5 \DeclareDocumentEnvironment{xltabular}{o m m}  
6 {\longtable[#3]}  
7 {\endlongtable}
```

File 362 **l warp-xltxtra.sty**

§ 458 Package **xltxtra**

(Emulates or patches code by WILL ROBERTSON, JONATHAN KEW.)

Pkg xltxtra xltxtra is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{xltxtra}  
2 \RequirePackage{realscripts}  
3 \RequirePackage{metalogo}
```

```

4 \newcommand*\TeX@logo@spacing[6]{}
5
6 \newcommand*{\vfrac}[2]{%
7   \textsuperscript{#1}/\textsubscript{#2}%
8 }
9
10 \newcommand\namedglyph[1]{%
11   \tempcnta=\XeTeXglyphindex "#1"\relax
12   \ifnum\tempcnta>0
13     \XeTeXglyph\tempcnta
14   \else
15     \xxt@namedglyph@fallback{#1}%
16   \fi}
17
18 \newcommand\xxt@namedglyph@fallback[1]{[#1]}
19
20 \DeclareDocumentCommand{\showhyphens}{m}{}

```

File 363 **l warp-xmpincl.sty**§ 459 Package **xmpincl***(Emulates or patches code by MAARTEN SNEEP.)*

Pkg xmpincl Emulated.

for HTML output: Discard all options for l warp-xmpincl:

```

1 \LWR@ProvidesPackageDrop{xmpincl}

2 \newcommand*{\includeXmp}[1]{}

```

File 364 **l warp-xpiano.sty**§ 460 Package **xpiano***(Emulates or patches code by ENRICO GREGORIO.)*

Pkg xpiano xpiano is patched for use by l warp.

for HTML output:

```

1 \LWR@ProvidesPackagePass{xpiano}

2 \ExplSyntaxOn
3 \NewDocumentCommand{\LWR@print@keyboard}{ O{} m }

```

```

4 {
5 \xpiano_keyboard:nn { #1 } { #2 }
6 }
7
8 \NewDocumentCommand{\LWR@HTML@keyboard}{ O{}m }
9 {
10 \begin{lateximage}*
11   [(-xpiano-\packagediagramname{}: \detokenize\expandafter{#2})]
12   [\detokenize\expandafter{#1}]
13 \xpiano_keyboard:nn { #1 } { #2 }
14 \end{lateximage}
15 }
16 \ExplSyntaxOff
17
18 \LWR@formatted[keyboard]

```

File 365 lwarf-xpinyin.sty

§ 461 Package **xpinyin***(Emulates or patches code by SOBEN LEE.)*

Pkg xpinyin xpinyin is partly supported. \xpinyin and pinyinscope are nullified, but \pinyin works.

for HTML output:

```

1 \LWR@ProvidesPackagePass{xpinyin}[2018/01/28]

2 \RenewDocumentEnvironment{pinyinscope}{O{}}
3
4 \RenewDocumentCommand{\xpinyin}{s O{} m}{%
5   \IfBooleanTF{#1}{#3}{\@firstoftwo{#3}}%
6 }
7
8 \RenewDocumentCommand{\enablepinyin}{}{%

```

File 366 lwarf-xtab.sty

§ 462 Package **xtab***(Emulates or patches code by PETER WILSON.)*

Pkg xtab xtab is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{xtab}

```

 Misplaced alignment
tab character &

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

```
\StartDefiningTabulars
\tablefirsthead
...
\StopDefiningTabulars
```

See section 9.9.

 `latximage` supertabular and xtab are not supported inside a `latximage`.

```
2 \newcommand{\LWRXT@firsthead}{}%
3
4 \newcommand{\tablefirsthead}[1]{%
5   \long\gdef\LWRXT@firsthead{#1}%
6 }
7
8 \newcommand{\tablehead}[1]{}
9
10 \newcommand{\tablelasthead}[1]{}
11
12 \newcommand{\notablelasthead}{}%
13
14 \newcommand{\tabletail}[1]{}
15
16 \newcommand{\LWRXT@lasttail}{}%
17
18 \newcommand{\tablelasttail}[1]{%
19   \long\gdef\LWRXT@lasttail{#1}%
20 }

21 \newcommand{\tablecaption}[2][]{%
22   \long\gdef\LWRXT@caption{%
23     \ifblank{#1}{%
24       {\caption{#2}}%
25       {\caption[#1]{#2}}%
26     }%
27   }%
28
29 \let\topcaption\tablecaption
30 \let\bottomcaption\tablecaption

31 \newcommand*\LWRXT@caption{}%
32
33 \newcommand*\shrinkheight[1]{}%
34
35 \newcommand*\xentrystretch[1]{}%
36
37 \NewDocumentEnvironment{xtabular}{s o m}
```

```

38 {%
39 \LWR@traceinfo{xtabular}%
40 \table%
41 \LWRXT@caption%
42 \begin{tabular}{#3}%
43 \TabularMacro\ifdefvoid{\LWRXT@firsthead}%
44 {\LWR@getmynexttoken}%
45 {\expandafter\LWR@getmynexttoken\LWRXT@firsthead}%
46 }%
47 {%
48 \ifdefvoid{\LWRXT@lasttail}%
49 {}%
50 {%
51 \TabularMacro\ResumeTabular%
52 \LWRXT@lasttail%
53 }%
54 \end{tabular}%
55 \endtable%
56 \LWR@traceinfo{xtabular done}%
57 }
58
59 \NewDocumentEnvironment{mpxtabular}[s o m]
60 {\minipage{\linewidth}\xtabular{#3}}
61 {\endxtabular\endminipage}

```

File 367 **l warp-xunicode.sty**

§ 463 Package **xunicode**

Pkg xunicode Error if xunicode is loaded after l warp.

Patch l warp-xunicode, but also verify that is was loaded before l warp:

for HTML output:

```

1 \LWR@loadbefore{xunicode}%
2
3 \LWR@ProvidesPackagePass{xunicode}[2011/09/09]

```

\textcircled becomes a span with a rounded border. \providecommand is used to avoid conflict with textcomp.

```

4 \providecommand*\f{\LWR@HTML@textcircled}[1]{%
5   \InlineClass[border: 1px solid \LWR@currenttextcolor]{textcircled}{#1}%
6 }
7
8 \LWR@formatted{textcircled}

```

File 368 **l warp-xurl.sty**

§ 464 Package **xurl**

Pkg xurl xurl is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{xurl}

File 369 **l warp-xy.sty**

§ 465 Package **xy**

(Emulates or patches code by KRISTOFFER H. ROSE, ROSS MOORE.)

Pkg xy xy is patched for use by l warp.

⚠ **\xypolygon** \xypolygon must be used inside the xy environment, or inside \xy ... \endxy.

for HTML output: 1 \LWR@ProvidesPackagePass{xy}

```
2 \AtBeginDocument{
3
4 \preto{\xy}{\begin{ lateximage }[(-xy-\~\packagediagramname)]}
5 \appto{\endxy}{\end{ lateximage }}
6
7 \@ifundefined{xymatrix}{}{
8 \LetLtxMacro{\LWR@origxymatrix}{\xymatrix}
9
10 \renewcommand{\xymatrix}[1]{%
11 \begin{ lateximage }[(-xy- xymatrix \packagediagramname)]
12 \LWR@origxymatrix{#1}
13 \end{ lateximage }
14 }
15 }
16
17 \@ifundefined{xygraph}{}{
18 \LetLtxMacro{\LWR@origxygraph}{\xygraph}
19
20 \renewcommand{\xygraph}[1]{%
21 \begin{ lateximage }[(-xy- xygraph \packagediagramname)]
22 \LWR@origxygraph{#1}
23 \end{ lateximage }}
```

```
24 }
25 }
26
27 }
```

File 370 **lwarf-zhlineskip.sty**

§ 466 Package **zhlineskip**

Pkg **zhlineskip** **zhlineskip** is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{zhlineskip}[2018/11/30]

2 \newcommand*\SetTextEnvironmentSinglespace[1]{}
3 \newcommand*\RestoreTextEnvironmentLeading{}
4 \newcommand*\SetMathEnvironmentSinglespace[1]{}
5 \newcommand*\RestoreMathEnvironmentLeading{}
```

File 371 **lwarf-zwpagelayout.sty**

§ 467 Package **zwpagelayout**

(Emulates or patches code by ZDENĚK WAGNER.)

Pkg **zwpagelayout** **zwpagelayout** is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{zwpagelayout}

2 \def\noBboxes{}
3 \onlypreamble\noBboxes
4
5 \expandafter\ifx\csname definecolor\endcsname\relax \else
6   \definecolor{cmykblack}{cmyk}{0,0,0,1}
7   \definecolor{grblack}{gray}{0}
8 %   \ifzwpl@redefineblack
9 %     \definecolor{black}{cmyk}{0,0,0,1}\color{black}
10 %   \fi
11  \definecolor{cmykred}{cmyk}{0,1,1,0}
12  \definecolor{cmykgreen}{cmyk}{1,0,1,0}
13  \definecolor{cmykblue}{cmyk}{1,1,0,0}
14  \definecolor{rgbred}{rgb}{1,0,0}
15  \definecolor{rgbgreen}{rgb}{0,1,0}
16  \definecolor{rgbbblue}{rgb}{0,0,1}
17 %  \ifzwpl@redefinetocmyk
```

```
18 \%      \definecolor{red}{cmyk}{0,1,1,0}
19 \%      \definecolor{green}{cmyk}{1,0,1,0}
20 \%      \definecolor{blue}{cmyk}{1,1,0,0}
21 \%  \fi
22 \fi
23
24 \let\OverprintTeXTeXtGState\relax
25
26 \DeclareRobustCommand\SetOverprint{\ignorespaces}
27 \DeclareRobustCommand\SetKnockout{\ignorespaces}
28 \DeclareRobustCommand\textoverprint[1]{{\SetOverprint#1}}
29 \DeclareRobustCommand\textknockout[1]{{\SetKnockout#1}}
30
31 \def\SetPDFminorversion#1{}
32 \onlypreamble\SetPDFminorversion
33
34 \newcommand*\Vcorr(){}
35
36 \DeclareRobustCommand\vb[1][]{}
37 \NewDocumentCommand{\NewOddPage}{* o}{}%
38 \NewDocumentCommand{\NewEvenPage}{* o}{}%
39 \def\SetOddPageMessage#1{\gdef\ZW@oddwarning}
40 \def\SetEvenPageMessage#1{\gdef\Z@evenwarning}
41 \def\ZW@oddwarning{Empty page inserted}\let\ZW@evenwarning\ZW@oddwarning
42
43 \def\clap#1{#1}
44
45 \def\CropFlap{2in}
46 \def\CropSpine{1in}
47 \def\CropXSpine{1in}
48 \def\CropXtrim{.25in}
49 \def\CropYtrim{.25in}
50 \def\UserWidth{5in}
51 \def\UserLeftMargin{1in}
52 \def\UserRightMargin{1in}
53 \def\UserTopMargin{1in}
54 \def\UserBotMargin{1in}
55 \def\thePageNumber{\LWR@origpound\,,\arabic{page}}
56 \ifTeX
57 \def\ifcaseZWdriver{\ifcase2}
58 \else
59 \def\ifcaseZWdriver{\ifcase1}
60 \fi
61 \DeclareRobustCommand\ZWifdriver[2]{}
```

File 372 **lwarp-patch-komascript.sty**

§ 468 Package **patch-komascript**

Pkg **lwarp-patch-komascript** Patches for komascript classes.

lwarp loads this package when scrbook, scrartcl, or scrreprt classes are detected.

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

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

 Not fully tested! Please send bug reports!

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

for HTML output: 1 \ProvidesPackage{lwarp-patch-komascript}

typearea is emulated.

2 \RequirePackage{lwarp-typearea}

tocbasic is emulated.

3 \RequirePackage{lwarp-tocbasic}

scrextend patches most of the new macros.

4 \RequirePackage{lwarp-scrextend}

Indexing macros, simplified for lwarp:

```
5 \AtBeginDocument{  
6  
7 \renewcommand*{\idx@heading}{%  
8   \idx@@heading{\indexname}%"  
9 }  
10  
11 \renewenvironment{theindex}{%  
12   \idx@heading%"  
13   \index@preamble\par\nobreak
```

```

14     \let\item\LWR@indexitem%
15     \let\subitem\LWR@indexsubitem%
16     \let\subsubitem\LWR@indexsubsubitem%
17 }
18 {}
19
20 \renewcommand*\indexspace{}
21
22 }% AtBeginDocument

```

The `\minisec` is placed inside a `<div>` of class `minisec`.

```

23 \renewcommand*{\minisec}[1]{
24 \begin{BlockClass}{minisec}
25 #1
26 \end{BlockClass}
27 }

```

The part and chapter preambles are placed as plain text just after each heading.

```

28 \@ifundefined{setpartpreamble}{}{
29 \RenewDocumentCommand{\setpartpreamble}{o o +m}{%
30 \renewcommand{\part@preamble}{#3}%
31 }
32 }
33
34 \@ifundefined{setchapterpreamble}{}{
35 \RenewDocumentCommand{\setchapterpreamble}{o o +m}{%
36 \renewcommand{\chapter@preamble}{#3}%
37 }
38 }

```

Simple captions are used in all cases.

```

39 \LetLtxMacro{\captionbelow}{\caption
40 \LetLtxMacro{\captionabove}{\caption
41
42 \LetLtxMacro{\captionofbelow}{\captionof
43 \LetLtxMacro{\captionofabove}{\captionof
44
45 \RenewDocumentEnvironment{captionbeside}{o m o o o s}
46 {}
47 {%
48 \IfValueTF{#1}{%
49 {\caption[#1]{#2}}%
50 {\caption{#2}}%
51 }
52
53 \RenewDocumentEnvironment{captionofbeside}{m o m o o o s}

```

```

54 {}
55 {%
56 \IfValueTF{#2}{%
57 {\captionof{#1}{#2}{#3}}{%
58 {\captionof{#1}{#3}}{%
59 }{%
60 \RenewDocumentCommand{\setcapindent}{s m}{}{%
61 \renewcommand*{\setcaphanging}{}{%
62 \renewcommand*{\setcapwidth}[2]{}{}{%
63 \renewcommand*{\setcapdynwidth}[2]{}{}{%
64 \RenewDocumentCommand{\setcapmargin}{s o m}{}{%

```

File 373 **l warp-patch-memoir.sty**

§ 469 Package **patch-memoir**

(Emulates or patches code by PETER WILSON.)

Pkg **l warp-patch-memoir** Patches for memoir class.

⚠ Not fully tested! Please send bug reports!

l warp loads this package when the memoir class is detected.

⚠ **options clash** While emulating memoir, l warp pre-loads a number of packages (section 469.1). This can cause an options clash when the user's document later loads the same packages with options. To fix this problem, specify the options before loading l warp:

```

\documentclass{memoir}
...
\PassOptionsToPackage{options_list}{package_name}
...
\usepackage{l warp}
...
\usepackage{package_name}

\verb|footnote| is not supported.

\verb|newfootnoteseries|, etc. are not supported.

```

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

Poem numbering is not supported.

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

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

for HTML output: 1 \ProvidesPackage{l warp-patch-memoir}

§ 469.1 Packages

These are pre-loaded to provide emulation for many of memoir's functions. memoir pretends that `abstract`, etc. are already loaded, via its "emulated" package mechanism, but l warp is directly loading the "l warp-" version of each, which happens to avoid memoir's emulation system.

```

2 \RequirePackage{l warp-abstract}%
3 \RequirePackage{l warp-array}%
4 \RequirePackage{l warp-booktabs}%
5 % \RequirePackage{l warp-ccaption}%
6 \RequirePackage{l warp-changepage}%
7 \RequirePackage{l warp-crop}
8 \RequirePackage{l warp-dcolumn}%
9 \RequirePackage{l warp-enumerate}%
10 \RequirePackage{l warp-epigraph}%
11 \RequirePackage{l warp-fancyvrb}%
12 \RequirePackage{l warp-footmisc}%
13 \RequirePackage{l warp-framed}%
14 \RequirePackage{l warp-hanging}%
15 \RequirePackage{l warp-makeidx}%
16 \DisemulatePackage{moreverb}
17 \RequirePackage{l warp-moreverb}
18 \RequirePackage{l warp-mparhack}
19 \RequirePackage{l warp-needspace}%
20 \RequirePackage{l warp-nextpage}%
21 \RequirePackage{l warp-pagenote}%
22 \RequirePackage{l warp-parskip}
23 \RequirePackage{l warp-setspace}%
24 \RequirePackage{l warp-showidx}
25 \RequirePackage{l warp-subfigure}%
26 \makeindex

```

`subfigure` is emulated via `subfig`, which pre-defines `subfigure` and `subtable`, but memoir does not, so they must be tested for here:

```

27 \LetLtxMacro{\LWR@memorignewsubfloat}{\newsubfloat}
28 \RenewDocumentCommand{\newsubfloat}{O{} m}{%
29     @ifundefined{c@sub#2}{%
30         \LWR@memorignewsubfloat[#1]{#2}%

```

```

31      }{}%
32 }
33
34 \RequirePackage{lwarf-tabularx}%
35 \RequirePackage{lwarf-titling}%
36 % \RequirePackage{lwarf-toctocbind}%
37 \RequirePackage{lwarf-tocloft}%
38 \RequirePackage{lwarf-verse}%

```

§ 469.2 Preliminary setup

Bypass the memoir package mechanism:

```
39 \LetLtxMacro{\LWR@orig}{\mem@old}
```

memoir already set the page size to a default, so it must be forced large for lwarf's use, to avoid tag overflows off the page.

```

40 \setstocksize{190in}{20in}
41 \setlrmarginsandblock{2in}{2in}{*}
42 \setulmarginsandblock{1in}{1in}{*}

```

§ 469.3 Laying out the page

```

43 \renewcommand*{\stockavi}{}
44 \renewcommand*{\stockav}{}
45 \renewcommand*{\stockaiv}{}
46 \renewcommand*{\stockaiii}{}
47 \renewcommand*{\stockbvi}{}
48 \renewcommand*{\stockbv}{}
49 \renewcommand*{\stockbiv}{}
50 \renewcommand*{\stockbiii}{}
51 % \renewcommand*{\stockmetriccrownvo}{}% in docs but not in the package
52 \renewcommand*{\stockmlargecrownvo}{}
53 \renewcommand*{\stockmdemyvo}{}
54 \renewcommand*{\stockmsmallroyalvo}{}
55 \renewcommand*{\pageavi}{}
56 \renewcommand*{\pageav}{}
57 \renewcommand*{\pageaiv}{}
58 \renewcommand*{\pageaiii}{}
59 \renewcommand*{\pagebvi}{}
60 \renewcommand*{\pagebv}{}
61 \renewcommand*{\pagebiv}{}
62 \renewcommand*{\pagebiii}{}
63 % \renewcommand*{\pagemetriccrownvo}{}% in docs but not in the package
64 \renewcommand*{\pagemlargecrownvo}{}
65 \renewcommand*{\pagemdemyvo}{}
66 \renewcommand*{\pagemsallroyalvo}{}

```

67

```
68 \renewcommand*{\stockdbill}{}
69 \renewcommand*{\stockstatement}{}
70 \renewcommand*{\stockexecutive}{}
71 \renewcommand*{\stockletter}{}
72 \renewcommand*{\stockold}{}
73 \renewcommand*{\stocklegal}{}
74 \renewcommand*{\stockledger}{}
75 \renewcommand*{\stockbroadsheet}{}
76 \renewcommand*{\pagedbill}{}
77 \renewcommand*{\pagestatement}{}
78 \renewcommand*{\pageexecutive}{}
79 \renewcommand*{\pageletter}{}
80 \renewcommand*{\pageold}{}
81 \renewcommand*{\pagelegal}{}
82 \renewcommand*{\pageledger}{}
83 \renewcommand*{\pagebroadsheet}{}
84
85 \renewcommand*{\stockpottvo}{}
86 \renewcommand*{\stockfoolscapvo}{}
87 \renewcommand*{\stockcrownvo}{}
88 \renewcommand*{\stockpostvo}{}
89 \renewcommand*{\stocklargecrownvo}{}
90 \renewcommand*{\stocklargepostvo}{}
91 \renewcommand*{\stocksmalldemyvo}{}
92 \renewcommand*{\stockdemyvo}{}
93 \renewcommand*{\stockmediumvo}{}
94 \renewcommand*{\stocksmalldroyalvo}{}
95 \renewcommand*{\stockroyalvo}{}
96 \renewcommand*{\stocksperrovalvo}{}
97 \renewcommand*{\stockimperialvo}{}
98 \renewcommand*{\pagepottvo}{}
99 \renewcommand*{\pagefoolscapvo}{}
100 \renewcommand*{\pagecrownvo}{}
101 \renewcommand*{\pagepostvo}{}
102 \renewcommand*{\pagelargecrownvo}{}
103 \renewcommand*{\pagelargepostvo}{}
104 \renewcommand*{\pagesmalldemyvo}{}
105 \renewcommand*{\pagedemyvo}{}
106 \renewcommand*{\pagemediumvo}{}
107 \renewcommand*{\pagesmalldroyalvo}{}
108 \renewcommand*{\pageroyalvo}{}
109 \renewcommand*{\pagesperrovalvo}{}
110 \renewcommand*{\pageimperialvo}{}
111
112 \renewcommand*{\memfontfamily}{}
113 \renewcommand*{\memfontenc}{}
114 \renewcommand*{\memfontpack}{}
115
116 \renewcommand*{\anyptfilebase}{}
117 \renewcommand*{\anyptsiz}{10}
```

```
118
119 \renewcommand*{\setstocksize}[2]{}
120 \renewcommand*{\settrimmedsize}[3]{}
121 \renewcommand*{\settrims}[2]{}
122
123 % \newlength{\lxvchars}
124 % \setlength{\lxvchars}{305pt}
125 % \newlength{\xlvchars}
126 % \setlength{\xlvchars}{190pt}
127 \renewcommand*{\setxlvchars}[1]{}
128 \renewcommand*{\setlxvchars}[1]{}
129
130 \renewcommand*{\settypeblocksize}[3]{}
131 \renewcommand*{\setlrmargins}[3]{}
132 \renewcommand*{\setlrmarginsandblock}[3]{}
133 \renewcommand*{\setbinding}[1]{}
134 \renewcommand*{\setulmargins}[3]{}
135 \renewcommand*{\setulmarginsandblock}[3]{}
136 \renewcommand*{\setcolsepandrule}[2]{}
137
138 \renewcommand*{\setheadfoot}[2]{}
139 \renewcommand*{\setheaderspaces}[3]{}
140 \renewcommand*{\setmarginnotes}[3]{}
141 \renewcommand*{\setfootins}[2]{}
142 \renewcommand*{\checkandfixthelayout}[1][]{}
143 \renewcommand*{\checkthelayout}[1]{}
144 \renewcommand*{\fixthelayout}(){}
145 %
146 % \newlength{\stockheight}
147 % \newlength{\trimtop}
148 % \newlength{\trimedge}
149 % \newlength{\stockwidth}
150 % \newlength{\spinemargin}
151 % \newlength{\foremargin}
152 % \newlength{\uppermargin}
153 % \newlength{\headmargin}
154 %
155 \renewcommand*{\typeoutlayout}){}
156 \renewcommand*{\typeoutstandardlayout}){}
157 \renewcommand*{\settypeoutlayoutunit}[1]({})
158 \renewcommand*{\fixpdflayout}){}
159 \renewcommand*{\fixdvipslayout}){}
160
161 \renewcommand*{\medievalpage}[1][]{}
162 \renewcommand*{\isopage}[1][]{}
163 \renewcommand*{\semiisopage}[1][]{}
164
165 \renewcommand{\setpagebl}[3]{}}
166 \renewcommand{\setpageml}[3]{}}
167 \renewcommand{\setpagetl}[3]{}}
```

```

168 \renewcommand{\setpagetm}[3]{}
169 \renewcommand{\setpagetr}[3]{}
170 \renewcommand{\setpagemr}[3]{}
171 \renewcommand{\setpagebr}[3]{}
172 \renewcommand{\setpagebm}[3]{}
173 \renewcommand{\setpagecc}[3]{}

```

§ 469.4 Text and fonts

```

174 \let\miniscule\tiny
175 \let\HUGE\Huge
176
177 \renewcommand*{\abnormalparskip}[1]{}
178 \renewcommand*{\nonzeroparskip}(){}
179 \renewcommand*{\traditionalparskip}={}
180
181 \let\onelinekip\baselineskip
182
183 \let\OnehalfSpacing\onehalfspacing
184 \let\DoubleSpacing\doublespacing
185 \renewcommand*{\setPagenoteSpacing}[1]{}
186 \renewcommand*{\setFloatSpacing}[1]{}
187 \let\SingleSpacing\singlespacing
188 \let\setSingleSpace\SetSinglespace
189 \let\SingleSpace\singlespace
190 \let\endSingleSpace\endsinglespace
191 \let\Spacing\spacing
192 \let\endSpacing\endspacing
193 \let\OnehalfSpace\onehalfspace
194 \let\endOnehalfSpace\endonehalfspace
195 \csletcs{OnehalfSpace*}{onehalfspace}
196 \csletcs{endOnehalfSpace*}{endonehalfspace}
197 \let\DoubleSpace\doublespace
198 \let\endDoubleSpace\enddoublespace
199 \csletcs{DoubleSpace*}{doublespace}
200 \csletcs{endDoubleSpace*}{enddoublespace}
201 \renewcommand*{\setDisplayskipStretch}[1]{}
202 \renewcommand*{\memdisplayskipstretch}){}
203 \renewcommand*{\noDisplayskipStretch}){}
204 \renewcommand*{\memdisplayskips}){}
205
206 \renewcommand*{\midsloppy}){}
207 \renewenvironment*{\midsloppypar}{}{}
208
209 \renewcommand*{\sloppybottom}){}

```

§ 469.5 Titles

```

210 \csletcs{titlingpage*}{titlingpage}

```

```

211 \csletcs{endtitlingpage*}{endtitlingpage}
212 \let\titlingpageend\relax
213 \newcommand{\titlingpageend}[2]{}
214 \let\andnext\and
215 \renewcommand*{\thanksmarkstyle}[1]{}
216 \renewcommand{\thanksfootmark}{%
217 \thanksscript{\tamarck}%
218 }
219
220 % \newlength{\thanksmarksep}
```

§ 469.6 Abstracts

```

221 \renewcommand*{\abstractcol}{}
222 \renewcommand*{\abstractintoc}{}
223 \renewcommand*{\abstractnum}{}
224 \renewcommand*{\abstractrunin}{}
```

§ 469.7 Document divisions

```

225
226 \def\@apppage{%
227   \part*{\appendixpagename}
228 }
229 \renewcommand\mempreaddappagetotohook{}
230 \renewcommand\mempostaddappagetotohook{}
231
232 \def\@sappage{%
233   \part*{\appendixpagename}
234 }
235
236 \csletcs{frontmatter*}{frontmatter}
237 \csletcs{mainmatter*}{mainmatter}
238 \renewcommand*{\raggedbottomsection}{}
239 \renewcommand*{\normalbottomsection}{}
240 \renewcommand*{\bottomsections skip}{}
241 \renewcommand*{\bottomsectionpenalty}{}
242 \csletcs{appendixpage*}{appendixpage}
243 \renewcommand*{\namedsubappendices}{}
244 \renewcommand*{\unnamedsubappendices}{}
245 \renewcommand*{\setsecnumdepth}[1]{% todo tocvssec2
246 \renewcommand*{\maxsecnumdepth}[1]{% todo tocvssec2
247 \renewcommand*{\beforebookskip}{}
248 \renewcommand*{\afterbookskip}{}
249 \renewcommand*{\beforeparts skip}{}
250 \renewcommand*{\afterparts skip}{}
251 \renewcommand*{\midbookskip}{}
252 \renewcommand*{\midparts skip}{}
253 \renewcommand*{\printbookname}{}
```

```
254 \renewcommand*{\booknamefont}{}  
255 \renewcommand*{\booknamenum}{}  
256 \renewcommand*{\printbooknum}{}  
257 \renewcommand*{\booknumfont}{}  
258 \renewcommand*{\printpartname}{}  
259 \renewcommand*{\partnamefont}{}  
260 \renewcommand*{\partnamenum}{}  
261 \renewcommand*{\printpartnum}{}  
262 \renewcommand*{\partnumfont}{}  
263 \renewcommand*{\printbooktitle}[1]{}  
264 \renewcommand*{\booktitlefont}{}  
265 \renewcommand*{\printparttitle}[1]{}  
266 \renewcommand*{\parttitlefont}{}  
267 \renewcommand*{\bookpageend}{}  
268 \renewcommand*{\bookblankpage}{}  
269 \renewcommand*{\nobookblankpage}{}  
270 \renewcommand*{\partpageend}{}  
271 \renewcommand*{\partblankpage}{}  
272 \renewcommand*{\nopartblankpage}{}  
273 \RenewDocumentCommand{\newleadpage}{s o m m}{}% todo  
274 \RenewDocumentCommand{\renewleadpage}{s o m m}{}% todo  
275 \renewcommand*{\leadpagetoclevel}{chapter}  
276  
277 \renewcommand*{\openright}{}  
278 \renewcommand*{\openleft}{}  
279 \renewcommand*{\openany}{}  
280 \renewcommand*{\clearforchapter}{}  
281 \renewcommand*{\memendofchapterhook}{}  
282 \renewcommand*{\chapterheadstart}{}  
283 % \newlength{\beforechapskip}  
284 \renewcommand*{\afterchapnum}{  
285 % \newlength{\midchapskip}  
286 \renewcommand*{\afterchaptertitle}{}  
287 % \newlength{\afterchapskip}  
288 \renewcommand*{\printchaptername}{}  
289 \renewcommand*{\chapnamefont}{}  
290 \renewcommand*{\chapnamenum}{}  
291 \renewcommand*{\printchapnum}{  
292 \renewcommand*{\chapnumfont}{}  
293 \renewcommand*{\printchaptertitle}[1]{}  
294 \renewcommand*{\chaptilefont}{}  
295 \renewcommand*{\printchapnum}{  
296 \renewcommand*{\indentafterchapter}{}  
297 \renewcommand*{\noindentafterchapter}{}  
298 \renewcommand*{\insertchapterspace}{}  
299  
300 \renewcommand*{\chapterstyle}[1]{}  
301 \renewcommand{\makechapterstyle}[2]{}  
302 \renewcommand*{\chapindent}{}  
303 \let\chapterprecis\cftchapterprecis
```

```
304 \let\chapterprecishere\cftchapterprecishere
305 \let\chapterprecistoc\cftchapterprecistoc
306 \renewcommand*{\precisfont}{}
307 \renewcommand*{\prechapterprecis}{}
308 \renewcommand*{\postchapterprecis}{}
309 \renewcommand{\precistotext}[1]{}
310 \renewcommand*{\precistocfont}{}
311 \renewcommand*{\precistocformat}{}
312 % \newlength{\prechapterprecisshift}
313
314 \renewcommand*{\setbeforesecskip}[1]{}
315 \renewcommand*{\setaftersecskip}[1]{}
316 \renewcommand*{\setsecindent}[1]{}
317 \renewcommand*{\setseheadstyle}[1]{}
318 \renewcommand*{\setbeforesubsecskip}[1]{}
319 \renewcommand*{\setaftersubsecskip}[1]{}
320 \renewcommand*{\setsubsecindent}[1]{}
321 \renewcommand*{\setsubseheadstyle}[1]{}
322 \renewcommand*{\setbeforesubsubsecskip}[1]{}
323 \renewcommand*{\setaftersubsubsecskip}[1]{}
324 \renewcommand*{\setsubsubsecindent}[1]{}
325 \renewcommand*{\setsubsubseheadstyle}[1]{}
326 \renewcommand*{\setbeforeparaskip}[1]{}
327 \renewcommand*{\setafterparaskip}[1]{}
328 \renewcommand*{\setparaindent}[1]{}
329 \renewcommand*{\setparaheadstyle}[1]{}
330 \renewcommand*{\setbeforesubparaskip}[1]{}
331 \renewcommand*{\setaftersubparaskip}[1]{}
332 \renewcommand*{\setsubparaindent}[1]{}
333 \renewcommand*{\setsubparaheadstyle}[1]{}
334 \renewcommand{\@hangfrom}[1]{#1}
335 \renewcommand{\sethangfrom}[1]{}
336 \renewcommand{\setsecnumformat}[1]{}
337
338 \renewcommand*{\hangsecnum}{}
339 \renewcommand*{\defaultsecnum}{}
340
341 \renewcommand*{\sechook}{}
342 \renewcommand{\setsechook}[1]{}
343 \renewcommand*{\subsechook}{}
344 \renewcommand{\setsubsechook}[1]{}
345 \renewcommand*{\subsubsechook}{}
346 \renewcommand{\setsubsubsechook}[1]{}
347 \renewcommand*{\parahook}{}
348 \renewcommand{\setparahook}[1]{}
349 \renewcommand*{\subparahook}{}
350 \renewcommand{\setsubparahook}[1]{}
351
352 \RenewDocumentCommand{\plainbreak}{s m}{\begin{center}~\end{center}}
353
```

```
354 \RenewDocumentCommand{\fancybreak}{s +m}{%
355 \begin{center}#2\end{center}%
356 }
357
358 \RenewDocumentCommand{\plainfancybreak}{s m m +m}{%
359 \begin{center}#4\end{center}%
360 }
361
362 \RenewDocumentCommand{\pfbreak}{s}{%
363 \begin{center}
364 \pfbreakdisplay
365 \end{center}
366 }
367
368 % \newlength{\pfbreakskip}
369 \renewcommand{\pfbreakdisplay}{*\quad*\quad*}
370
371 \renewcommand{\makeheadstyles}[2]{}
372 \renewcommand*{\headstyles}[1]{}
```

§ 469.8 Pagination and headers

```
373 \renewcommand*{\savepagenumber}{}
374 \renewcommand*{\restorepagenumber}{}
375 \renewcommand*{\uppercaseheads}{}
376 \renewcommand*{\nouppercaseheads}{}
377
378 \renewcommand*{\bookpagemark}[1]{}
379 \renewcommand*{\partmark}[1]{}
380 \renewcommand*{\bibmark}{}
381 \renewcommand*{\indexmark}{}
382 \renewcommand*{\glossarymark}{}
383
384 \LWR@origpagestyle{empty}
385 \renewcommand*{\ps@empty}{}
386 \renewcommand*{\makepagestyle}[1]{}
387 \renewcommand*{\emptypshook}{}
388 % \renewcommand*{\empty@oddhead}{}
389 % \renewcommand*{\empty@oddfoot}{}
390 % \renewcommand*{\empty@evenhead}{}
391 % \renewcommand*{\empty@evenfoot}{}
392 \renewcommand*{@oddhead}{}
393 \renewcommand*{@oddfoot}{}
394 \renewcommand*{@evenhead}{}
395 \renewcommand*{@evenfoot}{}
396 \renewcommand*{\aliaspagestyle}[2]{}
397 \renewcommand*{\copypagestyle}[2]{}
398
399 \renewcommand*{\makeevenhead}{}
```

```

400 \renewcommand*{\makeoddhead}[4]{}
401 \renewcommand*{\makeevenfoot}[4]{}
402 \renewcommand*{\makeoddfoot}[4]{}
403 \renewcommand*{\makerunningwidth}[3]{}
404 % \newlength{\headwidth}
405 \renewcommand*{\makeheadrule}[3]{}
406 \renewcommand*{\makefootrule}[3]{}
407 \renewcommand*{\makeheadfootruleprefix}[3]{}
408 % \newlength{\normalrulethickness}
409 % \setlength{\normalrulethickness}{.4pt}
410 % \newlength{\footruleheight}
411 % \newlength{\footruleskip}
412 \renewcommand*{\makeheadposition}[5]{}
413 \renewcommand{\makepsmarks}[2]{}
414 \renewcommand*{\makeheadfootstrut}[3]{}
415
416 \renewcommand{\createplainmark}[3]{}
417 \renewcommand{\memUHead}[1]{}
418 \renewcommand{\createmark}[5]{}
419 \renewcommand*{\clearplainmark}[1]{}
420 \renewcommand*{\clearmark}[1]{}
421 \renewcommand{\addtopsmarks}[3]{}
422 \renewcommand{\ifonlyfloats}[2]{#2}
423 \renewcommand*{\mergepagefloatstyle}[3]{}
424
425 \renewcommand*{\framepichead}={}
426 \renewcommand*{\framepictextfoot}={}
427 \renewcommand*{\framepichook}={}
428 \renewcommand*{\showheadfootlocoff}={}
429 \renewcommand*{\showtextblocklocoff}={}

```

§ 469.9 Paragraphs and lists

```

430 \renewcommand{\hangfrom}[1]{#1}
431 \let\centerfloat\centering
432 \renewcommand*{\raggedyright}[1][]{}
433 % \newlength{\ragrparindent}
434 \renewcommand{\sourceatright}[2][]{\attribution{#2}}
435 \let\memorigdbs\LWR@endofline
436 \let\memorigpar\par
437 \let\atcentercr\LWR@endofline
438
439 \renewcommand*{\linenottooshort}[1][]{}
440 \renewcommand*{\russianpar}(){}
441 \renewcommand*{\lastlinerulefill}(){}
442 \renewcommand*{\lastlineparrule}){}
443 \renewcommand*{\justlastraggedleft}){}
444 \renewcommand*{\raggedrightthenleft}){}
445 \renewcommand*{\leftcenterright}){}

```

```
446
447 \renewcommand{\leftspringright}[4]{%
448 \begin{minipage}{#1\linewidth}\#3\end{minipage}\qquad%
449 \begin{minipage}{#2\linewidth}\begin{flushright}\#4\end{flushright}\end{minipage}%
450 }
451
452 \renewenvironment*{blockdescription}
453 {\LWR@descriptionstart\LWR@origdescription}
454 {\enddescription}
455 \renewcommand*{\blockdescriptionlabel}[1]{\textbf{#1}}
456 \renewenvironment*{labelled}[1]{\begin{description}}{\end{description}}
457 \renewenvironment*{flexlabelled}[6]{\begin{description}}{\end{description}}
458 \renewcommand*{\tightlists}{}
459 \renewcommand*{\defaultlists}{}
460 \RenewDocumentCommand{\firmlists}{s}{}
461 \renewcommand*{\firmlist}{}
462 \renewcommand*{\tightlist}{}
463 \renewcommand*{\zerotrivseps}{}
464 \renewcommand*{\savetrvseps}{}
465 \renewcommand*{\restoretrvseps}{}
```

§ 469.10 Contents lists

```
466 \csletcs{tableofcontents*}{tableofcontents}
467 \csletcs{listoffigures*}{listoffigures}
468 \csletcs{listoftables*}{listoftables}
469 \renewenvironment{KeepFromToc}{}{}
470 \renewcommand*{\onecoltocetc}{}
471 \renewcommand*{\twocoltocetc}{}
472 \renewcommand*{\ensureonecol}{}
473 \renewcommand*{\restorefromonecol}{}
474 \renewcommand*{\doccoltocetc}{}
475 \renewcommand*{\maxtocdepth}[1]{\% tocvsec2}
476 \renewcommand*{\settocdepth}[1]{\% tocvsec2}
477
478 \renewcommand{\tocheadstart}{}
479 \renewcommand{\printtoctitle}[1]{}
480 \renewcommand{\tocmark}{}
481 \renewcommand{\aftertoctitle}{}
482 \renewcommand{\lofheadstart}{}
483 \renewcommand{\printloftitle}[1]{}
484 \renewcommand{\lofmark}{}
485 \renewcommand{\afterloftitle}{}
486 \renewcommand{\lotheadstart}{}
487 \renewcommand{\printlottitle}[1]{}
488 \renewcommand{\lotmark}{}
489 \renewcommand{\afterlottitle}{}
490
491 \renewcommand*{\setpnumwidth}[1]{}
```

```
492 \renewcommand*{\setrmarg}[1]{}
493 \renewcommand*{\cftbookbreak}{}%
494 \renewcommand*{\cftpartbreak}{}%
495 \renewcommand*{\cftchapterbreak}{}%

496 % \newlength{\cftbeforebookskip}
497 % \newlength{\cftbookindent}
498 % \newlength{\cftbooknumwidth}
499 \renewcommand*{\cftbookfont}{}%
500 \renewcommand*{\cftbookname}{}%
501 \renewcommand*{\cftbookpresnum}{}%
502 \renewcommand*{\cftbookaftersnum}{}%
503 \renewcommand*{\cftbookaftersnumb}{}%
504 \renewcommand*{\cftbookleader}{}%
505 \renewcommand*{\cftbookdotsep}{1}
506 \renewcommand*{\cftbookpagefont}{}%
507 \renewcommand*{\cftbookafterpnum}{}%
508 \renewcommand*{\cftbookformatpnum}[1]{}%
509 \renewcommand*{\cftbookformatpnumhook}[1]{}%
```

Part is already defined by tocloft.

```
510 % \newlength{\cftbeforechapterskip}
511 % \newlength{\cftchapterindent}
512 % \newlength{\cftchapternumwidth}
513 \renewcommand*{\cftchapterfont}{}%
514 \renewcommand*{\cftchaptername}{}%
515 \renewcommand*{\cftchapterpresnum}{}%
516 \renewcommand*{\cftchapteraftersnum}{}%
517 \renewcommand*{\cftchapteraftersnumb}{}%
518 \renewcommand*{\cftchapterleader}{}%
519 \renewcommand*{\cftchapterdotsep}{1}
520 \renewcommand*{\cftchapterpagefont}{}%
521 \renewcommand*{\cftchapterafterpnum}{}%
522 \renewcommand*{\cftchapterformatpnum}[1]{}%
523 \renewcommand*{\cftchapterformatpnumhook}[1]{}%

524 % \newlength{\cftbeforesections skip}
525 % \newlength{\cftsectionindent}
526 % \newlength{\cftsectionnumwidth}
527 \renewcommand*{\cftsectionfont}{}%
528 \renewcommand*{\cftsectionname}{}%
529 \renewcommand*{\cftsectionpresnum}{}%
530 \renewcommand*{\cftsectionaftersnum}{}%
531 \renewcommand*{\cftsectionaftersnumb}{}%
532 \renewcommand*{\cftsectionleader}{}%
533 \renewcommand*{\cftsectiondotsep}{1}
534 \renewcommand*{\cftsectionpagefont}{}%
535 \renewcommand*{\cftsectionafterpnum}{}%
536 \renewcommand*{\cftsectionformatpnum}[1]{}%
```

```
537 \renewcommand*{\cftsectionformatpnumhook}{[1]{}}
```

```
538 % \newlength{\cftbeforesubsectionskip}
```

```
539 % \newlength{\cftsubsectionindent}
```

```
540 % \newlength{\cftsubsectionnumwidth}
```

```
541 \renewcommand*{\cftsubsectionfont}{}{}
```

```
542 \renewcommand*{\cftsubsectionname}{}{}
```

```
543 \renewcommand*{\cftsubsectionpresnum}{}{}
```

```
544 \renewcommand*{\cftsubsectionaftersnum}{}{}
```

```
545 \renewcommand*{\cftsubsectionaftersnumb}{}{}
```

```
546 \renewcommand*{\cftsubsectionleader}{}{}
```

```
547 \renewcommand*{\cftsubsectiondotsep}{[1]}
```

```
548 \renewcommand*{\cftsubsectionpagefont}{}{}
```

```
549 \renewcommand*{\cftsubsectionafterpnum}{}{}
```

```
550 \renewcommand*{\cftsubsectionformatpnum}{[1]}{}
```

```
551 \renewcommand*{\cftsubsectionformatpnumhook}{[1]}{}
```



```
552 % \newlength{\cftbeforesubsubsectionskip}
```

```
553 % \newlength{\cftsubsubsectionindent}
```

```
554 % \newlength{\cftsubsubsectionnumwidth}
```

```
555 \renewcommand*{\cftsubsubsectionfont}{}{}
```

```
556 \renewcommand*{\cftsubsubsectionname}{}{}
```

```
557 \renewcommand*{\cftsubsubsubsectionpresnum}{}{}
```

```
558 \renewcommand*{\cftsubsubsubsectionaftersnum}{}{}
```

```
559 \renewcommand*{\cftsubsubsubsectionaftersnumb}{}{}
```

```
560 \renewcommand*{\cftsubsubsubsectionleader}{}{}
```

```
561 \renewcommand*{\cftsubsubsubsectiondotsep}{[1]}
```

```
562 \renewcommand*{\cftsubsubsubsectionpagefont}{}{}
```

```
563 \renewcommand*{\cftsubsubsubsectionafterpnum}{}{}
```

```
564 \renewcommand*{\cftsubsubsubsectionformatpnum}{[1]}{}
```

```
565 \renewcommand*{\cftsubsubsubsectionformatpnumhook}{[1]}{}
```



```
566 % \newlength{\cftbeforeparagraphskip}
```

```
567 % \newlength{\cftpaginindent}
```

```
568 % \newlength{\cftpaginwidth}
```

```
569 \renewcommand*{\cftpaginfont}{}{}
```

```
570 \renewcommand*{\cftpaginname}{}{}
```

```
571 \renewcommand*{\cftpaginpresnum}{}{}
```

```
572 \renewcommand*{\cftpaginhaftersnum}{}{}
```

```
573 \renewcommand*{\cftpaginhaftersnumb}{}{}
```

```
574 \renewcommand*{\cftpaginleader}{}{}
```

```
575 \renewcommand*{\cftpagindotsep}{[1]}
```

```
576 \renewcommand*{\cftpaginpagefont}{}{}
```

```
577 \renewcommand*{\cftpaginafterpnum}{}{}
```

```
578 \renewcommand*{\cftpaginformatpnum}{[1]}{}
```

```
579 \renewcommand*{\cftpaginformatpnumhook}{[1]}{}
```



```
580 % \newlength{\cftbeforesubparagraphskip}
```

```
581 % \newlength{\cftsubparagraphindent}
```

```
582 % \newlength{\cftsubparagraphnumwidth}
```

```
583 \renewcommand*{\cftsubparagraphfont}{}  
584 \renewcommand*{\cftsubparagraphname}{}  
585 \renewcommand*{\cftsubparagraphpresnum}{}  
586 \renewcommand*{\cftsubparagraphaftersnum}{}  
587 \renewcommand*{\cftsubparagraphaftersnumb}{}  
588 \renewcommand*{\cftsubparagraphleader}{}  
589 \renewcommand*{\cftsubparagraphdotsep}{1}  
590 \renewcommand*{\cftsubparagraphpagefont}{}  
591 \renewcommand*{\cftsubparagraphafterpnum}{}  
592 \renewcommand*{\cftsubparagraphformatpnum}[1]{}  
593 \renewcommand*{\cftsubparagraphformatpnumhook}[1]{}  
  
594 % \newlength{\cftbeforefigureskip}  
595 % \newlength{\cftfigureindent}  
596 % \newlength{\cftfigurenumwidth}  
597 \renewcommand*{\cftfigurefont}{}  
598 \renewcommand*{\cftfigurename}{}  
599 \renewcommand*{\cftfigurepresnum}{}  
600 \renewcommand*{\cftfigureaftersnum}{}  
601 \renewcommand*{\cftfigureaftersnumb}{}  
602 \renewcommand*{\cftfigureleader}{}  
603 \renewcommand*{\cftfiguredotsep}{1}  
604 \renewcommand*{\cftfigurepagefont}{}  
605 \renewcommand*{\cftfigureafterpnum}{}  
606 \renewcommand*{\cftfigureformatpnum}[1]{}  
607 \renewcommand*{\cftfigureformatpnumhook}[1]{}  
  
608 % \newlength{\cftbeforesubfigureskip}  
609 % \newlength{\cftsubfigureindent}  
610 % \newlength{\cftsubfigurenumwidth}  
611 \newcommand*{\cftsubfigurefont}{}  
612 \newcommand*{\cftsubfigurename}{}  
613 \newcommand*{\cftsubfigurepresnum}{}  
614 \newcommand*{\cftsubfigureaftersnum}{}  
615 \newcommand*{\cftsubfigureaftersnumb}{}  
616 \newcommand*{\cftsubfigureleader}{}  
617 \newcommand*{\cftsubfiguredotsep}{1}  
618 \newcommand*{\cftsubfigurepagefont}{}  
619 \newcommand*{\cftsubfigureafterpnum}{}  
620 \newcommand*{\cftsubfigureformatpnum}[1]{}  
621 \newcommand*{\cftsubfigureformatpnumhook}[1]{}  
  
622 % \newlength{\cftbeforetablesip}  
623 % \newlength{\cfttableindent}  
624 % \newlength{\cfttablenumwidth}  
625 \renewcommand*{\cfttablefont}{}  
626 \renewcommand*{\cfttablename}{}  
627 \renewcommand*{\cfttablepresnum}{}  
628 \renewcommand*{\cfttableaftersnum}{}  
629 \renewcommand*{\cfttableaftersnumb}{}  

```

```

630 \renewcommand*{\cfttableleader}{}
631 \renewcommand*{\cfttabledotsep}{1}
632 \renewcommand*{\cfttablepagefont}{}
633 \renewcommand*{\cfttableafterpnum}{}
634 \renewcommand*{\cfttableformatpnum}[1]{}
635 \renewcommand*{\cfttableformatpnumhook}[1]{}

636 % \newlength{\cftbeforesubtableskip}
637 % \newlength{\cftsubtableindent}
638 % \newlength{\cftsubtablenumwidth}
639 \newcommand*{\cftsubtablefont}{}
640 \newcommand*{\cftsubtablename}{}
641 \newcommand*{\cftsubtablepresnum}{}
642 \newcommand*{\cftsubtableaftersnum}{}
643 \newcommand*{\cftsubtableaftersnumb}{}
644 \newcommand*{\cftsubtableleader}{}
645 \newcommand*{\cftsubtabledotsep}{1}
646 \newcommand*{\cftsubtablepagefont}{}
647 \newcommand*{\cftsubtableafterpnum}{}
648 \newcommand*{\cftsubtableformatpnum}[1]{}
649 \newcommand*{\cftsubtableformatpnumhook}[1]{}

650 \renewcommand*{\booknumberline}[1]{}
651 \renewcommand*{\partnumberline}[1]{}
652 \renewcommand*{\chapternumberline}[1]{}
653 \renewcommand*{\numberlinehook}[1]{}
654 % \renewcommand*{\cftwhatismyname}{}%
655 \renewcommand*{\booknumberlinehook}[1]{}
656 \renewcommand*{\partnumberlinehook}[1]{}
657 \renewcommand*{\chapternumberlinehook}[1]{}
658 \renewcommand{\numberlinebox}[2]{}
659 \renewcommand{\booknumberlinebox}[2]{}
660 \renewcommand{\partnumberlinebox}[2]{}
661 \renewcommand{\chapternumberlinebox}[2]{}
662 %
663 % \newlength{\cftparskip}
664 \renewcommand*{\cftpagenumbersoff}[1]{}
665 \renewcommand*{\cftpagenumberson}[1]{}
666 \renewcommand*{\cftlocalchange}[3]{}
667 \renewcommand*{\cftaddtitleline}[4]{}
668 \renewcommand*{\cftaddnumtitleline}[4]{}
669 \renewcommand{\cftinsertcode}[2]{}
670 \renewcommand{\cftinserthook}[2]{}
671 \renewcommand{\settocpreprocessor}[2]{}
672 \DeclareRobustCommand{\cftpagenumbersoff}[1]{}
673 \DeclareRobustCommand{\cftpagenumberson}[1]{}

```

§ 469.11 Floats and captions

\newfloat [⟨1: within⟩] {⟨2: type⟩} {⟨3: ext⟩} {⟨4: capname⟩}

```

674 \RenewDocumentCommand{\newfloat}{o m m m}{%
675 \IfValueTF{#1}{%
676 {\DeclareFloatingEnvironment[fileext=#3,within=#1,name={#4}]{#2}}{%
677 {\DeclareFloatingEnvironment[fileext=#3,name={#4}]{#2}}{%

```

newfloat package automatically creates the `\listof` command for new floats, but float does not, so remove `\listof` here in case it is manually created later.

```

678 \cslet{listof#2s}\relax%
679 \cslet{listof#2es}\relax%
680 }

```

`\newlistof` [*<within>*] {*<type>*} {*<ext>*} {*<listofname>*}

Emulated through the `\newfloat` mechanism. Note that memoir uses a different syntax than tocloft for the name.

```

681 \RenewDocumentCommand{\newlistof}{o m m m}{%
682 {%
683 \IfValueTF{#1}{%
684 {\newlistentry[#1]{#2}{#3}{0}}{%
685 {\newlistentry[#2]{#3}{0}}{%
686 @namedef{ext@#2}{#3}}{%
687 @ifundefined{c@#3depth}{\newcounter{#3depth}}{}{%
688 \setcounter{#3depth}{1}}{%
689 @namedef{#3mark}{}}{%
690 @namedef{#2}{\listof{#2}{#4}}{%
691 @namedef{cftmake#3title}{}}{%
692 @ifundefined{cftbefore#3titleskip}{%
693     \expandafter\newlength\csname cftbefore#3titleskip\endcsname
694     \expandafter\newlength\csname cftafter#3titleskip\endcsname
695 }{}}{%
696 @namedef{cft#3titlefont}{}}{%
697 @namedef{cftafter#3title}{}}{%
698 @namedef{cft#3prehook}{}}{%
699 @namedef{cft#3posthook}{}}{%
700 }

```

```
701 \renewcommand{\setfloatadjustment}[2]{}{}
```

Borrowed from the lwarp version of keyfloat:

```

702 \NewDocumentEnvironment{KFLTmemoir@marginfloat}{O{-1.2ex} m}{%
703 {%
704 \LWR@BlockClassWP{float:right; width:2in; margin:10pt}{\marginblock}{%
705 \captionsetup{type=#2}}{%
706 }{%
707 {%
708 \endLWR@BlockClassWP{%
709 }

```

```
710
711 \DeclareDocumentEnvironment{marginfigure}{o}
712   {\begin{KFLTmemoir@marginfloat}{figure}}
713   {\end{KFLTmemoir@marginfloat}}
714
715 \DeclareDocumentEnvironment{margintable}{o}
716   {\begin{KFLTmemoir@marginfloat}{table}}
717   {\end{KFLTmemoir@marginfloat}}
718
719 \renewcommand{\setmarginfloatcaptionadjustment}[2]{}
720 \renewcommand{\setmpjustification}[2]{}
721 \renewcommand*{\mpjustification}{}%
722 \renewcommand*{\setfloatlocations}[2]{}%
723 \DeclareDocumentCommand{\suppressfloats}{o}{}%
724 \renewcommand*{\FloatBlock}{}%
725 \renewcommand*{\FloatBlockAllowAbove}{}%
726 \renewcommand*{\FloatBlockAllowBelow}{}%
727 \renewcommand*{\setFloatBlockFor}{}%
728
729 \renewcommand{\captiontitlefinal}[1]{}%
730
731 \renewcommand{\fleqtable}{\tablename}%
732 \renewcommand{\flegfigure}{\figurename}%
733 \renewcommand{\flegtoctable}{}%
734 \renewcommand{\flegtocfigure}{}%
735
736 \renewcommand{\subcaption}[2][]{%
737 \ifblank{#1}{\subfloat[#2]}{\subfloat[#1][#2]}%}
738 }
739
740 \renewcommand{\contsubcaption}{\ContinuedFloat\subcaption}%
741
742 \LetLtxMacro{\subcaptionref}{\subref}%
743
744 \renewcommand*{\tightsubcaptions}{}%
745 \renewcommand*{\loosesubcaptions}{}%
746
747 \renewcommand*{\subcaptionsize}[1]{}%
748 \renewcommand*{\subcaptionlabelfont}[1]{}%
749 \renewcommand*{\subcaptionfont}[1]{}%
750 \renewcommand*{\subcaptionstyle}[1]{}%
751
752 \renewcommand*{\hangsubcaption}{}%
753 \renewcommand*{\shortsubcaption}{}%
754 \renewcommand*{\normalsubcaption}{}%
755
756 \RenewDocumentEnvironment{sidecaption}{o m o}
757 }
```

```
758 {
759 \IfValueTF{#1}{\caption[#1]{#2}}{\caption{#2}}%
760 \IfValueT{#3}{\label{#3}}%
761 }
762
763 % \newlength{\sidecapwidth}
764 % \newlength{\sidecapsep}
765 \renewcommand*{\setsidecaps}[2]{}
766 \renewcommand*{\sidecapmargin}[1]{}
767 % \newif\ifscapmargleft
768 \scapmargleftfalse
769 \renewcommand*{\setsidecappos}[1]{}
770
771 \RenewDocumentEnvironment{sidecontcaption}{m o}
772 {}
773 {%
774 \ContinuedFloat%
775 \caption{#1}}%
```

Without `\@capttype`, the section is referred to instead.

```
776 \IfValueT{#2}{\label[\@capttype]{#2}}%
777 }
```

`\sidenamedlegend` does not appear to use the TOC argument.

```
778 \renewenvironment{sidenamedlegend}[2][]{%
779 \begin{center}%
780 \@nameuse{@capttype name}\CaptionSeparator#2%
781 \end{center}%
782 }%
783 {}%
784
785 \renewenvironment{sidelegend}[1]%
786 {\begin{center}%
787 #1%
788 }%
789 }%
790 {\end{center}%
791
792 \renewcommand*{\sidecapstyle}{}%
793 \renewcommand*{\overridescapmargin}[1]{}%
794 % \newlength{\sidecapraise}%
795 \renewcommand*{\sidecapfloatwidth}{\linewidth}%
796
797 \LetLtxMacro{\ctabular}{\tabular}%
798 \LetLtxMacro{\endctabular}{\endtabular}%
799
800 \renewcommand{\autorows}[5][]{%
```

```
801 #5
802 }
803
804 \renewcommand{\autocols}[5] []{%
805 #5
806 }
```

§ 469.12 Page notes

```
807 \renewcommand*{\feetabovefloat}{}
808 \renewcommand*{\feetbelowfloat}{}
809 \renewcommand*{\feetatbottom}{}
810
811 \renewcommand*{\verbfootnote}[2][]{%
812 \PackageError{l warp,memoir}%
813 {Verbatim footnotes are not yet supported by l warp.}%
814 {This may be improved some day.}%
815 }
816
817 \renewcommand*{\plainfootnotes}{}
818 \renewcommand*{\twocolumnfootnotes}{}
819 \renewcommand*{\threecolumnfootnotes}{}
820 \renewcommand*{\paragraphfootnotes}{}
821 \renewcommand*{\footfudgefiddle}{}
822
823 \renewcommand*{\newfootnoteseries}[1]{%
824 \PackageError{l warp,memoir}%
825 {Memoir footnote series are not yet supported by l warp.}%
826 {This may be improved some day.}%
827 }
828
829 \renewcommand*{\plainfootstyle}[1]{}
830 \renewcommand*{\twocolumnfootstyle}[1]{}
831 \renewcommand*{\threecolumnfootstyle}[1]{}
832 \renewcommand*{\paragraphfootstyle}[1]{}
833
834 \renewcommand*{\footfootmark}{}
835 \renewcommand*{\footmarkstyle}[1]{}
836
837 % \newlength{\footmarkwidth}
838 % \newlength{\footmarksep}
839 % \newlength{\footparindent}
840
841 \renewcommand*{\foottextfont}{}
842
843 \renewcommand*{\marginparmargin}[1]{}
844 \renewcommand*{\sideparmargin}[1]{}
845
846 \LetLtxMacro{\sidepar}{\marginpar}
```

```

847 \renewcommand*{\sideparfont}{}
848 \renewcommand*{\sideparform}{}
849 \LWR@providelength{\sideparvshift}
850
851 \renewcommand*{\parnopar}{}
852
853 \renewcommand{\sidebar}[1]{\begin{quote}#1\end{quote}}
854 \renewcommand*{\sidebarmargin}[1]{}
855 \renewcommand*{\sidebarfont}{}
856 \renewcommand*{\sidebarform}{}
857 % \newlength{\sidebarhsep}
858 % \newlength{\sidebarvsep}
859 % \newlength{\sidebarwidth}
860 % \newlength{\sidebartopsep}
861 \renewcommand{\setsidebarheight}[1]{}
862 \renewcommand*{\setsidetabs}[6]{}
863 \renewcommand*{\footnotesatfoot}{}
864 \renewcommand*{\footnotesinmargin}{}
865
866 \LetLtxMacro{\sidefootnote}{\footnote}
867 \LetLtxMacro{\sidefootnotemark}{\footnotemark}
868 \LetLtxMacro{\sidefootnotetext}{\footnotetext}
869
870 \renewcommand*{\sidefootmargin}[1]{}
871 % \newlength{\sidefoothsep}
872 % \newlength{\sidefootvsep}
873 % \newlength{\sidefootwidth}
874 % \newlength{\sidefootadjust}
875 % \newlength{\sidefootheight}
876 \renewcommand*{\setsidefootheight}[1]{}
877 % \renewcommand*{\sidefootfont}{}% in docs but not in the package
878 \renewcommand*{\setsidefeet}[6]{}
879 \renewcommand*{\sidefootmarkstyle}[1]{}
880 \renewcommand*{\sidefoottextfont}{}
881 \renewcommand*{\sidefootform}{}
882
883 \renewcommand*{\continuousnotenums}{\pncontopttrue}% from pagenote
884 \renewcommand*{\notepageref}{}
885 \renewcommand*{\prenotetext}{}
886 \renewcommand*{\postnotetext}{}
887 \renewcommand*{\idtextinnotes}[1]{}
888 \renewcommand*{\printpageinnotes}[1]{}
889 \renewcommand*{\printpageinnoteshyperref}[1]{}
890 \renewcommand*{\foottopagenote}{}
891 \renewcommand*{\pagetofootnote}{}

```

§ 469.13 Decorative text

```
892 \renewcommand*{\epigraphposition}[1]{}
```

```

893 \renewcommand*\{epigraphtextposition}[1]{}
894 \renewcommand*\{epigraphsourceposition}[1]{}
895 \renewcommand*\{epigraphfontsize}[1]{}
896 \renewcommand*\{epigraphforheader}[2][]{}
897 \renewcommand*\{epigraphpicture}{}

```

§ 469.14 Poetry

```

898 \renewcommand*\{vphantom}{}
899 \renewcommand*\{vleftofline}[1]{#1}
900 % \let\linenumberfrequency\poemlines
901 % \renewcommand*\{linenumberfont}[1]{}
902
903 \DeclareDocumentCommand{\PoemTitle}{s o o m}{%
904 \IfValueTF{#2}{%
905 {\poemtitle[#2]{#4}}{%
906 {\poemtitle{#4}}{%
907 }{%
908
909 \renewcommand*\{NumberPoemTitle}{}
910 \renewcommand*\{PlainPoemTitle}{}
911 \renewcommand*\{poemtitlestyle}{}
912 \renewcommand*\{poemtitlestarmark}[1]{}
913 \renewcommand*\{poemtitlestarpstyle}{}
914 \renewcommand*\{PoemTitleheadstart}{}
915 \renewcommand*\{printPoemTitlenonum}{}
916 \renewcommand*\{printPoemTitlenum}{}
917 \renewcommand*\{afterPoemTitlenum}{}
918 \renewcommand*\{printPoemTitletitle}[1]{}
919 \renewcommand*\{afterPoemTitle}{}
920 \newlength{\midpoemtitleskip}
921 \renewcommand*\{PoemTitlenumfont}{}
922 \renewcommand*\{PoemTitlefont}{}

```

§ 469.15 Boxes, verbatims and files

```

923 \renewenvironment{qframe}{\framed}{\endframed}
924 \renewenvironment{qshade}{\shaded}{\endshaded}

```

Use the comment package:

```

925 \renewcommand*\{commentsoff}[1]{\includecomment{#1}}
926 \renewcommand*\{commentson}[1]{\excludecomment{#1}}
927 \LetLtxMacro\renewcomment\commentson
928
929 \renewcommand*\{setverbatimfont}[1]{}
930 \renewcommand*\{tabson}[1]{}
931 \renewcommand*\{tabsoff}{}
932 \renewcommand*\{wrappingon}{}

```

```

933 \renewcommand*{\wrappingoff}{}
934 \renewcommand*{\verbatimindent}{}
935 \renewcommand*{\verbatimbreakchar}[1]{}
936 \DefineVerbatimEnvironment{fboxverbatim}{Verbatim}{frame=single}

```

`boxedverbatim` is already defined by `moreverb`. `boxedverbatim*` does not appear to work at all, even in a minimal print memoir document.

```

937 \renewcommand*{\b vbox}{}
938 \renewcommand*{\b topandtail}{}
939 \renewcommand*{\b sides}{}
940 \renewcommand*{\nob vbox}{}
941 % \newlength\b vboxsep
942 \renewcommand*{\b toprulehook}{}
943 \renewcommand*{\b topmidhook}{}
944 \renewcommand*{\b endrulehook}{}
945 \renewcommand*{\b leftsidehook}{}
946 \renewcommand*{\b rightsidehook}{}
947 \renewcommand*{\b perpagetrue}{}
948 \renewcommand*{\b perpagefalse}{}
949 \renewcommand{\b topofpage}[1]{}
950 \renewcommand{\b endofpage}[1]{}
951 \renewcommand*{\linenumberfrequency}[1]{}
952 \renewcommand*{\resetbvlinenumber}{}
953 \renewcommand*{\setbvlinenums}[2]{}
954 \renewcommand*{\linenumberfont}[1]{}
955 \renewcommand*{\b numbersinside}{}
956 \renewcommand*{\b numbersoutside}{}

```

§ 469.16 Cross referencing

```

957 \renewcommand*{\fref}[1]{\cref{#1}}
958 \renewcommand*{\tref}[1]{\cref{#1}}
959 \renewcommand*{\pref}[1]{\cpageref{#1}}
960 \renewcommand*{\Aref}[1]{\cref{#1}}
961 \renewcommand*{\Bref}[1]{\cref{#1}}
962 \renewcommand*{\Pref}[1]{\cref{#1}}
963 \renewcommand*{\Sref}[1]{\cref{#1}}
964 \renewcommand*{\figurerefname}{Figure}
965 \renewcommand*{\tablerefname}{Table}
966 \renewcommand*{\pagerefname}{page}
967 \renewcommand*{\bookrefname}{Book-}
968 \renewcommand*{\partrefname}{Part-}
969 \renewcommand*{\chapterrefname}{Chapter~}
970 \renewcommand*{\sectionrefname}{\S}
971 \renewcommand*{\appendixrefname}{Appendix~}
972 \LetLtxMacro{\titleref}{\nameref}
973 \renewcommand*{\headnameref}{}

```

```

974 \renewcommand*{\tocnameref}{}
975
976 \providecounter{LWR@currenttitle}
977
978 \renewcommand*{\currenttitle}{%
979     \addtocounter{LWR@currenttitle}{1}%
980     \label{currenttitle}\arabic{LWR@currenttitle}}%
981     \nameref{currenttitle}\arabic{LWR@currenttitle}}%
982 }
983
984 \renewcommand*{\theTitleReference}[2]{}
985 \renewcommand*{\namerefon}{}
986 \renewcommand*{\namerefoff}{}

```

§ 469.17 Back matter

Redefined to write the LWR@autoindex counter instead of page. Note that memoir has two versions, depending on the use of hyperref.

```

987 \AtBeginDocument{
988
989 \def\@wrindexhyp#1|||\{%
990     \addtocounter{LWR@autoindex}{1}%
991     \LWR@new@label{LWRindex-\arabic{LWR@autoindex}}%
992 %     \ifshowindexmark\@showidx{#1}\fi
993     \protected@write\@auxout{}{%
994         \string\@wrindexm@m{\@idxfile}{#1}{\arabic{LWR@autoindex}}%
995     \endgroup
996     \@esphack}%

```

\specialindex behaves like a regular \index, pointing to where \specialindex is used. If \specialindex is used inside a figure or table after the \caption, then the hyperlink will be given the name of that particular figure or table.

```

997 \def\@wrspindexhyp#1|||\{%
998     \addtocounter{LWR@autoindex}{1}%
999     \LWR@new@label{LWRindex-\arabic{LWR@autoindex}}%
1000 %     \ifshowindexmark\@showidx{#1}\fi
1001     \protected@write\@auxout{}{%
1002         \string\@wrindexm@m{\@idxfile}{#1}{\@nameuse{the\@sptheadidx}}%
1003         \string\@wrindexm@m{\@idxfile}{#1}{\arabic{LWR@autoindex}}%
1004     }%
1005     \endgroup
1006     \@esphack}%
1007
1008 }% \AtBeginDocument

```

Patched to use _html filename and \BaseJobname:

```

1010 \catcode`\_=12%
1011 \renewcommand*{\makeindex}[1][\BaseJobname]{%
1012   \if@filesw
1013     \def\gindex{\@bsphack%
1014       \@ifnextchar [{\@index}{\@index[\BaseJobname]}}
1015     \def\specialindex{\@bsphack\@spindex}%
1016     \makememindexhook
1017     \expandafter\newwrite\csname #1@idxfile\endcsname
1018     \expandafter\immediate\openout \csname #1@idxfile\endcsname #1_html.idx\relax
1019     \typeout{Writing index file #1_html.idx }%
1020   \fi}
1021 \catcode`\_=8%

```

Patched to use _html filename and \BaseJobname. This will later be patched by the lwarf core.

```

1022 \catcode`\_=12%
1023 \renewcommand{\printindex}[1][\BaseJobname]{\@input@{\#1_html.ind}}
1024 \catcode`\_=8%
1025 \DeclareDocumentCommand{\newblock}{}{%
1026 %
1027 \renewcommand*{\showindexmarks}{}
1028 \renewcommand*{\hideindexmarks}{}
1029 %
1030 \renewcommand*{\xindyindex}{}

```

§ 469.18 Miscellaneous

```

1031 \renewcommand*{\changemarks}{}
1032 \renewcommand*{\nochangemarks}{}
1033 \renewcommand*{\added}[1]{}
1034 \renewcommand*{\deleted}[1]{}
1035 \renewcommand*{\changed}[1]{}
1036 %
1037 \renewcommand*{\showtrimsoff}{}
1038 \renewcommand*{\showtrimson}{}
1039 \renewcommand*{\trimXmarks}{}
1040 \renewcommand*{\trimLmarks}{}
1041 \renewcommand*{\trimFrame}{}
1042 \renewcommand*{\trimNone}{}
1043 \renewcommand*{\trimmarkscolor}{}
1044 \renewcommand*{\trimmarks}{}
1045 \renewcommand*{\tmarktl}{}
1046 \renewcommand*{\tmarktr}{}
1047 \renewcommand*{\tmarkbr}{}
1048 \renewcommand*{\tmarkbl}{}

```

```

1049 \renewcommand*{\tmarktm}{}
1050 \renewcommand*{\tmarkmr}{}
1051 \renewcommand*{\tmarkbm}{}
1052 \renewcommand*{\tmarkml}{}
1053 \renewcommand*{\trimmark}{}
1054 \renewcommand*{\quarkmarks}{}
1055 \renewcommand*{\registrationColour}[1]{}
1056
1057 \renewcommand*{\leavespergathering}[1]{}
1058
1059 \renewcommand*{\noprelistbreak}{}
1060
1061 \renewcommand*{\cleartorecto}{}
1062 \renewcommand*{\cleartoverso}{}
1063
1064 \renewenvironment{vplace}[1]{}{}{}
```

§ 469.19 **ccaption emulation**

```

1065 \renewcommand*{\captiondelim}[1]{\renewcommand*{\CaptionSeparator}{\#1}}
1066 \renewcommand*{\captionnamefont}[1]{}
1067 \renewcommand*{\captiontitlefont}[1]{}
1068 \renewcommand*{\flushleftright}{}
1069 \renewcommand*{\centerlastline}{}
1070 \renewcommand*{\captionstyle}[2]{}{}
1071 \DeclareDocumentCommand{\captionwidth}{m} {}
1072 \renewcommand*{\changecaptionwidth} {}
1073 \renewcommand*{\normalcaptionwidth} {}
1074 \renewcommand*{\hangcaption} {}
1075 \renewcommand*{\indentcaption}[1] {}
1076 \renewcommand*{\normalcaption} {}
1077 \renewcommand{\precaption}[1] {}
1078 \renewcommand{\postcaption}[1] {}
1079 \renewcommand{\midbicaption}[1] {}
1080 \renewcommand{\contcaption}[1]{%
1081 % \ContinuedFloat%
1082 % \caption{\#1}%
1083 \begin{LWR@figcaption}%
1084 % later becomes \caption*
1085 \LWR@isolate{@nameuse{@capttype name}}%
1086 \thechapter.\the\value{@capttype}\CaptionSeparator\LWR@isolate{\#1}%
1087 \end{LWR@figcaption}
1088 \newlength{\abovelegendskip}
1089 \setlength{\abovelegendskip}{0.5\baselineskip}
1090 \newlength{\belowlegendskip}
1091 \setlength{\belowlegendskip}{\abovelegendskip}
```

The extra \\ here forces a
 in HTML when \legend is used in a \marginpar.

```
1092 \renewcommand{\legend}[1]{\begin{center}#1\\\end{center}}%
1093
1094 \renewcommand{\namedlegend}[2][]{%
1095 \begin{center}%
1096 \cnameuse{fleg}\@capttype}\CaptionSeparator#2\\%
1097 \end{center}%
1098 \cnameuse{flegtoc}\@capttype}{#1}%
1099 }
1100
1101 \renewcommand{\newfixedcaption}[3][\caption]{%
1102 \renewcommand{#2}{\def\@capttype{#3}#1}%
1103 \renewcommand{\renewfixedcaption}[3][\caption]{%
1104 \renewcommand{#2}{\def\@capttype{#3}#1}%
1105 \renewcommand{\providefixedcaption}[3][\caption]{%
1106 \providecommand{#2}{\def\@capttype{#3}#1}%
1107
1108 \renewcommand{\bitwonumcaption}[6][]{%
1109 \ifblank{#2}{\caption{#3}}{\caption[#2]{#3}}%
1110 \addtocounter{\@capttype}{-1}%
1111 \begingroup%
1112 \csdef{\@capttype name}{#4}%
1113 \ifblank{#5}{\caption{#6}}{\caption[#5]{#6}}%
1114 \endgroup%
1115 \ifblank{#1}{}{\label{#1}}%
1116 }
1117
1118 \LetLtxMacro\bionenumcaption\bitwonumcaption% todo
1119
1120 \renewcommand{\bicaption}[5][]{%
1121 \ifblank{#2}{\caption{#3}}{\caption[#2]{#3}}%
1122 \begin{LWR@figcaption}% later becomes \caption*
1123 \LWR@isolate{#4} \thechapter.\the\value{\@capttype}\CaptionSeparator\LWR@isolate{#5}%
1124 \end{LWR@figcaption}
1125 \ifblank{#1}{}{\label{#1}}%
1126 }
1127
1128 \renewcommand{\bicontcaption}[3]{%
1129 \contcaption{#1}%
1130 \begingroup%
1131 \csdef{\@capttype name}{#2}%
1132 \contcaption{#3}%
1133 \endgroup%
1134 }
1135 % only in ccaption, not in memoir:
1136 % \LetLtxMacro\longbitwonumcaption\bitwonumcaption%
1137 % \LetLtxMacro\longbionenumcaption\bitwonumcaption%
1138 % \LetLtxMacro\longbicaption\bicaption%
1139
1140 \RenewDocumentCommand{\subtop}{O{} O{} m}{%
```

```
1141 \subfloat[#1][#2]{#3}%
1142 }
1143
1144 \RenewDocumentCommand{\subbottom}{O{} O{} m}{%
1145 \subfloat[#1][#2]{#3}%
1146 }
1147
1148 \renewcommand{\contsubtop}{%
1149 \ContinuedFloat\addtocounter{@captive}{1}%
1150 \subtop}
1151
1152 \renewcommand{\contsubbottom}{%
1153 \ContinuedFloat\addtocounter{@captive}{1}%
1154 \subbottom}
1155
1156 \renewcommand{\subconcluded}{}
1157
1158 \let\subfigure\subbottom
1159 \let\subtable\subtop
1160
1161 \let\contsubtable\contsubtop
1162 \let\contsubfigure\contsubbottom

1163 \newcommand{\newfloatentry}[4][\empty]{TODO: newfloatentry}
1164 \newcommand{\newfloatlist}[5][\empty]{TODO: newfloatlist}
1165 \newcommand{\newfloatenv}[4][\empty]{TODO: newfloatenv}
1166 \DeclareRobustCommand{\newfloatpagesoff}[1]{}
1167 \DeclareRobustCommand{\newfloatpageson}[1]{}
1168 \newcommand{\setnewfloatindents}[3]{}
```

§ 469.20 Final patchwork

```
1169 \newlistof{tableofcontents}{toc}{\contentsname}
1170 \newlistof{listoffigures}{lof}{\listfigurename}
1171 \newlistof{listoftables}{lot}{\listtablename}
```

Change History

§470 Chg Hist

For the most recent changes, see page 1069.

v0.10	General: 2016/03/08 Initial version . . 1	Docs: Table: Cross-referencing data structures. 490	
v0.11	General: 2016/03/11 1	Docs: Table: Float data structures. 502	
	Added section: Operating-System portability. 222	Docs: Trademarks section. 202	
	Added section: Selecting the operating system. 128	Docs: Troubleshooting cross-references. 197	
	Test Suite: MS-WINDOWS in README.txt 1	Test Suite: Assigned cleveref name for Test Float. 1	
	Test Suite: limages and index in README.txt 1	Test Suite: Floatrow 1	
v0.12	\LWR@newhtmlfile: Bugfix: toc with numbered files. 373	v0.15	General: 2016/04/06 1
	General: 2016/03/14 1	Added. 749	
	Global: Uses \p@(type) in float captions. 1	Ampersand (&): Fixed handling when passed as an argument. . 432	
	Test Suite: Sub-figures 1	Docs: Added warning icons for items needing special attention. 205	
v0.13	\CaptionSeparator: Fix for newer babel package. 506	Docs: Clarify print/HTML output. 128	
	\LWR@LwarpStart: \up and \fup . . 394	Docs: Moved the supported features table to the introduction. 79	
	General: 2016/03/24 1	Files: lwarp_formal.css added. . . . 1	
	Fix dollar-redefined bug for newer package. 944	Fix: steps counter 748	
	Removed package: subfig 1	Fixed & handling. 746	
	Test Suite: Ordinals, Subcaption . . 1	Test Suite: test_suite_formal.css file added. 1	
v0.14	\LWR@htmlsectionfilename: Fix: Links to home page. 335	v0.16	General: 2016/04/11 1
	General: 2016/03/31 1	\titlingpage: Improved print-output spacing. 403	
	floatrow: Added. 744	xfrac: Adjusted for the use of any font. 1003	
	Docs: Commands for a successful HTML conversion. 132	Added XeLaTeX, LuaLaTeX support. 206	
	Docs: Commands into a warpprint environment. 129	Docs: Font and UTF-8 support. . 113	
	Docs: Newclude limitations. . . . 176	Docs: Moved location of \usepackage{lwarp}. 116	
		Docs: Text not converting. 197	
		Lwarp no longer selects fonts. 113, 231	
		Removed package: suffix 1	

Test Suite: Improved titlingpage.	403	verse: Supports verse, memoir packages.	980
Test Suite: Lwarp no longer selects fonts.	1	minipage: Fix: \linewidth, \textwidth, \textheight inside a minipage.	580
Test Suite: Supports XeLaTeX, LuaLaTeX.	1		
v0.17		v0.19	
\LWR@htmlsectionfilename: Fix: Links when entire doc is one HTML page.	335	\HTMLfilename: Docs: Escape filename underscores.	334
General: 2016/04/14	1	\HomeHTMLfilename: Docs: Escape filename underscores.	334
mdframed: Added.	820	\LWR@LwarpStart: Enabled \\ equal to \newline.	393
Test Suite: Fix: Print-version front-matter page numbers.	1	\LWR@doequation: MATHJAX support.	544
Test Suite: Mdframed	1	\LWR@doubledollar: MATHJAX support.	537
v0.18		\LWR@filestart: l warp_mathjax.txt loaded.	390
\LWR@includegraphicsb: Add: svgz file extension.	769	\LWR@minipagestartpars: Suppresses paragraph tags between minipages.	600
em, ex, %, px dimensions preserved.	769	\LWR@subsingledollar: MATHJAX support.	531
Fix: \linewidth, \textwidth, \textheight inside a minipage.	769	\LateximageFontSizeName: Add: User-adjustable math/lateximage font size.	554
Improved HTML output linebreaks.	769	\hspace: Fix: \hspace length computations.	602
\LWR@myshorttoc: Reorganize \HomeHTMLfilename logic.	510	\minipagefullwidth: Added: No width tag for the next minipage in HTML.	579
\LWR@newhtmlfile: sidetoc after title, improving responsive design.	372	\warpHTMLonly: Added.	230
\LWR@requesttoc: Reorganize \HomeHTMLfilename logic.	396	\warpprintonly: Replaces \rowprintedonly.	230
\LWR@subhyperref: Improved HTML output linebreaks.	498	\xfracHTMLfontsize: Added.	1003
\LWR@subhyperrefclass: Improved HTML output linebreaks.	499	General: 2016/05/19	1
\LWR@subinlineimage: Suppress extra space.	500	File: l warp.css: Improved toc outline display.	1
\hspace: \hspace supported.	602	Files: l warp.css and l warp_formal.css: Improved responsive design.	1
General: 2016/05/19	1	Microtype disabled during HTML generation	231
File: l warp.css: Improved toc outline display.	1	PDF Unicode input characters.	220
Files: l warp.css and l warp_formal.css: Improved responsive design.	1	Test Suite: Verse package	1
Microtype disabled during HTML generation	231	\lateximage: pdfcrop: --hires added.	558
PDF Unicode input characters.	220	Reorganize \HomeHTMLfilename logic.	558
Test Suite: Verse package	1	Suppress extra space.	558

Introduction: MATHJAX support	
mentioned.	76
Options: mathsvg and mathjax	225
Supports colored \rule.	993
titleps: null \pagestyle and	
\thispagestyle for HTML. . .	945
v0.20	
\BlockClassSingle: Renamed from	
"LWR@htmldivclassline".	348
\HTMLDescription: Added	
\NewHTMLdescription.	
(Renamed in v0.30)	360
\HTMLFilename: No longer escape	
underscores.	334
\HomeHTMLFilename: No longer	
escape underscores.	334
\InlineClass: Renamed from	
"inlineclass".	349
\LWR@LwarpStart: Fix: math cross	
references.	394
\LWR@closeparagraph: \unskip	
extra spaces.	353
No break tags in the start/end of a	
tabular.	353
\LWR@endofline: Fix: \\	600
\LWR@filestart: Adds meta	
description.	390
\LWR@htmldivclass: Added	
optional style.	347
\LWR@htmlelementclass: Added	
optional style.	346
\LWR@htmlsectionfilename:	
HTMLFilename: removed	
additional trailing '-', and may be	
empty.	335
Sections called "Index" or "index"	
have an underscore prepended to	
their filenames if no prefix. . .	335
\LWR@includegraphicsb: Fix:	
\linewidth in a floatrow.	769
Fix: Expands filename.	769
\LWR@longtabledatacaptiontag:	
Fix: Parses in captions.	472
\LWR@section: Combined	
higher-level sections together	
into files.	380
\LWR@setOSWindows: Auto-detects	
operating system.	223
\LWR@subhtmlelementclass:	
Factored code.	346
\SetHTMLFileName: Add: Control	
file numbers.	335
\cpagerefFor: User-redefinable	
word for page references.	576
\dotfill: Inserts an ellipsis.	599
\hfill: Inserts a \quad.	599
\hrulefill: Inserts a short rule. . .	599
\hspace: Add: Supports HTML thin	
breakable space.	602
\hyperindexref: Print mode	
provided in case hyperref not	
used.	521
\pageref: Added.	497
\tracingl warp: Added.	243
General: 2017/02/09	1
afterpage: Added.	623
alltt: Added.	629
bookmark: Added.	657
caption and subcaption	
supported.	1
cleveref and referencing patches:	
Applied \AfterEndPreamble. .	575
draftwatermark: Added.	709
eso-pic: Added.	720
everypage: Added.	721
extramarks: Added.	722
fancyhdr: Added.	728
float: Improved float caption type	
handling.	741
hyperref: Additional user macros. .	782
keyfloat: Added.	797
letterspace: User-interface	
emulated.	800
listings: Added.	805
ltcaption: Added.	813
lwarp-newproject: Added.	256
microtype: User-interface	
emulated.	833
needspace: Added.	846
nowidow: Added.	849
placeins: Added.	875
ragged2e: Added.	883
setspace: Improved support. . .	904
textpos: Added.	936
titleps: Added.	945
titlesec: Added.	949
titletoc: Added.	951
titling: Improved compatibility. .	953
tocloft: Added.	961
wallpaper: Added.	986

wrapfig: Added.	988	<i>lwarpmk</i> : Fix: <i>lwarpmk</i> uses	
xetexko-vertical: Added.	1002	latexitimages text file instead of	
Added @, <, > columns.	426	shell script.	308
Added single-expansion data arrays.	330	Add: Errors for misplaced	
Code factored into independent <i>l warp_html</i> files.	614	packages.	207
Docs: Examples for generating HTML file names.	126	Docs: Added internet class.	85
Docs: Improved index.	1	Docs: Added TeX2page, GladTeX. . .	85
Enhanced titling support.	401	Docs: Installing on WINDOWS.	92
File: <i>l warp.css</i> : Minor fixes for validation.	1	File: <i>l warp_tutorial.txt</i> added. . .	96
File: <i>l warpmk</i> used to compile print, HTML, indexes, and latexitimages.	1	v0.22	
Fix: \linewidth in a floatrow. . .	746	\LWR@parseDcolumn: Added tabular D column.	442
Moved sidebar and example code to test suite.	1	\LWR@parsebangcolumn: Added tabular ! column.	437
Page geometry set to 6in wide with large margins.	232	\LWR@parsetablecols: Unknown table column types become l.	
Parallel versions of aux files for print/HTML.	1	Added tabular D, !, X columns. . .	444
Removed reliance on make, grep, gawk.	1	\LWR@printmccoldata: Added tabular D, !, and X columns. . .	466
Tabular: \unskip extra spaces. . .	426	General: 2017/03/02	1
Test Suite: HTML meta descriptions.	1	abstract: Added.	616
verbatim: Added.	413	changepage: Added.	669
BlockClass: Added optional style. .	348	dcolumn: Added.	706
Renamed from "blockclass". . .	348	ftnright: Added.	758
\LWR@nestspan: Fix: Minipages inside a span.	343	geometry: Nullified commands. .	760
v0.21		indentfirst: Added.	795
\LWR@LwarpStart: Changed latexitimages to a .txt file. . .	393	layout: Added.	800
\LWR@filestart: Skip title if not given.	390	lscape: Added.	812
\LWR@newhtmlfile: Skip title if not given.	372	mcaption: Added.	820
\marginpar: Fixed source listing. .	366	nameref: Added.	844
\marginparBlock: Fixed source listing.	366	nextpage: Added.	846
General: 2017/02/23	1	parskip: Added.	867
fontenc: Added.	752	showkeys: Added.	906
<i>lwarpmk</i> : Fix: <i>lwarpmk</i> again for WINDOWS.	308	sidecap: Added.	906
<i>lwarpmk</i> : Fix: <i>lwarpmk</i> <i>limages</i> for WINDOWS.	308	tabularx: Added.	930
		variorref: Supported.	140
		verse: Added.	980
v0.23		\LWR@parsetablecols: Fix for vert bar column type.	444
		\LWR@printmccoldata: Fix for vert bar column type.	466
		General: 2017/03/02	1
v0.24		\LWR@htmlfileref: Fix: Index links while \tracingl warp.	493
		\hspace: Add: \hspace \fill converts to 2em	602

\hypertocfloat: List of floats	
responds to <code>lofdepth</code> ,	
<code>lotdepth</code>	517
General: 2017/03/15	1
<code>floatrow</code> : Support for <code>subfig</code>	744
<code>subfig</code> : Added.	921
<code>tikz</code> : For <code>tikz v3.0.0</code> or later,	
auto-loads <code>tikz babel</code> library if	
necessary.	944
Docs: Filename underscore.	120, 134
Fix for inline images.	944
No longer preloads <code>subcaption</code> ;	
conflicted with <code>subfig</code>	234
<code>picture</code> : Fix for inline images.	578
v0.25	
\LWR@loadnever: Added the ability	
to prevent conflicting packages.	208
\addcontentsline: Handles	
theorems.	509
General: 2016/03/22	1
<code>amsthm</code> : Added.	636
<code>ellipsis</code> : Added.	711
<code>emptypage</code> : Added.	712
<code>framed</code> : Added.	756
<code>lips</code> : Added.	804
<code>mdframed</code> : Help avoid	
hyphenation.	822
<code>ntheorem</code> : Added.	849
<code>showidx</code> : Added.	906
<code>theorem</code> : Added.	937
Basic L ^A T _E X theorems: improved	
<code>css</code>	414
Docs: Adds credits for patched	
code.	1
Docs: Testing <code>lwarp</code>	195
Fix: Allows XE ^L _A T _E X and Lua ^L _A T _E X	
to preload graphics and	
<code>graphicx</code>	210
v0.26	
General: 2017/03/31	1
<code>lwarp.css</code> : Improved responsive	
<code>marginpar</code> and <code>marginblock</code>	264
<code>cutwin</code> : Added.	704
<code>endnotes</code> : Added.	714
<code>floatflt</code> : Added.	742
<code>footmisc</code> : Added.	752
<code>footnotehyper</code> : Added.	754
<code>footnote</code> : Added.	753
<code>marginfix</code> : Added.	819
<code>marginnote</code> : Added.	819
mparhack: Added.	836
pagenote: Supported as-is.	864
sidenotes: Added.	907
Docs: Improved MiK _T E _X install	
instructions.	90, 92
Dollar span avoided in a	
<code>lateximage</code>	530
Footnotes now are L ^A T _E X boxes	
instead of pagenotes.	361
\latextimage: Labels track page	
numbers of <code>lateximages</code>	558
Print mode now uses a <code>minipage</code>	
of <code>\ linewidth</code>	558
<code>picture</code> : Fix for <code>\ makebox</code> in	
<code>picture</code>	578
v0.27	
\LWR@footnotetext: Fix for table	
footnote par tags.	362
General: 2017/04/04	1
<code>letrine</code> : Added.	801
<code>microtype</code> : Fix with Xe ^L _A T _E X,	
Lua ^L _A T _E X.	833
<code>soul</code> : Added.	915
<code>ulem</code> : Added.	975
Docs: Installing utilities for	
MACOS.	94
Docs: Limitations of <code>saveboxes</code>	135
Page geometry modified to reduce	
line overflow.	232
v0.28	
\@rindex: Improved indexing.	520
\HTMLAuthor: Added \HTMLauthor.	
(Renamed in v0.30.)	359
\LWR@LwarpEnd: If FormatEPUB or	
FormatWP, no bottom nav.	396
\LWR@LwarpStart:	
FormatWordProcessor forces	
single-file output.	393
\LWR@filestart: Adds HTML meta	
author.	390
\LWR@forcenewpage: Forces new	
PDF page before major	
environments.	340
\LWR@htmlcomment: Breaks ligatures	
in HTML comments.	345
\LWR@includegraphicsb: Adapts to	
<code>graphics</code> syntax.	769
\LWR@newhtmlfile: If FormatEPUB	
or FormatWP: skips headers,	
footers, nav.	372

\LWR@parsetablecols: Added L, C, R, J column types.	444
\LWR@startref: Removed space. . .	495
\chapter: If EPUB, prints footnotes before each section.	387
\hyperindexref: Improved indexing.	521
\textup: Fixed span class.	590
General: 2017/04/14	1
glossaries: Added.	762
graphics: Added.	763
tabularx: Fix for optional pos. .	930
tabulary: Added.	930
lwarpmk: Add: <i>printglossary</i> and <i>htmlglossary</i> commands. .	308
Added boolean FormatEPUB. . . .	249
Added boolean FormatWP.	249
Added boolean HTMLDebugComments.	243
Added boolean HTMLMarkFloats, changed to WPMarkFloats as of v0.42.	249
Docs: Modfyiing lwarpmk and index processing.	196
File: l warp_mathjax.txt: Updated CDN repository. . . .	306
Forced oneside to maintain large right margin.	232
v0.29	
\lWR@includegraphicsb: Fix: Error when no optional arguments. .	769
General: 2017/04/15	1
*.lwarpmkconf: Add: language option for config files. . . .	264
lwarpmk.conf: Add: language option for config files. . . .	264
lwarpmk: Add: language option for config files.	308
Add: lwarpmklang option for l warp.	226
Docs: Using a glossary	109
v0.30	
\CSSFilename: Renamed from \NewCSS.	358
\HTMLAuthor: Renamed from \HTMLAuthor.	359
\HTMLDescription: Renamed from \NewHTMLdescription. . . .	360
\HTMLFirstPageTop: Renamed from \SetFirstPageTop.	358
\HTMLLanguage: Renamed from \MetaLanguage.	389
\HTMLPageBottom: Renamed from \SetPageBottom.	358
\HTMLPageTop: Renamed from \SetPageTop.	358
General: 2017/04/29	1
l warp-newproject removed, and combined with l warp.	256
lwarpmk: Add: xdyfile configuration option.	308
lwarpmk: Fix: <i>xindy</i> and <i>texindy</i> adjusted for <i>pdflatex</i> , <i>xelatex</i> and <i>lualatex</i>	308
lwarpmk: Fix: <i>xindy</i> now used for print index generation with <i>latexmk</i>	308
lwarpmk: language now used for both index and glossary generation.	308
File: l warp_html.xdy renamed to l warp.xdy.	305
Fix: *.css files only written in print mode.	264
Fix: l warp.xdy only written in print mode.	305
Fix: l warp_mathjax.txt: Only written in print mode.	306
Option lwarpmklang changed to IndexLanguage.	226
Option OSWindows replaces macro \warpOSwindows.	226
Option xdyFilename added. . . .	226
Option latexmk replaces macro \UseLatexmk.	228
Options HomeHTMLfilename and HTMLfilename replace macros \HomeHTMLfilename and \HTMLfilename.	227
v0.31	
General: 2017/05/15	1
keyfloat: Improved compatibility. .	797
v0.32	
\RequirePackage: Fix: Ignores blanks in package list.	238
General: 2016/06/09	1
glossaries: Prevent error with \glo@name not defined.	522

<i>lwarpmk</i> : Fix: <code>io.lines()</code> changed to <code>file:lines()</code> due to <i>luatex</i> changes.	308
v0.33	
\HTMLAuthor: Fix: Provides empty default author if none given.	359
\LWR@loadbefore: Fix: No \PackageError if already loaded.	208
\LWR@parseatcolumn: Fix: Column alignment with leftmost @.	436
\LWR@tableddatasinglecolumntag: Fix: Macros in tabular could cause extra data cell.	452
\LWR@vspace: Add: \vspace nullified.	603
\StartDefiningTabulars: Add: Avoids error: Misplaced alignment tab character &.	331
General: 2017/07/10	1
amsmath: Removed <code>fleqn</code> option.	629
fancyhdr: Fix: Optional args for \lhead, etc.	728
Add: Tabular at and bang columns now have their own HTML columns.	426
cleveref: Fix: Loaded \AtEndPreamble.	575
Fix: Incorrectly-inline math environments.	549
New handling of & to localize catcode changes.	426
v0.34	
\@fnsymbol: Text symbols instead of math.	404
\InlineClass: Moved optional argument in front of mandatory.	349
\LWR@htmldivclass: Moved optional argument in front of mandatory.	347
\LWR@htmlelementclass: Moved optional argument in front of mandatory.	346
\LWR@htmlelementclassline: Moved optional argument in front of mandatory.	347
\LWR@htmlspanclass: Moved optional argument in front of mandatory.	344
\LWR@nullfonts: Improved font control.	592
\LWR@restoreorigformatting: booktabs: Works inside \lateximage.	523
Improved font control.	523
\LWR@subhtmlelementclass: Moved optional argument in front of mandatory.	346
\LWR@tableddatacolumntag: booktabs: Works inside \lateximage.	480
\fboxBlock: Added.	586
\makebox: Fix: Handles paren arg.	584
General: 2017/08/08	1
babel-french: Adds fixed-width HTML spaces to punctuation.	341
balance: Added.	652
booktabs: Works inside \lateximage.	483, 657
boxedminipage2e: Added.	659
crop: Added.	703
enumerate: Added.	714
enumitem: Added, no longer required.	714
everyshi: Added.	721
fancybox: Added.	724
fancyvrb: Added, no longer required.	731
figcaps: Added.	737
filecontents: Required. Patched for morewrites.	233
floatpag: Added.	743
flushend: Added.	749
fullpage: Added.	759
hyperxmp: Added.	788
idxlayout: Added.	789
marginfit: Added.	818
mdframed: Improved <code>mdtheorem</code> patch.	828
moreverb: Added.	835
paralist: Added.	865
pdfescape: Added.	868
pdfsync: Added.	872
prelim2e: Added.	877
rotfloat: Added.	890
savetrees: Added.	892
shadow: Added.	905
syntonly: Added.	929
titleps: No longer required.	945

titleref:	Prevented.	948
xmpincl:	Added.	1006
Added.		998
Docs:	Horizontal space limitations.	1
Docs:	Misplaced alignment character.	197
File:	<i>l warp_mathjax.txt</i> : Version change.	306
File:	<i>README.txt</i> : updated.	1
Fix:	Added the <i>eqnarray</i> environments.	549
Improved font control.		589
Lists refactored to remove enumitem requirement.		416
Verbatim refactored to remove fancyvrb requirement.		411
tabular:	<i>booktabs</i> : Works inside <i>latextimage</i> .	485
latextimage:	Fix: <i>latextimage</i> with <i>minipage</i> , <i>\parbox</i> , <i>\makebox</i> , <i>\fbox</i> , <i>\framebox</i> , <i>\raisebox</i> , <i>\scalebox</i> , <i>\reflectbox</i> .	558
BlockClass:	Moved optional argument in front of mandatory.	348
fminipage:	Added.	586
\LWR@nestspan:	Fix: Minipages, BlocksClass, and lists inside a span.	343
v0.35		
General:	2017/08/08	1
	Fix: <i>\textbf</i> and related.	589
v0.36		
\LWR@HTMLsanitize:	Fix for babel-french.	554
\LWR@HTMLsanitizeexpand:	Fix for babel-french.	555
\LWR@closeparagraph:	Extra HTML source space after paragraphs.	353
\LWR@currenttextcolor:	Fix for \rule when xcolor not loaded.	597
\LWR@footnotetext:	Extra HTML source space after paragraphs. Force HTML superscripts.	362
\LWR@nullfonts:	Fix: Filenames while using MATHJAX.	592
\LWR@restoreorigformatting:	siunitx: Improved super/subscripts in a <i>latextimage</i> .	523
\LWR@section:	Improved spacing.	380
\LWR@stopars:	Extra HTML source space after paragraphs.	356
\fbox:	Fix: Uses <i>\fboxrule</i> and <i>\fboxsep</i> .	585
\framebox:	Fix: Handles width and horiz position.	585
\makebox:	Fix: Handles width and horiz position.	584
General:	2017/08/17	1
babel-french:	Adjustements for French variants, load order, footnotes, ellipses.	341
footnote:	Extra HTML source space after paragraphs.	753
siunitx:	Fix for babel-french.	567
siunitx:	Improved symbol support.	910
transparent:	Added.	971
upref:	Added.	979
xcolor:	Added <i>\fcolorboxBlock</i> , <i>\colorboxBlock</i> .	989
xcolor:	Fix: Background none in print mode.	989
xcolor:	Refactored \LWR@colorstyle.	993
xcolor:	Uses <i>\fboxrule</i> and <i>\fboxsep</i> .	989
xcolor:	<i>\fcolorbox</i> etc. now work inside <i>latextimage</i> .	989
Docs:	Reorganized: Special cases and limitations.	132
Source:	Improved formatting.	1
tabular:	Fix for babel-french.	485
latextimage:	Footnotes appear in regular text instead of the <i>latextimage minipage</i> .	558
v0.37		
\include:	Maintains independent aux files for HTML.	241
General:	2017/08/19	1
LATeX accents:	Added.	254
babel-french:	Adjustment for load order.	341
color:	Prevented.	700
comment:	Maintains independent cutfiles for print, HTML.	230
siunitx:	Improved symbol support.	910
textcomp:	Improved support.	933
<i>l warpmk</i> :	Removes additional HTML aux files.	308

File handles reorganized.	240
v0.38	
\@seccntformat: Added for	
appendix.	379
\ForceHTMLPage: Added.	377
\ForceHTMLTOC: Added.	377
\LWR@section: \part* starts a new	
HTML page, for appendix.	380
Modified spacing, uses	
\newline.	380
\newline: Added trailing \quad.	516
\part: Fix with article class.	386
General: 2017/08/27	1
appendix: Added.	642
arabicfront: Added.	642
chappg: Added.	670
color: Forces xcolor as well.	700
fix2col: Added.	738
fncychap: Added.	750
grffile: Added.	777
metalogo: Added.	830
nonumonpart: Added.	848
nopageno: Added.	848
pagenote: Option page disabled.	864
realscripts: Added.	884
relsize: Added.	886
romanbarpagenumber: Added.	889
romanbar: Added.	889
scalefnt: Added.	892
siunitx: Removed from lwp core.	910
textcomp: Removed from lwp	
core.	933
tocbibind: Added.	959
xltextra: Added.	1005
<i>lwarpmk</i> : Added <i>print1</i> and	
<i>html1</i> actions.	308
Added \markboth, \sloppy, etc.	339
Docs: Enhanced <i>Supported</i>	
<i>Features</i> table.	79
Docs: Index, tocbibind.	146
Docs: Starred sections.	142
v0.39	
\@maketitle: titling version.	956
Native LATEX version.	405
Removed minipages.	405, 956
Supports authblk with <div>s of	
class oneauthor instead of	
tabular.	405, 956
\AddSubtitlePublished: Added.	407
\LWR@domulticolumn: Add: Optional	
vpos and # rows.	469
\LWR@restoreorigformatting:	
Appended with \appto instead of	
calling various macros.	523
\LWR@tabledataacolumntag: Don't	
start a data cell if see	
\TabularMacro.	480
\ResumeTabular: Added.	478
\TabularMacro: Added.	478
\multicolumnrow: Added.	476, 841
\printauthor: Removed minipages.	402
Supports authblk with <div>s of	
class oneauthor instead of	
tabular.	402
\thanksmarkseries: Removed	
minipage footnotes.	957
General: 2017/09/05	1
a4wide: Added.	616
a4: Added.	615
a5comb: Added.	616
addlines: Added.	623
any size: Added.	641
authblk: Added.	651
bigdelim: Added.	654
bigstrut: Added.	655
ebook: Added.	710
fullwidth: Added.	759
midpage: Added.	834
multirow: Add: New optional vpos	
argument.	839
multirow: Add: Supports left/right	
border for bigdelim.	839
multirow: Fix: Long text	
argument.	839
supertabular: Added.	927
textarea: Added.	932
titling: Improved compatibility.	953
titling: Removed extraneous	
center environments.	954
typearea: Added.	974
xtabular: Added.	1007
zwpagelayout: Added.	1011
Docs: Reorganized tabular	
discussion.	168
Titlepage \published and	
\subtitle removed.	
\AddSubtitlePublished	
restores.	406
titlepage: Clear pending footnotes.	400

Removed minipages.	400	sectsty: Added.	902
titlingpage: Clear pending footnotes.	953	stabular: Added.	920
v0.40		tabls: Added.	929
@chapcntformat: Added for tocbibind, anonchap.	379	textcomp: Additional symbols, improved XeLaTeX and LuaLaTeX support.	933
\LWR@HTMLhline: Added.	484	tocbibind: Improved for \simplechapter.	959
\LWR@includegraphicsb: Add: Full <code>\graphicspath</code> support.	769	xltxta: Fix for <code>\showhyphens</code> with XeLaTeX.	1005
\LWR@nullfonts: Fix: Long arguments for expandable command.	592	No longer preloads <code>xfrac</code>	235
\LWR@restoreorigformatting: Improved LATEX logos inside a <code>lateximage</code>	523	v0.41	
Improved symbols inside a <code>lateximage</code>	523	\LWR@addcmidruletrim: Add: <code>\cmidrule</code> trims.	457
Nullified <code>\InlineClass</code> , etc. inside a <code>lateximage</code>	523	\LWR@clearcmidrules: Add: <code>\cmidrule</code> trims.	454
\LWR@tabledatacolumntag: Fix for <code>bigdelim</code> : <code>\ldelem</code> , <code>\rdelem</code> . . .	480	\LWR@closetabledatacell: Add: Mute <code>></code> for <code>\bottomrule</code>	430
\chapter: Added support for <code>quotchap</code>	387	Fix: At/bang column with <code>\multirow</code>	430
\multicolumnrow: Fix: Adapts to older <code>multirow</code> and <code>xparse</code> . . .	476	Fix: Cancel <code><</code> for <code>\multicolumn</code>	431
\simplechapterdelim: Added for tocbibind, anonchap.	379	\LWR@domulticolumn: Add: <code>\cmidrule</code> trims.	469
\underline: Added.	597	Added vertical rules.	470
General: 2017/09/25	1	\LWR@nullifyNoAutoSpacing: Fix: <code>\NoAutoSpacing</code> in a <code>tabular</code> with <code>babel-french</code>	484
adjmulticol: Added.	622	\LWR@parsebarcolumn: Added vertical rules.	439
anonchap: Added.	641	\LWR@printatbang: Add: <code>\cmidrule</code> trims.	450
bigdelim: Improved documentation.	654	Add: Mute at and bang columns for <code>\bottomrule</code>	450
cuted: Added.	704	\LWR@printbartag: Added vertical rules.	450
dblfnote: Added.	705	\LWR@subaddcmidruletrim: Added.	457
fnpos: Added.	751	\LWR@subcmidrule: Add: <code>\cmidrule</code> trims.	455
graphics: Moved out of the <code>lwrap</code> core.	763	\LWR@tabledatasinglecolumntag: Add: <code>\cmidrule</code> trims.	452
graphics: Restores <code>\includegraphics</code> and <code>\DeclareGraphicsExtensions</code> in a <code>lateximage</code>	763	Add: Mute <code><</code> for <code>\bottomrule</code>	452
graphicx: Moved out of the <code>lwrap</code> core.	777	\LWR@tabularfinishrow: Unfinished tabular rows automatically filled.	433
grffile: Directly supported.	777	\mcolrowcell: Added for <code>\multicolumnrow</code> cells.	483
midfloat: Added.	834	General: 2017/10/07	1
multirow: Improved <code>bigdelim</code> borders.	839	multirow: Add: <code>\cmidrule</code> trims.	839
pfnote: Added.	874	Added vertical rules.	840
quotchap: Added.	882	Fix: <code>< spec</code>	840

Improved rules.	657
tabular: Fix: \NoAutoSpacing in a tabular with babel-french.	485
v0.42	
\@ensuredmath: Improved	
\ensuremath.	539
\@textsubscript: Added.	596
\@textsuperscript: Added.	596
\LWR@HTMLhline: If FormatWP force explicit border.	484
\LWR@addformatwpalignment: If FormatWP add explicit style for cell alignment.	459
\LWR@addrulewidth: If FormatWP force explicit border.	458
\LWR@amsmathbody: Fix: Numbering and naming AMS math environments.	557
\LWR@amsmathbodynumbered: Fix: Numbering and naming AMS math environments.	558
\LWR@doequation: If FormatWP print LaTeX expression.	544
\LWR@domulticolumn: If FormatWP add cell alignment.	471
\LWR@doubledollar: If FormatWP print LaTeX expression.	537
Improved \ensuremath.	537
Improved line spacing with mathjax.	537
\LWR@floatbegin: If FormatWP add a text frame.	503
\LWR@floatend: If FormatWP add a text frame.	504
\LWR@htmlmathlabel: If FormatWP print LaTeX expression.	548
\LWR@includegraphicsb: Fix: Filename expansion.	769
If FormatWP, use explicit size.	770
\LWR@remembertag: Fix: Numbering and naming AMS math environments.	557
\LWR@restoreorigformatting: Improved \ensuremath.	525
\LWR@subaddcmidruletrim: Opt if no rule given.	457
\LWR@subsingledollar: If FormatWP print LaTeX expression.	531
\LWR@tabledatasinglecolumntag:	
If FormatWP add cell alignment.	452
\LaTeX: If FormatWP use explicit style.	608
\TeX: If FormatWP use explicit style.	607
\hspace: If FormatWP add \quads.	602
\listoffigures: Added boolean WPMarkLOFT.	513
\listoftables: Added boolean WPMarkLOFT.	513
\marginpar: If FormatWP emulate a wrapfig.	366
\marginparBlock: If FormatWP emulate a wrapfig.	366
\rule: If FormatWP add \quads.	606
\tableofcontents: Added boolean WPMarkTOC.	513
\underline: If FormatWP, use explicit styles for \underline, etc.	597
General: 2017/10/30	1
\textbf and related: If FormatWP, use explicit styles for \textsc, etc.	589
algorithmicx: If FormatWP add \quads.	628
epigraph: If FormatWP add HTML styles.	715
fancybox: If FormatWP add HTML styles.	724
floatflt: Added width.	742
includegraphics: Fix: Class key.	768
keyfloat: If FormatWP add explicit HTML style.	798
moreverb: Simplified formatting of listings.	835
multirow: If FormatWP add cell alignment.	840
overpic: Added.	863
realscripts: Fix for subscripts in a lateximage.	884
sidenotes: If FormatWP add explicit HTML style.	908
siunitx: Improved \ensuremath.	910
soul: If FormatWP, add explicit styles.	915
textcomp: Improved \interrobangdown.	933
wrapfig: If FormatWP add explicit HTML style.	988

Added boolean WPMarkLOFT.	250
Added boolean WPMarkMath.	250
Added boolean	
WPMarkMinipages.	249
Added boolean WPMarkTOC.	250
Added boolean WPTitleHeading.	250
Docs: Added support page.	2
Docs: Improper \prevdepth.	197
Docs: Reorganized math	
limitations	159
File: l warp_mathjax.txt:	
Updated siunitx script.	306
Fix: Numbering and naming AMS	
math environments.	556
If FormatWP force explicit border.	657
If FormatWP, shift section	
headings.	251
tabbing: Added.	414
latextimage: Fix: Numbering and	
naming AMS math	
environments.	559
center: If FormatWP use explicit	
text-align.	564
minipage: Added boolean	
WPMarkMinipages.	582
If FormatWP add a text frame.	581
eqnarray: Fix: Numbering and	
naming AMS math	
environments.	551
If FormatWP print LaTeX	
expression.	550
LWR@BlockClassWP: Added to factor	
code.	349
LWR@figcaption: If FormatWP forces	
italic captions.	507
v0.43	
\LWR@domulticolumn: Fix for	
vertical rules.	470
Fix: Multicolumn trim.	469, 470
\LWR@maybeprintpendingfootnotes:	
Added FootnoteDepth.	365
\LWR@nullfonts: Fix: Nullify dollar	
inside filenames.	592
\LWR@parsetablecols: Ignore	
spaces in col spec.	444
\LWR@printmccoltype: Added	
vertical rules.	465
\LWR@section: Fix: Expansion in	
comparison.	381
Fix: Math in section name.	382, 385
Fix: Nullify fonts inside HTML	
comment.	382
\TabularMacro: \newcommand	
instead of \relax to fix	
supertabular and xtab.	478
\href: Made robust.	499
\nameref: Made robust.	497
\nolinkurl: Made robust.	499
\url: Made robust.	500
General: 2017/11/08	1
breakurl: Added.	660
hyperref: Made robust.	782, 784, 786
hyperref: \Gauge added.	787
luatodonotes: Added.	815
todonotes: Added.	969
LWR@currentautosec: Added.	380
Added FootnoteDepth.	361
Docs: HTML settings table.	120
Docs: Reorganized HTML	
customization.	120
v0.44	
\@currentlabelname: Adjustment	
for koma-script.	490
\HTMLTitle: Added.	359
\LWR@addformatwpalignment: Fix	
for multicolumn alignment if	
FormatWP.	459
\LWR@backgroundcolor: Added.	994
\LWR@filestart: Add \HTMLTitle.	392
Fix \HTMLAuthor.	391
\LWR@listitem: Added list and	
trivlist.	418
\LWR@patchlists: Added list and	
trivlist.	421
\LWR@strresult: Fix:	
\providecommand.	429
\LWR@textcurrentcolor: xcolor:	
Added	
\LWR@textcurrentcolor.	993
\addcontentsline: Automatic	
\LWR@newfloatanchor.	509
\chapter: Add preamble for	
koma-script.	387
\marginparBlock: Added.	366
\nopagecolor: xcolor: Fix for	
\nopagecolor.	995
\part: Add preamble for	
koma-script.	386
\title: Added \thetitle.	359
General: 2017/11/22	1

algorithmicx: Improved comment symbol.	628	list: Added list and trivlist.	419
atbegshi: Added.	646	\LWR@nestspan: Added list and trivlist.	343
cancel: Added.	663	v0.45	
changepage: Additional options.	669	\@currentHref: Added.	498
easy-todo: Added.	709	\@donoparitem: Modified for HTML.	416
fancyref: Added.	729	\@item: Modified for HTML.	416
fixmetodonotes: Added.	739	\@mklab: Modified for HTML.	416
fixme: Added.	738	\CSSFilename: Improved filenames with underscores.	358
fontenc: Allowed after l warp.	752	\LWR@LwarpStart: Fix: Lateximages on incorrect pages with Mathjax.	395
hang: Added.	779	\LWR@filenamenoblanks: Fix: Section names with \ .	369
ifoddpage: Added.	790	Fix: Section names with underscores.	369
ltxtable: Added.	814	\LWR@includegraphicsb: Improved URLs with underscores.	769
luatodonotes: Improved.	815	\LWR@newautoidanchor: Fix: No anchor if frozen autoid.	505
l warp-patch-komascript: Added.	1013	\LWR@notmemoirloadafter: Added.	207
overpic: Fix: Groups for lateximages.	863	\LWR@printpendingmpfootnotes: Added.	365
pdfsync: Fixes.	872	\LWR@startref: Fix: Labels with underscores.	496
preview: Added.	877	\LWR@subhyperref: Improved URLs with underscores.	498
scrextend: Added.	893	\LWR@subhyperrefclass: Improved URLs with underscores.	499
scrhack: Added.	896	\LWR@sublabel: Fix: Labels with underscores.	493
scrlayer-notecolumn: Added.	898	\LWR@tabledatacolumntag: Fix: Empty line between rows.	482
scrlayer-scrpage: Added.	898	\chapter: Add optional heading title for memoir.	387
scrlayer: Added.	896	\newpage: Added.	600
section: Added.	901	\nolinkurl: Fix: Underscore in URL.	499
soulpos: Added.	917	\normalmarginpar: Added.	367
soulutf8: Added.	918	\reversemarginpar: Added.	367
supertabular: Fix for caption.	928	\section: Add optional heading title for memoir.	387
tikz: Fix: Groups for lateximages.	944	\tableofcontents: Fix: Empty sidetoc.	512
tocbasic: Added.	958	Fix: Patch \AtBeginDocument.	512
tocloft: Added \newlistentry.	966	\url: Improved URLs with underscores.	500
tocloft: Improved \newlistof.	966	General: 2018/01/14	1
tocstyle: Added.	967	array: Added.	643
todonotes: Improved.	969	babel-french: Robust commands.	341
todo: Added.	968	backref: Added.	652
typearea: Added expert commands.	974		
watermark: Added.	987		
xcolor: Added			
\LWR@currenttextcolorstyle.	993		
xcolor: Added			
\LWR@findcurrenttextcolor.	993		
xtab: Fix for caption.	1008		
Adjustment for koma-script.	217		
AMS environments: Fix: Groups for lateximages.	630		
If pdfLaTeX, require T1 and UTF-8 encoding.	219		
picture: overpic: Fix: Groups for lateximages.	578		

breakurl: Fix: Underscore in URL	660	v0.46	
changebar: Added	668	\LWR@closeparagraph: Fix: Tabular	
cite: Added	698	empty lines	354
continue: Added	702	\LWR@closeprevious: Fix: Stack	
endfloat: Added	712	unnesting	339
epigraph: Support for memoir	716	\LWR@forcenewpage: Fix: Improper	
fancyvrb: Improvements	731, 732	\prevdepth	340
flafter: Added	740	\LWR@lookforpackagename: Fix:	
fltrace: Added	749	Spaces in \usepackage	238
footnpag: Added	755	\LWR@providelength: Added	221
fwlw: Added	759	\LWRPrintStack: Name changed	
hanging: Added	780	from \PrintStack	338
hyperref: Fix: Underscore in		\makebox: Fix: Lateximage in a	
URL	782, 783	\makebox	584
l warp-patch-memoir: Added	1016	\popclose: Fix: Stack unnesting	329
memhfixc: Added	830	\pushclose: Fix: Stack unnesting	328
memoir: Added	611	General: 2018/01/23	1
natbib: Added	844	\LWR@tabularpardepth added	428
pagesel: Added	865	amsthm: Adapted to trivlist	
prettyref: Added	877	changes	639
subfigure: Added	926	mdframed: Fixes for SVG math or	
subfig: Fix for subcaption end tag	925	lateximage in title	824
subfig: Fix: Math in subcaptions	922	mdframed: Fixes for footnotes	825
textfit: Added	936	ntheorem: Adapted to trivlist	
titleref: Added	948	changes	849
turnthepage: Added	974	theorem: Adapt to trivlist	
Allows memoir's preloaded		changes	939, 941
packages	210	tabular: Fix: Tabular empty lines	488
Docs: xparse warnings	178	list: Fix: Stack unnesting	419
Docs: Fix for double hyphens	94	v0.47	
Docs: Improved install		\LWR@HTML@caption@begin: Fix:	
instructions	94	Argument passed to	
Docs: Improved MiKTEX install		\LWR@origcaption@begin	508
instructions	90	\LWR@LwarpStart: Fix for SVG math	
Docs: Moved table so doesn't		in \nameref	395
interfere with install docs	89	\LWR@WPcell: Fix: Line wrap at HTML	
File: l warp_mathjax.txt: Allow		hyphen	459
MATHJAX inside tabbing	306	\LWR@createautosec: Fix: Line	
File: l warp_mathjax.txt: Allow		wrap at HTML hyphen	378
MATHJAX inside verse	306	\LWR@domulticolumn: Fix: Line	
Fix: Empty sidetoc	512	wrap at HTML hyphen	470
Improved: Robust \, \ , and		\LWR@floatbegin: Fix: Line wrap at	
\textrellipsis commands	598	HTML hyphen	503
Separate LWR@thisautoidWP for		\LWR@htmclosecomment: Add	
word processor <div>s	505	\mbox to prevent line breaks	345
thebibliography: Patched to		\LWR@newautoidanchor: Fix: Line	
emphasize titles	523	wrap at HTML hyphen	505
minipage: Fix: Improper		\LWR@printopenlist: Fix: Line	
\prevdepth	580, 582, 583	wrap at HTML hyphen	416

\LWR@startref:	Fix: Line wrap at HTML hyphen.	495
\LWR@sublabel:	Fix: Line wrap at HTML hyphen.	494
\LWR@subsingledollar:	Added svg math image baseline adjust and em sizing.	531
	Fix: Line wrap at HTML hyphen.	535
\captionlistentry:	Fix: Line wrap at HTML hyphen.	509
\hypertoc:	Fix: Line wrap at HTML hyphen.	516
\hypertocfloat:	Fix: Line wrap at HTML hyphen.	517
\makebox:	Fix: Line wrap at HTML hyphen.	584
General:	2018/01/30	1
	adjmulticol: Fix: Line wrap at HTML hyphen.	623
	blowup: Added.	656
	caption: Added.	664
	caption: Also loads \warp-caption.	240
	changepage: Fix for pagecheck macros.	669
	endheads: Added.	713
	epigraph: Fix: Line wrap at HTML hyphen.	715
	hanging: Fix: Line wrap at HTML hyphen.	781
	hang: Fix: Line wrap at HTML hyphen.	779
	keyfloat: Fix for svg math in captions.	797
	midpage: Fix: Line wrap at HTML hyphen.	834
	multirow: Fix: Line wrap at HTML hyphen.	839
	multitoc: Added.	842
	ntheorem: Fix: Line wrap at HTML hyphen.	854
	realscripts: Fix: Line wrap at HTML hyphen.	884
	scrextend: Fix: Line wrap at HTML hyphen.	893
	sectionbreak: Added.	902
	sidenotes: Fix for svg math in captions.	907
	subfig: Fix for svg math in captions.	922
	subfig: Fix: Support \nameref...	922
xurl:	Added.	1010
\warpmk:	<i>pdfcrop</i> : Removed <i>hires</i> <td>308</td>	308
\teximage:	Added css style option.	558
	Fix: Line wrap at HTML hyphen.	563
\center:	Fix: Line wrap at HTML hyphen.	564
\minipage:	Fix: Line wrap at HTML hyphen.	581
\flushleft:	Fix: Line wrap at HTML hyphen.	564
\flushright:	Fix: Line wrap at HTML hyphen.	564
\enumerate:	Fix: Line wrap at HTML hyphen.	420
\itemize:	Fix: Line wrap at HTML hyphen.	420
\LWR@BlockClassWP:	Fix: Line wrap at HTML hyphen.	349
v0.48		
\@C@setcpageref:	Fix for new v0.21 of cleveref.	576
\@C@setcref:	Fix for new v0.21 of cleveref.	575
\@C@setcrefrange:	Fix for new v0.21 of cleveref.	575
\@biblabel:	Improved bibliography label.	522
\@item:	Honors \makelabel.	416
\@maketitle:	Fix: Errors with IEEEtran class.	406
\LWR@LwarpStart:	Adjusted space around captions.	394
\LWR@ProvidesPackageDrop:	Fix: Options with braces.	240
\LWR@addtabularrulecolor:	colortbl: Added.	460
\LWR@addtabularrulecolors:	colortbl: Added.	461
\LWR@closetabledatacell:	colortbl: Added.	431
\LWR@includegraphicsb:	Fix: Virtual page size limited to a group.	769, 773
\LWR@lookforpackagename:	Fix: Parsing similar package names.	237
\LWR@newautopagelabel:	Fix: TOC, LOF, LOT links.	371

\LWR@newhtmlfile: Fix: TOC, LOF, LOT links.	375
\LWR@nullfonts: Fix: \newline in title.	592
\LWR@parsedrequirepackagenames: Fix: Parsing similar package names.	236
\LWR@parsestablecols: Fix: Ignore optional tabular column arguments.	446
\LWR@restoreorigformatting: Fix: Spacing in SVG math, lateximage, Tikz.	524
\LWR@section: Fix: TOC, LOF, LOT links.	385
\LWR@tableddatasinglecolumntag: colortbl: Added.	453
\LWR@textcurrentfont: Added. Improves font control.	594
\bibliography: Fix: \BaseJobname for bibliography.	522
\centerline: Added.	565
\l@part: Adapts to classes without \part.	518
\leftline: Added.	565
\mbox: Nullified for HTML.	584
\rightline: Added.	565
\thempfootnote: Removed \itshape.	365
General: 2018/02/14	1
acronym: Added.	621
acro: Added.	619
chapterbib: Added.	670
colortbl: Added.	447, 460, 700
fancyref: Now directly supported.	729
hypcap: Added.	781
hypernat: Added.	782
hyperref: \texorpdfstring now uses the TEX string.	786
luatodonotes: Improved \todototoc.	815
siunitx: Changes fraction to symbol.	912
siunitx: Improved svg math.	910, 912
siunitx: Improved color output.	910
stfloats: Added.	921
todonotes: Improved \todototoc.	969
vmargin: Added.	983
xfrac: Fix: Added groups around super/subscripts to localize LWR@nestspan changes.	1003
Docs: Converting an existing document.	112
Improved font control.	594, 595
tabular: colortbl: Added.	486
lateximage: Print mode boxed to natural width.	564
abstract: Allow optional name.	408
v0.49	
\LWR@addtabularcellcolor: xcolor: Added tabular row colors.	463
\LWR@domulticolumn: xcolor: Added tabular row colors.	471
\LWR@printlength: Fix: Group printlen changes.	235
\affiliation: Fix: Adapts to classes which already provide.	400
\href: Fix: Adapt to classes.	499
\noalign: Fix: \noalign inside tabular.	483
\url: Fix: Adapt to classes.	500
General: 2018/02/19	1
amsmath: Fix: Patches for \eqref.	630
eso-pic: Fix for \AddToShipoutPicture.	721
figsize: Added.	737
fnlineno: Added.	751
hypdestopt: Added.	782
hyphenat: Added.	788
lineno: Added.	802
luacolor: Added.	815
pagegrid: Added.	864
pdfrender: Added.	872
resizegather: Added.	888
vertbars: Added.	982
vwcol: Added.	984
xcolor: Added tabular row colors.	447, 999
Fix: Adapt to classes.	599
v0.50	
\@ensuredmath: Fix: Use lateximage even if MathJax.	539
Improved SVG math alt tags.	539
\LWR@atbeginverbatim: Improved column alignment.	412
\LWR@doequation: Improved SVG math display.	544

\LWR@doubledollar: Improved svg	
math alt tags.	538
Improved SVG math display.	538
\LWR@footnotetext: Robustify	
macros.	363
\LWR@htmrefsectionfilename:	
Fix: SVG math in a section name.	336
\LWR@newhtmlfile: Fix: SVG math	
in a section name.	374
\LWR@nullfonts: Fix: \underline	
in sectioning file name.	593
Robustify macros.	593
\LWR@overline: Added.	597
\LWR@subsingledollar: Fix: Use	
lateximage even if MathJax.	531
Improved SVG math alt tags.	531
MD5 hash avoids duplicate svg	
math.	536
\LWR@vspace: Robustify macros.	603
\newline: Robustify macros.	600
\textsubscript: Robustify macros.	596
\textsuperscript: Robustify	
macros.	596
General: 2018/03/03	1
l warp.css: Improved svg display	
math centering.	264
l warp_one_limage.txt: Added.	305
amsmath: Fix: Upright tags for	
svgmath.	630
axodraw2: Added.	652
bytefield: Added.	663
dblfloatfix: Added.	705
diagbox: Added.	706
epstopdf: Added.	717
listings: Force flexible columns.	805
morefloats: Added.	835
nonfloat: Added.	848
ntheorem: Fix: Not standard nor	
amsthm selected.	857
pbox: Added.	867
phfqt: Added.	874
schemata: Added.	892
siunitx: Fix: Loads xcolor.	910
siunitx: Improved svg math alt	
tags.	912
siunitx: Improved units.	567, 910, 913
xy: Added.	1010
l warpmk: Error if	
lateximages.txt does not	
exist.	308
l warpmk: Error if l warpmk.conf	
points to l warp.	308
l warpmk: Improved error	
messages.	308
l warpmk: MD5 hash avoids	
duplicate svg math.	308
l warpmk: Multiprocess support	
making lateximages.	308
AMS environments: Improved svg	
math display.	630
Fix: Load fontspec if necessary.	231
Robustify macros.	595
lateximage: Fix: SVG math in a	
section name.	561
MD5 hash avoids duplicate svg	
math.	560, 563
eqnarray: Improved svg math	
display.	551
v0.51	
\@ensuredmath: Hashes	
\ensuremath.	539
\@item: Restored list label space.	418
\LWR@HTMLsanitize: Fix: Escapes	
double quotes.	554
\LWR@HTMLsanitizeexpand: Fix:	
Escapes double quotes.	555
\LWR@LwarpStart: MathJax: Nullifies	
\ensuremath.	396
\LWR@addbaselinemarker:	
Improved svg math baseline.	530
\LWR@atbeginverbatim: Adds	
vertical offset.	411
\LWR@customizeMathJax: MathJax:	
Nullifies \ensuremath.	371
\LWR@doequation: Fix:	
\addcontentsline inside svg	
math. Provides an autoid anchor.	544
\LWR@doubledollar: Fix:	
\addcontentsline inside svg	
math. Provides an autoid anchor.	538
\LWR@findcurrenttextcolor:	
Added	
\LWR@findcurrenttextcolor	
when no xcolor.	597
\LWR@newautoidanchor: Fix: No	
autoid is inside a lateximage.	505
\LWR@newhtmlfile: MathJax:	
Nullifies \ensuremath.	376
\LWR@subsingledollar: Fix:	
\ensuredmath inside svg image.	532

Fix: <code>\teximage</code> inside $\mathcal{M}\mathcal{S}$	
<code>\text</code>	533
Fix: Honors text font around <code>svg</code>	
<code>math</code>	533
Fix: SVG math with enclosed	
<code>\teximage</code>	532
Improved SVG math baseline.	534
SVG math baseline improved with	
invisible rule at corner.	536
Typeset <code>svg</code> math only once	
during measurement.	533
<code>\LWR@textcurrentcolor</code> : <code>xcolor</code> :	
<code>\LWR@textcurrentcolor</code> if	
<code>xcolor</code> not loaded.	598
<code>\addcontentsline</code> : Add missing	
support for float mechanism if	
necessary.	509
No anchor ID if inside <code>svg</code> image.	509
<code>\displaymathnormal</code> : Processing for	
complicated display math.	547
<code>\displaymathother</code> : Processing for	
complicated display math.	547
<code>\textcolor</code> : Fix: SVG math color.	994
General: 2018/03/24	1
HTML entity used for text dollar.	530
<code>\warp_one_limage.txt</code> :	
<code>pdftocairo -noshrink</code> added.	305
afterpackage: No longer required.	233
chemfig: Added.	671
chemformula: Added.	672
chemgreek: Added.	677
chemmacros: Added.	677
chemnum: Added.	697
epstopdf-base: Added.	718
fancybox: Fix: Optional tag for	
<code>\item</code> in a span.	727
grid: Added.	777
listings: Forces cleared options.	806
ltxgrid: Added.	813
mhchem: Added.	831
tikz: Fix for <code>\tikz</code> macro.	944
tikz: Fix for <code>\tikz</code> with optional	
argument.	944
titling: Fix for <code>\thanks</code> mark.	955
<code>\warpmk</code> : <code>pdfcrop</code> : Restored hires	
option.	308
<code>\warpmk</code> : <code>pdftocairo -noshrink</code>	
added.	308
AMS environments: Fix:	
<code>\addcontentsline</code> inside <code>svg</code>	
<code>math</code> . Provides an autoid anchor.	630
Docs: tikz limitations.	165
Docs: Multiple authors and	
affiliations.	142
Docs: Things to avoid.	132
Docs: Updated Converting an	
existing document.	112
Fix: Remember original <code>\#</code> in case	
is redefined.	252
<code>\teximage</code> : Added additional	
hashing option.	558
Fix: <code>\teximage</code> inside $\mathcal{M}\mathcal{S}$	
<code>\text</code>	559
Processing for complicated display	
<code>math</code>	561
<code>\alignat</code> : Fix: Added.	635
<code>\eqnarray</code> : Fix: <code>\addcontentsline</code>	
inside <code>svg</code> math. Provides an	
autoid anchor.	551
<code>\LWR@displaymathother</code> : Processing	
for complicated display math.	540
<code>\LWR@equationother</code> : Processing for	
complicated display math.	541
v0.52	
<code>\ensuredmath</code> : Improved hashing	
expansion.	539
<code>\mpfootnotetext</code> : Fix: Paragraph	
handling.	364
<code>\CustomizeMathJax</code> : Added.	371
<code>\LWR@addbaselinemarker</code> :	
Warnings if	
<code>\warp_baseline_marker.png</code> is	
not present or if <code>graphicx</code> s not	
loaded.	530
<code>\LWR@customizedMathJax</code> : Added.	371
<code>\LWR@doequation</code> : Fix: <code>equation*</code>	
now based on <code>equation</code> instead	
of <code>displaymath</code>	544
Fix: <code>equation*</code> with <code>split</code>	544
<code>\LWR@filenamenoblanks</code> : Fix:	
<code>\FileDepth</code> with non-utf8	
encoding.	370
<code>\LWR@footnotetext</code> : Fix: Paragraph	
handling.	363
<code>\LWR@nullfonts</code> : Fix:	
<code>\texorpdfstring</code> in section	
names.	594

\LWR@section: Fix: Footnote	
numbering: Limited HTML	
comment if starred.	382
Fix: Footnote numbering: Use	
short toc entry for HTMLDebug	
comments.	382
\LWR@subsingledollar: Added	
user-adjustable svg math font	
scaling.	534
\LateximageFontSize: Added	
user-adjustable svg math font	
scaling.	554
\href: Fix: #, %, &, ~, _ in URL.	499
\nolinkurl: Fix: #, %, &, ~, _ in URL.	499
\theHTMLTitleSeparator: Fix:	
\FileDepth with non-utf8	
encoding.	389
\url: Fix: #, %, &, ~, _ in URL.	500
General: 2018/04/01	1
breakurl: Fix: #, %, &, ~, _ in URL.	660
endfloat: Updated for v2.6.	712
fancyvrb: Initial support for	
\VerbatimFootnotes.	724, 731
hyperref: Fix: #, %, &, ~, _ in	
URL.	782, 783, 785
nicefrac: Added.	847
url: Added.	979
\lwarpmk: Fix: Memory overflow	
when spawning tasks.	308
\lwarpmk: Fix: Skip image	
generation if from page 0.	308
Changed FootnoteDepth default	
to \subsubsection.	361
Docs: Improved install	
instructions.	91, 92
Fix: MathJax script line wraps.	
Reduced right margin.	232
If pdfLaTeX, allow other input	
encoding.	219
pkggraphics: Added defaults.	766
pkggraphicx: Updated for v1.1a.	767
pkggraphicx: Updated for v1.1b.	767
Restore \kill in a lateximage.	812
tabbing: Fix to allow inside	
lateximage.	414
lateximage: Fix for hash expansion.	560
v0.53	
General: 2018/04/01	1
\lwarpmk: Added	
\lwarpmk cleanimages.	308
v0.54	
\LWR@afterendverbatim: Added	
vspace argument.	413
\LWR@atbeginverbatim: Improved	
column alignment.	412
\LWR@earlyloadnever: Added.	209
\LWR@endfloatalignment: Honor	
\centering, etc. in floats.	506
\LWR@floatalignment: Honor	
\centering, etc. in floats.	506
\LWR@floatbegin: Honor	
\centering, etc. in floats.	504
\LWR@floatend: Honor \centering,	
etc. in floats.	504
\LateximageFontSizeName:	
Defaults to normalsize.	554
\centering: Added debug	
comment.	565
\raggedleft: Added debug	
comment.	565
\raggedright: Added debug	
comment.	565
General: 2018/04/22	1
*.lwarpmkconf: Option	
IndexLanguage changed to	
xindyLanguage.	264
*.lwarpmkconf: Option	
pdftotextEnc added.	264
*.lwarpmkconf: Option	
xdyFilename changed to	
xindyStyle.	264
*.lwarpmkconf: Option	
xindyCodepage added.	264
lwarpm.css: Fix:	
Text-decoration-skip: auto.	264
lwarpmk.conf: Option	
IndexLanguage changed to	
xindyLanguage.	264
lwarpmk.conf: Option	
pdftotextEnc added.	264
lwarpmk.conf: Option	
xdyFilename changed to	
xindyStyle.	264
lwarpmk.conf: Option	
xindyCodepage added.	264
bibunits: Added.	653

chngpage: Added.	669
forest: Added.	755
glossaries: Fix when not using babel or polyglossia.	762
gridset: Added.	778
hyperref: Fix: \hyperref and \hyperlink with special chars in text.	784, 785
hyperref: Fix: \ref in \hyperref and \hyperlink caused nested link.	784, 785
\warp-patch-memoir: Update for v3.7g.	1020
magaz: Added.	817
ragged2e: Fix: \centering, etc. .	883
textcomp: Fix for \textperthousand.	933
tikz: Fixes for \pgfpicture, minipages, fit, align, font. . . .	944
\warpmk: Added pdftotextenc. .	308
\warpmk: Added xindycodepage. .	308
\warpmk: Changed language to xindylanguage.	308
\warpmk: Changed xdyfile to xindystyle.	308
\warpmk: Improved error if configuration file does not exist.	308
\warpmk: Increased prominence for error for an unknown command.	308
\warpmk: Verifies HTML version exists before \warpmk limages. .	308
\warpmk: Verifies image references before \warpmk limages. . . .	308
Add: pdftotextEnc.	226
Add: xindyCodepage.	226
Added early check for disallowed packages.	210
Docs: BibTeX.	144
Docs: Macros in sectioning names.	132
Never load aecompl.	210
Option IndexLanguage changed to xindyLanguage.	226
Option xdyFilename changed to xindyStyle.	226
verse: Fix: Line spacing.	410
v0.55	
\WWR@LwarpStart: Fix: Overfull boxes in \teximage.	394
\WWR@floatbegin: Fix: Float optional args.	504
\WWR@nullfonts: Removed extraneous space which appeared in file links.	594
\phantomsection: Fix: \ForceHTMLTOC with \phantomsection.	607
General: 2018/04/26	1
clrdblpg: Added.	700
Fix: \centering, etc. for koma-script.	504
Fix: QED symbols in \teximage.	640, 861
v0.56	
\WWR@addcdashline: arydshln: Added.	459
\WWR@addmulticolverrulecolor: Adds support for dashed vertical rules.	468
Adds support for double vertical rules.	468
\WWR@addtabularhrulecolor: Adds support for arydshln dashed rules.	460
Adds support for double \hlines and \midrules.	460
\WWR@addtabularrulecolors: Adds support for dashed vertical rules. .	461
Adds support for double vertical rules.	461
\WWR@closeparagraph: Added support for parnotes.	354
\WWR@domulticolumn: Adds support for dashed vertical rules. . . .	470
Adds support for double vertical rules.	470
\WWR@floatbegin: Adds a <class> per float package style.	503
\WWR@multicolpartext: Fix: \multicolumn parameters. . .	465
\WWR@openparagraph: Added support for parnotes.	352
\WWR@parsebarcolumn: Adds support for double vertical rules. .	439
\WWR@parsecoloncolumn: arydshln: Added.	440
\WWR@parsesemicoloncolumn: arydshln: Added.	440

\LWR@parsetablecols: Added array	
W column.	446
\LWR@printmccoldata: Added array	
W column.	467
Fix: \multicolumn parameters. .	467
\LWR@printmccoltype: Added array	
W column.	465
Adds support for dashed vertical	
rules.	465
Adds support for double vertical	
rules.	465
\LWR@tabledatacolumntag: Fix:	
\morecmidrules	481
\LWR@textcurrentfont: Added	
span.textbf, etc.	594
General: 2018/05/12	1
*.lwarpmkconf: Records	
--shell-escape.	264
lwarpm.css: Added div.textbf,	
etc.	264
lwarpm.css: Added span.textbf,	
etc.	264
lwarpmk.conf: Records	
--shell-escape.	264
arydshln: Added.	426, 643
lua-check-hyphen: Added.	814
paralist: Fixes for compactenum,	
compactitem, compactdesc. .	865
parnotes: Added.	866
quoting: Added.	883
tocenter: Added.	960
underscore: Added.	977
lwarpmk: Added	
lwarpmk pdftosvg.	308
lwarpmk: Supports	
--shell-escape.	308
Added \thinspace.	598
Docs: lwarpmk pdftosvg.	110
LWR@blocktextcurrentfont: Added	
div.textbf, etc.	594
v0.57	
\BlockClassSingle: Improved	
print/HTML output selection. .	348
\InlineClass: Improved	
print/HTML output selection. .	349
\LWR@customizeMathJax: MathJax:	
Supports \footnote,	
\footnotemark.	371
\LWR@ref@ignorestar: subcaption:	
Fix: \subref.	497
\LWR@subhyperref: Fix: Text	
catcodes.	498
\LWR@subhyperreftext: Fix: Text	
catcodes.	498
\LWR@subsingledollar: Fix:	
Dynamic inline math	
expressions.	532, 533, 536
\LWR@vspace: Improved print/HTML	
output selection.	603
\StartDefiningMath: Added.	331
\boxframe: xcolor: Fix: Colored	
\boxframe.	999
\colorbox: xcolor: New system for	
switching print and HTML	
outputs.	995
\colorboxBlock: xcolor: New	
system for switching print and	
HTML outputs.	995
\fboxBlock: Improved print/HTML	
output selection.	586
\fcolorbox: xcolor: New system for	
switching print and HTML	
outputs.	996
\framebox: Improved print/HTML	
output selection.	585
\href: Fix: Text catcodes.	
....	499
\inlinemathother: Added.	332
\listof: Fix: Provide \l@name if not	
defined.	514
\makebox: Improved print/HTML	
output selection.	584
\mathimagename: Added.	529
\mbox: Improved print/HTML output	
selection.	584
\multicolumnrow: multirow:	
Improved print/HTML output	
selection.	841
Improved print/HTML output	
selection.	476
\newfloat: rotfloat: Added float	
styles.	891
rotfloat: Fix for listof sideways	
floats.	891
\packagediagramname: Added.	529
\parbox: Improved print/HTML	
output selection.	583
\raisebox: Improved print/HTML	
output selection.	588
\reflectbox: Improved print/HTML	
output selection.	776

\resizebox: Improved print/HTML output selection	776	New system for switching print and HTML outputs	247
\rotatebox: Improved print/HTML output selection	774	minipage: Improved print/HTML output selection	580
\rule: Fix: Colored rules	605	BlockClass: Improved print/HTML output selection	348
\scalebox: Improved print/HTML output selection	775	fminipage: Improved print/HTML output selection	586
\textcolor: xcolor: New system for switching print and HTML outputs	994	LWR@BlockClassWP: Improved print/HTML output selection	349
General: 2018/06/06	1	v0.58	
lwrap.css: Added ruled, boxed, boxruled floats	264	\LWR@HTML@caption@begin: Improved print/HTML output selection	508
lwrap.css: Increased float vertical margins	264	\LWR@HTML@caption@end: Improved print/HTML output selection	508
algorithm2e: Added	624	\LWR@HTML@ref: Improved print/HTML output selection	496
bigdelim: Improved print/HTML output selection	654	\LWR@doindexentry: Adds support for \see, \seealso, \emph, \textbf, etc	521
breakurl: Fix: Text catcodes	660	\lwrhyperindexrefnullified: Adds support for \see, \seealso, \emph, \textbf, etc	521
colortbl: New system for switching print and HTML outputs	701, 702	\LWR@indexitem: Accepts optional arg for repeatindex	520
ellipsis: Added		\dotfill: Improved print/HTML output selection	599
\midwordellipsis	711	\hfill: Improved print/HTML output selection	599
errata: Added	719	\hrulefill: Improved print/HTML output selection	599
float: Added float styles	741	\hyperindexref: Adds support for \see, \seealso, \emph, \textbf, etc	521
float: Fix: Do not pre-define \l@name	741	\printindex: Fix: Extra \newpage to flush pending \index writes	818
ltablex: Added	813	General: 2018/07/07	1
marginnote: Fix: Long optional argument	819	*.lwrapmkconf: Added option makeindexstyle	264
multirow: Improved print/HTML output selection	839	*.lwrapmkconf: Added options makeindex and xindy	264
register: Added	885	*.lwrapmkconf: Generated \AtBeginDocument	264
subcaption: Fix: \subref	783	lwrap.xdy: Requires makeindex.xdy	305
trimclip: Added	972	lwrap.xdy: Supports bold, italic	305
vowel: Added	983	lwrap_html.ist: Added	304
xellipsis: Added	1001	lwrapmk.conf: Added option makeindexstyle	264
xfrac: Improved print/HTML \scalebox control	1003		
xltabular: Added	1005		
xpiano: Added	1006		
lwrapmk: Improved code factoring	308		
lwrapmk: Improved error handling	308		
Docs: Recompiling lwrapmk or css files	195		
Docs: Recreating the index for lwrap source	192		

<i>lwarpmk.conf</i> : Added options makeindex and xindy.	264	Replaced each \csuse with \nameuse to force error if undefined.	1
<i>lwarpmk.conf</i> : Generated \AtBeginDocument.	264	tabbing: Improved print/HTML output selection.	414
array: Improved print/HTML output selection.	643	v0.59	
attachfile2: Added.	648	\LWR@addbaselinemarker: Uses .eps if DVI <i>latex</i>	530
attachfile: Added.	647	\LWR@includegraphicsb: Fix: Expand filename.	770
cases: Added.	668	Now works with .pdf and .eps filename extensions.	769
imakeidx: Added.	791	\LWR@latexmkcmb: Fix: --shell-escape with <i>latexmk</i> . .	258
index: Added.	795	\LWR@writeconf: Compilation commands now preassigned by <i>l warp</i> instead of being computed by <i>lwarpmk</i>	263
intopdf: Added.	797	\[: Fix with \displaymathnormal. .	539
<i>lwarpmk-patch-komascript</i> : Modified indexing.	1013	General: 2018/09/07	1
<i>lwarpmk-patch-memoir</i> : Fix for \specialindex.	1038	Slunits: Added.	909
<i>lwarpmk-patch-memoir</i> : Fix for multiple indexes.	1039	accsupp: Added.	619
makeidx: Added. Moved from <i>l warp</i> core.	818	amsmath: Moved from the <i>l warp</i> core.	629
memoir: Fix for \firsthline, \lasthline.	478	asymptote: Added.	646
memoir: Fix for booktabs.	483	axessibility: Added.	651
pdfpages: Added.	869	breqn: Added.	661
pdfx: Added.	873	bxpapersize: Added.	662
repeatingindex: Added.	888	canoniclayout: Added.	664
splitidx: Added.	918	chemformula: Fix for \NMR.	694
textcomp: Improved print/HTML output selection.	933	draftcopy: Added.	708
<i>lwarpmk</i> : Added makeindex and xindy options.	308	epstopdf-base: Improved.	718
<i>lwarpmk</i> : Added -p option for project name.	308	epstopdf: Improved.	717
<i>lwarpmk</i> : Added optional list of names for <i>lwarpmk printindex</i> and /cmdslwarpmk htmlindex. .	308	fnbreak: Added.	750
<i>lwarpmk</i> : Glossary generation now uses <i>makeglossaries</i>	308	nccfancyhdr: Added.	845
<i>lwarpmk</i> : <i>lwarpmk clean</i> removes all *.ind and *.idx files.	308	pdftricks: Added.	873
Added makeindex option.	227	pst-eps: Added.	879
Added xindy option.	228	pstricks: Added.	880
Added option makeindexStyle. .	225	units: Added support for MathJax. .	977
Docs: Index, <i>makeindex</i> , imakeidx.	146	xunicode: Added.	1009
Docs: Misplaced \omit.	197	<i>lwarpmk</i> : Added <i>lwarpmk epstopdf</i>	308
Fix: memoir and ccaption.	210	<i>lwarpmk</i> : Consolidated compiling options into printlatexcmd and HTMLLatexcmd.	308
Improved print/HTML output selection.	598	<i>lwarpmk</i> : Double instead of single-dashed --shell-escape option.	308
		<i>lwarpmk</i> : Error if <i>lwarpmk.conf</i> format changed.	308

<i>lwarpmk</i> : Warning if operating system changed.	308
Added option dvipdfmx.	228
Added option dvipdfm.	228
Added option dvips.	228
Docs: <i>lwarpmk estopdf</i> .	110
File: <i>l warp_mathjax.txt</i> : Fix:	
Removed chapter number from tagged non-numeric MathJax equations.	306
File: <i>l warp_mathjax.txt</i> :	
Updated to MathJax v2.7.4.	306
picture: Added an alt tag.	578
v0.60	
\LWR@clearmidrules: tabular: Fix for midrules.	454
\LWR@parsonormalcolumn: tabular: Improved memory management: Not using xstring.	441
\LWR@tabledatausingeolumntag: tabular: Improved memory management: Not using xstring.	452, 453
\LWR@tabularendofline: Fix: Slowdown for long tables.	436
General: 2018/09/19	1
tabular: Improved memory management: Global boolean.	427
tabular: Improved memory management: Not using xstring.	429, 839
2up: Added.	615
booklet: Added.	656
bophook: Added.	659
diagbox: Fix for par tags.	707
draftfigure: Added.	708
fancytabs: Added.	730
fullminipage: Added.	758
grid-system: Added.	778
layaureo: Added.	799
leading: Added.	800
listings: Fix for HTML entities.	805
listings: Fix if inside a list.	806, 809
thumbs: Added.	943
thumb: Added.	943
widows-and-orphans: Added.	987
v0.61	
\DeclareGraphicsExtensions: Fix: EPS for DVI LATEX.	764
\LWR@HTMLLatexCmd: Added HTMLLatexCmd option.	263
Added PrintLatexCmd option.	263
\LWR@addcompilecmd: Removed spaces.	258
\LWR@closetabledatacell: Fix: Par tags in tabular.	430
\LWR@hyperindexrefnullified: Made robust.	521
\LWR@includegraphicsb: Fix: EPS for DVI LATEX.	770
Set keys before using filename, for epsfig.	770
\LWRopseq: Added spaces.	223
\RequirePackage: Support up to 20 packages.	238
\inlinemathnormal: Changed name from \StopDynamicMath to \inlinemathnormal.	332
\inlinemathother: Changed name from \StartDynamicMath to \inlinemathother.	332
\listof: Fix: newfloat lists.	514
\l warpsetup: Added.	224
General: 2018/10/13	1
l warp.css: Footnotes text align left.	264
l warp.css: Minipage table and footnotes: tighter margin.	264
chkfloat: Added.	698
cmdtrack: Added.	700
copyrightbox: Added.	703
dprogress: Added.	708
epsfig: Added.	717
lua-visual-debug: Added.	814
pdfprivacy: Added.	872
psfragx: Added.	878
psfrag: Added.	878
pstool: Added.	879
refcheck: Added.	885
srcltx: Added.	919
srctex: Added.	920
supertabular: Fix for caption w/o opt arg.	928
thinsp: Added.	941
threadcol: Added.	942
uspace: Added.	979
vpe: Added.	984
xbmks: Added.	989
xtab: Fix for caption w/o opt arg.	1008

Added HTMLLatexCmd option.	227
Added PrintLatexCmd option.	227
Docs: \tracingl warp	243
Docs: HTML entities.	133
Docs: Compiling using custom shell commands.	179
Docs: Fonts.	113
Docs:	
HTMLDebugComments	120, 243
Docs: Multiple indexes.	202
Don't write configuration files if processing pstool image.	256
Spaces redefined	
\AtBeginDocument	598
v0.62	
\@partcntformat: Added for ctex.	380
\@partnameformat: Added for ctex.	380
\InlineClass: Added optional word-processing style. Replaces \LWR@HTMLtextstyle.	349
\LWR@PreloadedPackage: Added.	566
\LWR@ProvidesPackagePass: Fix: Unknown option error.	239
\LWR@endofline: Extra space if optional arg.	600
\LWR@filestart: Refactored.	392
\LWR@includegraphicsb: Fix:	
FormatWP.	770
Fix: Filename expansion.	771
\LWR@isolate: Added.	221
\LWR@textcurrentfont: Added print version.	598
Tracks depth to avoid nesting repeated font changes.	594
\colorboxBlock: Fix: Horiz white space.	996
\fcolorbox: Fix: No longer requires xifthen.	571
\fcolorboxBlock: Fix: Horiz white space.	997
\l@chapter: Don't define if no \chapter.	518
\slshape: Added.	596
\textup: Fixed WP span class.	590
\theHTMLSection: Added.	390
\theHTMLTitleSection: Added.	390
\theHTMLTitleSeparator: Refactored.	389
General: 2018/11/19	1
\textbf and related:	
Improved font detection.	589
\l warp.css: Added css for xfrac, nicefrac.	264
\l warp.css: Fixed css for \textup.	264
\l warp.css: Reduced margins in titlepage.	264
\l warp_formal.css: Fix: Font for verse.	299
2in1: Added.	615
CJKutf8: Prevented unless xeCJK.	699
CJK: Prevented unless xeCJK.	699
asymptote: Improved alt tags.	646
bitpattern: Added.	655
calc: Fix: Required for print version.	233
chngpage: Fix: Loads \l warp-chngpage.	669
ctexpatch: Added patch.	613
flippdf: Added.	740
musicography: Added.	842
nicefrac: Improved font control and css, honors nice, ugly.	847
notespages: Added.	848
octave: Added.	862
pdfcomment: Added.	868
pdfmarginpar: Added.	868
register: Updated to v1.8.	885
rviewport: Added.	891
semantic-markup: Added.	903
textcomp: Fix conflict with xunicode.	935
tram: Added.	971
twoup: Added.	974
ulem: Improved compatibility with CJKulem.	975
ulem: Now works in a \lateimage.	975
unitsdef: Added.	978
units: Improved font control and css, honors loose, tight.	977
xchangebar: Added.	1001
xfrac: Improved css.	1003
xunicode: Fix conflict with textcomp.	1009
Added early checks for CJK, CJKutf8.	210
Docs: asymptote.	167
Docs: miktex-poppler-bin-*	94

Docs: <i>MiKTeX Console</i>	90	l warp.css: Added css for vertical writing	264
Docs: Improved MiKTeX install instructions	90	l warp.css: Improved css for mdframed	264
Docs: UTF-8 locale	182	amsthm, mdframed: Fix for enforced load order	636
File: <i>l warp_mathjax.txt</i> :		emumitem: Nullify <i>\DrawEnumitemLabel</i>	714
Removed inoperable siunitx extension	306	everyhook: Fix for <i>bxjs*</i> classes	233
Fix for <i>\em</i>	592	geometry: Fix for <i>bxjs*</i> classes	232
Fix: Horiz white space	998	mdframed: Avoid thin rules	822
Logos: CSS instead of <i><sup></i> , <i><sub></i>	607	mdframed: Improved font control	826–828
Logos: Fix for X _E T _E X logo if graphics is not loaded	610	stfloats: Adapted to <i>ltj*</i> classes	921
Logos: Improved CSS	607	xpinyin: Added	1007
Logos: Made robust	607	zhlineskip: Added	1011
fcolorminipage: Fix: Horiz white space	998	Added pTEXsupport:	206
Fix: No longer requires <i>xifthen</i>	573	Docs: \linkhomename:	122
fminipage: Fix: Horiz white space	588	Docs: \sidetocname:	122
LWR@blocktextcurrentfont: Added print version	598	Fix: Default \LWR@mdfive:	218
v0.63		Improved titles:	825, 826
\LWR@HTMLLatexCmd: <i>ujarticle</i> and related: Compile options	262	pTEX: Encoding	219
\LWR@LwarpStart: Fixes for <i>xeCJK</i>	393	pTEX: Load upquote	220
\LWR@atbeginverbatim: Fix for <i>xeCJK</i>	411	pTEX: No newunicodechar	220
\LWR@checkloadbefore: Added	208	\LWR@BlockClassWP: Fix for <i>xeCJK</i>	349
\LWR@checkloadfilename: Added to reduce number of <i>l warp-*</i> files	236	v0.64	
\LWR@checkloadnever: Added	209	\LWR@HTMLLatexCmd: <i>utarticle</i> and related: Added	262
\LWR@compileuplatex: Added	259	\LWR@checkloadfilename: Prevented bitfield, doublespace, newthm, rplain, si	236
\LWR@createautosec: Fix for <i>xeCJK</i>	378	\LWR@section: Support for <i>ujarticle</i> and related	383
\LWR@earlyclassloadnever: Added	209	\enskip: Made robust	601
\LWR@filestart: Fix: Break ligature for <i>luatexko</i>	391	\qquad: Made robust	601
\LWR@htmclosecomment: Fix: Break ligature for <i>luatexko</i>	345	\quad: Made robust	601
\LWR@isolate: Fix for <i>xeCJK</i>	221	\theHTMLTitleSeparator: Added <i>utarticle</i> and related	389
\LWR@notltjloadafter: Added more classes	207	General: 2018/12/08	1
Added	207	<addlines>: Updated to v0.3</addlines>	623
\LWR@subhtmlelementclass: Fix for <i>xeCJK</i>	346	biblatex: Added patch for C _E T _E X	653
\LinkHome: Fix: Print version	337	bsheaders: Added	662
\linkhomename: Added	337	gmeometric: Added	763
General: 2018/12/03	1	marginal: Added	818

Added utarticle and related.	612
v0.65	
\LWR@LwarpEnd: Improved css for page layout.	396
\LWR@LwarpStart: Improved css for page layout.	395
\LWR@PreloadedPackage:	
\AtBeginDocument to avoid option clashes.	566
\LWR@hyperindexrefnullified:	
Added \textsi.	521
\LWR@includegraphicsb: graphics:	
Added \includegraphics alt key.	770, 772
\LWR@newhtmlfile: Error if duplicate file name.	373
Improved css for page layout. 372, 375	
\LWR@nullfonts: Added \sishape. 593	
Added \textsi.	592
\LWR@restoreorigformatting:	
Added \sishape.	524
Fix: tabular*.	525
\enskip: Changed to Unicode EN SPACE.	601
\quad: Changed to Unicode EM SPACE.	601
\sishape: Added \sishape.	595
\textsi: Added.	591
General: 2018/12/22	1
lwarp.css: Added \sishape, \textsi.	264
lwarp.css: Improved css for page layout.	264
lwarp.css: Improved css for quotations.	264
lwarp.css: Siderocto left for improved \marginpars.	264
l warp_formal.css: Siderocto left for improved \marginpars.	299
l warp_sagebrush.css: Siderocto left for improved \marginpars. 294	
bounddvi: Added.	659
embrac: Added.	711
footnoterange: Added.	754
gentombow: Added.	760
geometry: Fix for bxjs* classes.	232
graphics: Added	
\includegraphics alt key.	570, 765, 766, 768
lltjext: Added.	809
multicolrule: Added.	838
multicol: Added \docolaction.	838
plardyshln: Added.	875
plexarydshln: Added.	876
plextcolortbl: Added.	876
plex: Added.	875
pxatbegshi: Added.	880
pxeveryshi: Added.	881
pxfnright: Added.	881
pxjahyper: Added.	881
tascmac: Added.	931
versionotes: Added.	982
Added \sishape.	253
Added early checks for jarticle, tarticle, and related.	210
Fix for \rensuji.	612
Fix space between class and id.	350
tabular: Added support for plect.	485
Fix: tabular*.	485
Fix: Rule color.	486
minipage: Refactored to later allow Japanese <t/y> argument.	580
LWR@figcaption: Uses <figurecaption> instead of <figcaption>.	507

Index of Objects

This is an index of macros, environments, booleans, counters, lengths, packages, classes, options, keys, files, and various other programming objects. Each is listed by itself, and also by category. In some cases, they are further subdivided by [class].

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition.

Symbols	
\\$	<i>530</i>
\&	<u>334</u> , <i>6822</i>
\(.	<u>9218</u>
\,	<u>133</u>
--shell-escape (option)	<u>113</u>
\@@setcpageref	<u>9993</u>
\@@setcref	<u>9956</u>
\@@setcrefrange	<u>9970</u>
\@author	<u>398</u>
\@begintheorem	<u>6488</u>
\@biblabel	<u>8888</u>
\@capttype	<u>8514</u>
\@chapcntformat	<u>5697</u>
\@currentHref	<u>8362</u>
\@currentlabelname	<u>8218</u>
\@date	<u>398</u>
\@dلبfloat	<u>8480</u>
\@donoparitem	<u>6514</u>
\@endtheorem	<u>6500</u>
\@ensuredmath	<u>9228</u>
\@float	<u>8480</u>
\@fnsymbol	<u>6277</u>
\@footnotetext	<u>5327</u>
\@include	<u>960</u>
\@item	<u>6527</u>
\@makecaption	<u>8535</u>
\@makefnmark	<u>5295</u>
\@makefntext	<u>5294</u>
\@maketitle	<u>56</u> , <u>6300</u>
\@mklab	<u>6508</u>
\@mpfootnotetext	<u>5329</u>
\@nbitem	<u>6594</u>
\@opargbegintheorem	<u>6494</u>
\@partcntformat	<u>5698</u>
\@partnameformat	<u>5699</u>
\@rowc@lors	<u>7143</u>
\@rowcolors	<u>7142</u>
\@seccntformat	<u>5695</u>
\@starttoc	<u>8648</u>
\@textsubscript	<u>10527</u>
\@textsuperscript	<u>10523</u>
\@title	<u>398</u>
\@wrglossary	<u>8836</u>
\@wrindex	<u>8825</u>
\[.	<u>9218</u>
\\"	<u>600</u>
\\$	<u>537</u>
\\$\$.	<u>537</u>
2in1 (package)	<u>615</u>
2up (package)	<u>615</u>

A

a4 (package)	<u>615</u>
a4wide (package)	<u>616</u>
a5comb (package)	<u>616</u>
abstract (environment)	<u>6363</u>
abstract (package)	<i>143</i> , <u>616</u>
\abstractname	<u>123</u> , <u>6362</u>
accsupp (package)	<u>619</u>
acro (package)	<u>619</u>
acronym (package)	<u>621</u>
\addcontentsline	<u>8599</u>
addlines (package)	<u>623</u>
\AddSubtitlePublished	<u>6328</u>
adjmulticol (package)	<u>622</u>
Adobe (program)	<u>86</u>
\affiliation	<u>6180</u>
afterpage (package)	<u>623</u>
algorithm2e (package)	<u>624</u>
algorithmicx (package)	<u>628</u>
algorithmx (package)	<u>172</u>

align (environment)	127	booklet (package)	656
align* (environment)	153	bookmark (package)	657
alignat (environment)	231	booktabs (package)	657
alignat* (environment)	257	boolean:	
alltt (package)	629	CombineHigherDepths	122, 368
\AmS	10812	FileSectionNames	123, 335
amsmath (package)	629	FormatEPUB	184, 249
amsthm (package)	636	FormatWP	186, 249
\and	398	HTMLDebugComments	123, 243
anonchap (package)	641	LWR@amsmultiline	548
anysize (package)	641	LWR@copiedsidetoc	512
appendix (package)	143, 642	LWR@doingapar	351
arabicfront (package)	642	LWR@doingcmidrule	427
array (package)	643	LWR@doingstartpars	351
\arrayrulecolor	7150	LWR@doingtbrule	427
\arrayrulecolornexttoken	7150	LWR@dynamicmath	332
arydshln (package)	643	LWR@emptyatbang	428
AsciiDoc (program)	86	LWR@exittingtabular	479
AsciiDoctor (program)	86	LWR@freezethisautoid	505
Asciidoctor-LaTeX (program)	86	LWR@indisplaymathimage	530
asymptote (package)	167, 645	LWR@infloatrow	768
atbegshi (package)	646	LWR@intabularmetadata	428
attachfile (package)	647	LWR@isstartingequation	556
attachfile2 (package)	648	LWR@mathmacro	331
\attrib	175, 409, 980	LWR@minipagefullwidth	579
\attribution	6373	LWR@minipagethispar	580
authblk (package)	143, 651	LWR@opttablecol	444
\author	130, 398	LWR@origmathjax	224
autosec	378	LWR@skipatbang	428
axessibility (package)	651	LWR@skippingmcolrowcell	428
axodraw2 (package)	652	LWR@skippingmrowcell	428
		LWR@starredlongtable	472
		LWR@startedrow	427
		LWR@tableparcell	428
		LWR@tabularcelladded	427
		LWR@tabularlumetmods	479
		LWR@tracinglwarp	243
		LWR@validtablecol	444
		LWR@verbtags	411
		mathjax	224
		usingOSWindows	223
		warpingHTML	224
		warpingprint	224
		WPMarkFloats	188, 249
		WPMarkLOFT	189, 250
		WPMarkMath	189, 250
		WPMarkMinipages	188, 249
		WPMarkTOC	188, 250
		WPTitleHeading	189, 250
		bophook (package)	659

B

babel (package)	176		
\backmatter	5673		
backref (package)	652		
balance (package)	652		
\BaseJobname	4657		
BaseJobname (option)	119, 225		
\bfseries	10506		
biblatex (package)	653		
\bibliography	8879		
\BibTeX	10807		
bibunits (package)	653		
bigdelim (package)	172, 654		
bigstrut (package)	655		
bitpattern (package)	655		
BlockClass (environment)	5015		
\BlockClassSingle	5029		
blowup (package)	656	bophook (package)	659

boundddvi (package)	659	cmap (package)	114
boxedminipage2e (package)	659	cmdtrack (package)	700
\boxframe	188	color (package)	165, 700
breakurl (package)	660	\colorbox	73
breqn (package)	661	\colorboxBlock	86, 9853
bsheaders (package)	662	colortbl (package)	700
BVerbatim (environment)	222	CombineHigherDepths (boolean)	122, 368
bxpapersize (package)	662	comment (package)	230
bytefield (package)	663	\ConTeXt	10804
		continue (package)	702
		copyrightbox (package)	703
		counter:	
		FileDepth	122, 368
C		FootnoteDepth	123, 361
calc (package)	233	lofdepth	517
cancel (package)	663	lotdepth	517
canoniclayout (package)	664	LWR@cellcolordepth	463
capt-of (package)	510	LWR@currentautosec	380
caption (package)	173, 240, 510, 664	LWR@externalfilecnt	529
\caption@begin	8580	LWR@hdashedlines	427
\caption@end	8580	LWR@hlines	427
\captionlistentry	8584	LWR@htmlfilenumber	335
\captionof	8615	LWR@latestautopage	507
\CaptionSeparator	8534	LWR@lateximagedepth	553
cases (package)	667	LWR@lateximagenumber	553
center (environment)	9748	LWR@LIPage	553
\centering	9772	LWR@midrulecounter	454
\centerline	9791	LWR@minipagedepth	579
changebar (package)	668	LWR@nextautoid	507
changepage (package)	669	LWR@nextautopage	507
chappg (package)	670	LWR@nextequation	541
\chapter	5898	LWR@prevFileDepth	379
chapterbib (package)	670	LWR@previousautopagelabel	370
chemfig (package)	670	LWR@spandepth	351
chemformula (package)	672	LWR@startingequation	556
chemgreek (package)	677	LWR@tablecolsindex	429
chemmacros (package)	677	LWR@tablecolswidth	429
chemnum (package)	697	LWR@tableLaTeXcolindex	429
chkfloat (package)	698	LWR@tabletotalLaTeXcols	429
chngpage (package)	669	LWR@tabletotalLaTeXcolsnext	429
cite (package)	698	LWR@tabularDepth	428
CJK (package)	699	LWR@tabularpardepth	428
CJKutf8 (package)	699	LWR@thisautoid	505
class (key) [Gin]	768	LWR@thisautoidWP	505
class:		SideTOCDepth	122, 515
internet	85	tocdepth	122
komascript	174	\cpagerefFor	9992
memoir	174, 175, 981	crop (package)	703
\cleardoublepage	10686	\CSSFilename	123, 126, 5250
\clearpage	10686	\CustomizeMathJax	5521, 5546

cuted (package)	704	warpall	129
cutwin (package)	704	warpHTML	125, 128
		warpprint	125, 128
environments:			
		abstract	6363
\date	130	align	127
dblfloatfix (package)	705	align*	153
dblfnote (package)	705	alignat	231
dcolumn (package)	706	alignat*	257
\DeclareGraphicsExtensions	2	BlockClass	5015
dejavu (package)	113	BVerbatim	222
description (environment)	6669	center	9748
diagbox (package)	706	description	6669
\displaymathnormal	162, 529, 546, 9410	enumerate	6650
\displaymathother	162, 529, 546, 9421	eqnarray	9470
ditaa (package)	183	equation	9388
\dotfill	10605	equation*	9396
\doublerulesepcolor	7152	fcolorminipage	166, 9894
\doublerulesepcolornexttoken	7152	flalign	179
dprogress (package)	708	flalign*	205
draftcopy (package)	708	flushleft	9764
draftfigure (package)	708	flushright	9756
draftwatermark (package)	709	fminipage	10245
dvipdfm (option)	116, 228	gather	75
dvipdfmx (option)	116, 228	gather*	101
dvips (option)	116, 228	itemize	6644
		lateximage	9618, 9743
E			
easy-todo (package)	709	list	6614
ebook (package)	710	longtable	2
ellipsis (package)	711	LWR@BlockClassWP	5056
embrac (package)	711	LWR@blocktextcurrentfont	10498, 10555
\emph	10306	LWR@displaymathnormal	9258
emptypage (package)	712	LWR@displaymathother	9263
\end@dblfloat	8497	LWR@equationother	9277
\end@float	8497	LWR@figcaption	8549
endfloat (package)	712	LWR@nestspan	4871
endheads (package)	713	math	9257
endnotes (package)	144, 713	minipage	10071
\enlargethispage	10685	multiline	20
\enskip	600, 10631	multiline*	47
enumerate (environment)	6650	picture	10052
enumerate (package)	714	quote	6389
enumitem (package)	714	tabbing	6473
environ (package)	234	tabular	8093
environment:		thebibliography	8889
fminipage	136	theindex	8801
lateximage	552	titlepage	6197
picture	578	titlingpage	14
titlepage	130	verbatim	6463
titlingpage	130		

verse	2, 6404	project.lwarpmkconf	264
warpall	727	project_html.tex	256
warpHTML	728	sample_project.css	127, 303
warpprint	728	tutorial.tex	96
epigraph (package)	715	filecontents (package)	233
epsfig (package)	716	FileDepth (counter)	122, 368
epstopdf (package)	166, 717	FileSectionNames (boolean)	123, 335
epstopdf (program)	164, 568	fix2col (package)	738
epstopdf-base (package)	717	fixme (package)	177, 738
eqnarray (environment)	9470	fixmetodonotes (package)	739
equation (environment)	9388	flafter (package)	740
equation* (environment)	9396	\flagverse	981
errata (package)	719	flalign (environment)	179
eso-pic (package)	720	flalign* (environment)	205
\etalchar	144	Flare (program)	86
etoolbox (package)	206	flippdf (package)	740
everyhook (package)	233	float (package)	172, 741
everypage (package)	721	floatflt (package)	742
everyshi (package)	721	floatpag (package)	743
expl3 (package)	233	floatrow (package)	173, 743
extramarks (package)	722	fltrace (package)	749
F			
fancybox (package)	137, 722	\flushbottom	4789
fancyhdr (package)	728	flushend (package)	749
fancyref (package)	729	flushleft (environment)	9764
fancytabs (package)	730	flushright (environment)	9756
fancyvrb (package)	730	fminipage (environment)	136, 10245
\fbox	136, 10220	fnbreak (package)	750
\fboxBlock	136, 10228	fncychap (package)	750
\fcolorbox	103, 9855	fnlineno (package)	751
\fcolorboxBlock	126, 9892	fnpos (package)	751
fcolorminipage (environment)	166, 9894	fontenc (package)	114, 751
figcaps (package)	737	fontspec (package)	114
figsize (package)	737	footmisc (package)	752
file:		footnote (package)	753
glyptounicode.tex	114	FootnoteDepth (counter)	123, 361
lwarp.css	127, 264	footnotehyper (package)	754
lwarp.ist	156, 304	footnoterange (package)	754
lwarp.xdy	156, 304	footnpag (package)	755
lwarp_baseline_marker.eps	530	\ForceHTMLPage	142, 5648
lwarp_baseline_marker.png	530	\ForceHTMLTOC	142, 5654
lwarp_formal.css	299	forest (package)	755
lwarp_mathjax.txt	306	FormatEPUB (boolean)	184, 249
lwarp_one_limage.cmd	305	FormatWP (boolean)	186, 249
lwarp_sagebrush.css	294	\framebox	10198
lwarp_tutorial.txt	96	framed (package)	756
lwarpmk.conf	263	FrameMaker (program)	86
lwarpmk.lua	196	\frontmatter	5670
project.css	127	ftnright (package)	758
		fullminipage (package)	758
		fullpage (package)	759

fullwidth (package)	759	\HTMLleftmargini (length)	175, 410, 411, 980
\fup	10531	\HTMLPageBottom	124, 399, 5244
\fussy	4791	\HTMLPageTop	124, 5241
fwlw (package)	759	\HTMLTitle	124, 131, 5271
G			
gather (environment)	75	\HTMLTitleAfterSection	124, 5991
gather* (environment)	101	\HTMLTitleBeforeSection	124, 5986
GELLMU (program)	85	\HTMLUnicode	4648
gentombow (package)	760	\HTMLVleftskip (length)	175, 410, 411, 980
geometry (package)	232, 760	hypcap (package)	781
gettitlestring (package)	233	hypdestopt (package)	782
[Gin]:		\hyperindexref	8861
class (key)	768	hypernat (package)	782
GladTeX (program)	85	hyperref (package)	140, 498, 782
glossaries (package)	144, 761	\hypertoc	8744
GlossaryCmd (option)	119, 144, 228, 761	\hypertocfloat	8764
glyptounicode.tex (file)	114	hyperxmp (package)	788
gmetric (package)	763	hyphenat (package)	788
graphics (package)	163, 763	I	
graphicx (package)	163, 777	idxlayout (package)	789
grffile (package)	165, 777	\if@titlepage	6175
grid (package)	777	ifoddpage (package)	790
grid-system (package)	778	ifplatform (package)	206
gridset (package)	778	imakeidx (package)	790
H			
hang (package)	779	\include	960
hanging (package)	780	\includegraphics	284
Hevea (program)	85	indentfirst (package)	795
\hfill	10601	InDesign (program)	86
\HomeHTMLFilename	4659	index (package)	795
HomeHTMLFilename (option)	116, 120, 227	\InlineClass	5040
\href	8391	\inlinemathnormal	162, 528, 4623
\hrulefill	10603	\inlinemathother	162, 528, 4619
\hskip	133	inputenc (package)	114
\hspace	133, 600, 10639, 10669, 10675	inputenx (package)	114
htlatex (program)	85	internet (class)	85
\HTMLAuthor	124, 131, 5276	intopdf (package)	797
HTMLDebugComments (boolean)	123, 243	itemize (environment)	6644
\HTMLDescription	124, 131, 5281	\itshape	10511
\HTMLEntity	4640	K	
\HTMLFilename	4658	\kern	133
HTMLfilename (option)	116, 120, 227	key:	
\HTMLFirstPageTop	124, 5238	[Gin]:	
htmlglossary (option) [lwarpmk]	144, 761	class	768
HTMLIndexCmd (option)	118, 227	keyfloat (package)	174, 797
\HTMLLanguage	123, 5968	komascript (class)	174
HTMLLatexCmd (option)	116, 179, 227	kvoptions (package)	224

L			
<code>\l@chapter</code>	8781	<code>listings</code> (package)	805
<code>\l@figure</code>	8792	<code>\listof</code>	8702
<code>\l@paragraph</code>	8790	<code>\listoffigures</code>	8676
<code>\l@part</code>	8780	<code>\listoftables</code>	8689
<code>\l@section</code>	8787	<code>\lltjext</code> (package)	809
<code>\l@subparagraph</code>	8791	<code>\modern</code> (package)	113, 114
<code>\l@subsection</code>	8788	<code>\lofdepth</code> (counter)	517
<code>\l@subsubsection</code>	8789	<code>\longtable</code> (environment)	2
<code>\l@table</code>	8793	<code>\longtable</code> (package)	171, 810
<code>\LaTeX</code>	10764	<code>\lotdepth</code> (counter)	517
<code>\LaTeX2HTML</code> (program)	85	<code>\landscape</code> (package)	812
<code>\LaTeXe</code>	10764	<code>\tablex</code> (package)	813
<code>\teximage</code> (environment)	552 , <u>9618</u> , 9743	<code>\tcaption</code> (package)	813
<code>\LateximageFontSizeScale</code>	9540	<code>\txgrid</code> (package)	813
<code>\LateximageFontSizeName</code>	9539	<code>\txtable</code> (package)	814
<code>\texmk</code> (option)	116, 120, 228	<code>\lua-check-hyphen</code> (package)	814
<code>\texmk</code> (program)	181	<code>\lua-visual-debug</code> (package)	814
<code>\texmkIndexCmd</code> (option)	119, 227	<code>\luacolor</code> (package)	815
<code>\TeXXML</code> (program)	85	<code>\LuaLaTeX</code>	10793
<code>\layaureo</code> (package)	799	<code>\LuaLaTeX</code> (program) [requirement]	89
<code>\layout</code> (package)	800	<code>\LuaTeX</code>	10793
<code>\leading</code> (package)	800	<code>\luatodonotes</code> (package)	177, 815
<code>\leftline</code>	9790	<code>\l warp</code> (package)	116
length:		<code>\l warp-patch-komascript</code> (package)	1013
<code>\HTMLleftmargini</code>	175, 410, 411, 980	<code>\l warp-patch-memoir</code> (package)	1015
<code>\HTMLvleftskip</code>	175, 410, 411, 980	<code>\l warp.css</code> (file)	127, 264
<code>\LWR@cmidrulewidth</code>	454	<code>\l warp.ist</code> (file)	156, 304
<code>\LWR@heavyrulewidth</code>	454	<code>\l warp.xdy</code> (file)	156, 304
<code>\LWR@lightrulewidth</code>	454	<code>\l warp_baseline_marker.eps</code> (file) . .	530
<code>\LWR@minipageheight</code>	579	<code>\l warp_baseline_marker.png</code> (file) . .	530
<code>\LWR@minipagewidth</code>	579	<code>\l warp_formal.css</code> (file)	299
<code>\LWR@tempheight</code>	601	<code>\l warp_mathjax.txt</code> (file)	306
<code>\LWR@tempraise</code>	601	<code>\l warp_one_limage.cmd</code> (file)	305
<code>\LWR@tempwidth</code>	601	<code>\l warp_sagebrush.css</code> (file)	294
<code>\LWR@thiscmidrulewidth</code>	454	<code>\l warp_tutorial.txt</code> (file)	96
<code>\LWR@thiscmidrulewidth</code>	454	[lwarpmk]:	
<code>\VerbatimHTMLWidth</code>	411	<code>\htmlglossary</code> (option)	144, 761
<code>\vleftmargini</code>	175, 410, 980	<code>\printglossary</code> (option)	144, 761
<code>\vleftskip</code>	175, 410, 980, 981	<code>\l warpmk</code> (option)	119, 226, 308
<code>\letltxmacro</code> (package)	206	<code>\l warpmk</code> (program)	196, 308
<code>\letterspace</code> (package)	800	<code>\l warpmk epstopdf</code> (program) . .	164, 568
<code>\letrine</code> (package)	801	<code>\l warpmk pdftosvg</code> (program) . .	164, 568
<code>\LibreOffice</code> (program)	86	<code>\l warpmk.conf</code> (file)	263
<code>\linebreak</code>	10679	<code>\l warpmk.lua</code> (file)	196
<code>\lineno</code> (package)	802	<code>\l warpsetup</code>	606
<code>\LinkHome</code>	4715, 4720	<code>\LWR@addbaselinemarker</code>	8981
<code>\linkhomename</code>	122, 4712	<code>\LWR@addcdashline</code>	7421
<code>\Linux</code> (program)	128, 222	<code>\LWR@addcmidruletrim</code>	7387
<code>\lips</code> (package)	804	<code>\LWR@addcmidrulewidth</code>	7418
<code>\list</code> (environment)	6614	<code>\LWR@addcompilecmd</code>	1335

\LWR@addformatwpa	7437	\LWR@descitem	6656
\LWR@addleftmostbartag	7222	\LWR@displaymathnormal (environment)	9258
\LWR@addmathjax	9326		
\LWR@addmulticolvertrulecolor	7696	\LWR@displaymathother (environment)	9263
\LWR@addrulewidth	7392	\LWR@docdashline	7351
\LWR@addtabularcellcolor	7565	\LWR@docmidrule	7335
\LWR@addtabularhrulecolor	7465	\LWR@doequation	9339
\LWR@addtabularrowcolor	7450	\LWR@doindexentry	8844
\LWR@addtabularrulecolors	7511	\LWR@doingapar (boolean)	351
\LWR@afterendverbatim	6446	\LWR@doingcmidrule (boolean)	427
\LWR@amsmathbody	9605	\LWR@doingstartpars (boolean)	351
\LWR@amsmathbodynumbered	9611	\LWR@doingtbrule (boolean)	427
\LWR@amsmultiline (boolean)	548	\LWR@domulticolumn	7734
\LWR@atbeginverbatim	6427	\LWR@doubledollar	9173
\LWR@backgroundcolor	36	\LWR@dynamicmath (boolean)	332
\LWR@beginhideamsmath	9455	\LWR@earlyclassloadnever	89
\LWR@blackborderpadding	10216	\LWR@earlyloadnever	79
\LWR@BlockClassWP (environment)	5056	\LWR@emptyatbang (boolean)	428
\LWR@blocktextcurrentfont (environment)	10498, 10555	\LWR@endfloatalignment	8530
\LWR@botnavigation	4734	\LWR@endhideamsmath	9463
\LWR@cdashlines	454	\LWR@endofline	10612
\LWR@cellcolordepth (counter)	463	\LWR@ensuredoingapar	5120
\LWR@cellHTMLcolor	7147	\LWR@equationother (environment)	9277
\LWR@checkloadbefore	59	\LWR@equationtag	9591
\LWR@checkloadfilename	827	\LWR@existingtabular (boolean)	479
\LWR@checkloadnever	73	\LWR@expandableformatted	1057
\LWR@clearmidrules	7302	\LWR@expandableformattedenv	1082
\LWR@closeparagraph	5144	\LWR@externalfilecnt (counter)	529
\LWR@closeprevious	4761	\LWR@figcaption (environment)	8549
\LWR@closetabledatagrid	6781	\LWR@filenamenoblanks	5432
\LWR@cmidrulewidth (length)	454	\LWR@filestart	6000
\LWR@colaferspec	430	\LWR@findcurrenttextcolor	10545
\LWR@colatspec	430	\LWR@findword	858
\LWR@colbangspec	430	\LWR@floatalignment	8516
\LWR@colbarspec	430	\LWR@floatalignmentname	8515
\LWR@colbeforespec	430	\LWR@floatbegin	8448
\LWR@columnHTMLcolor	7145	\LWR@floatend	8482
\LWR@columnspecklookahead	6900	\LWR@floatstyle	2
\LWR@compilecmd	1332	\LWR@footnotetext	5298
\LWR@compileuplate	1364	\LWR@forceminwidth	10205
\LWR@convertto	559	\LWR@forcenewpage	4778
\LWR@copiedsidetoc (boolean)	512	\LWR@formatted	1042
\LWR@copyfile	995	\LWR@formattedenv	1072
\LWR@createautosec	5679	\LWR@formatting	1041
\LWR@currentautosec (counter)	380	\LWR@freezethisauto	505
\LWR@currentcss	5249	\LWR@fVstyle	56
\LWR@currenttextcolor	10542	\LWR@getexpparray	4592
\LWR@customizedMathJax	5520	\LWR@getmynexttoken	6735
\LWR@customizeMathJax	5526	\LWR@hdashedlines (counter)	427

\LWR@heavyrulewidth (length)	454	\LWR@lateximagedepthref	8248
\LWR@hidelatexequation	9312	\LWR@lateximagenumber (counter)	553
\LWR@hlines (counter)	427	\LWR@lateximagenumberref	8251
\LWR@HTML@caption@begin	8559	\LWR@lateximagesfile	957
\LWR@HTML@caption@end	8571	\LWR@latextmkcmd	1350
\LWR@HTML@Gin@setfile	115	\LWR@latextmkdvipdfm	1355
\LWR@HTML@ref	8333	\LWR@latextmkvar	1344
\LWR@htmlblockcomment	4957	\LWR@lightrulewidth (length)	454
\LWR@htmlblocktag	4959	\LWR@LIpage (counter)	553
\LWR@HTMLcline	8078	\LWR@listitem	6599
\LWR@htmclosecomment	4929	\LWR@loadafter	23
\LWR@htmlcomment	4950	\LWR@loadbefore	50
\LWR@htmdivclass	4992	\LWR@loadnever	65
\LWR@htmdivclassend	4995	\LWR@longtabledatacaptiontag	7794
\LWR@htmlelement	5005	\LWR@lookforpackagename	861
\LWR@htmlelementclass	4979	\LWR@lwarpconfversion	1326
\LWR@htmlelementclassend	4984	\LWR@LwarpEnd	6149, 10857
\LWR@htmlelementclassline	4998	\LWR@LwarpStart	6080, 10857
\LWR@htmlelementend	5008	\LWR@maketitlesetup	33, 6269
\LWR@htmlfilename (counter)	335	\LWR@mathmacro (boolean)	331
\LWR@htmlfileref	8245	\LWR@maybenewtablerow	7154
\LWR@HTMLhline	8061	\LWR@maybeprintpendingfootnotes	5372
\LWR@HTMLLatexCmd	1374	\LWR@midrulecounter (counter)	454
\LWR@htmlabel	9440	\LWR@midrules	454
\LWR@htmlemulticolumn	7783	\LWR@minipagedepth (counter)	579
\LWR@htmlopencomment	4929	\LWR@minipagefullwidth (boolean)	579
\LWR@htmrefsectionfilename	4702	\LWR@minipageheight (length)	579
\LWR@HTMLsanitize	9543	\LWR@minipagestartpars	10619
\LWR@HTMLsanitizeexpand	9560	\LWR@minipagestopars	10622
\LWR@htmselectionfilename	4669	\LWR@minipagethispar (boolean)	580
\LWR@htmllspan	4906	\LWR@minipagewidth (length)	579
\LWR@htmllspanclass	4914	\LWR@multicolother	7637
\LWR@htmlltag	4924	\LWR@multicolpartext	7631
\LWR@htmlltagc	4860	\LWR@multicolskip	7643
\LWR@hyperindexrefnullified	8849	\LWR@multirowborder	3
\LWR@imageysizebox	114	\LWR@mynexttoken	6727
\LWR@includegraphicsb	133	\LWR@myshorttoc	8625
\LWR@indexitem	8813	\LWR@nameref	8242
\LWR@indexnameref	8843	\LWR@nestspan (environment)	4871
\LWR@indexsubitem	8817	\LWR@new@label	8282
\LWR@indexsubsubitem	8821	\LWR@newautoidanchor	8503
\LWR@indisplaymathimage (boolean)	530	\LWR@newautopagelabel	5512
\LWR@infloatrow (boolean)	768	\LWR@newhtmlfile	5549
\LWR@instertatbangcols	6773	\LWR@nextautoid (counter)	507
\LWR@intabularmetadata (boolean)	428	\LWR@nextautopage (counter)	507
\LWR@isolate	565	\LWR@nextequation (counter)	541
\LWR@isstartingequation (boolean)	556	\LWR@notltjloadafter	37
\LWR@itemizeitem	6637	\LWR@notmemoirloadafter	34
\LWR@latestautopage (counter)	507	\LWR@nullfonts	10431
\LWR@lateximagedepth (counter)	553	\LWR@nullifyNoAutoSpacing	8080

\LWR@nulllistfills	6606	\LWR@restoreorigformatting	8908
\LWR@opacity	113	\LWR@restoreoriglists	6703
\LWR@openparagraph	5126	\LWR@rotboxorigin	297
\LWR@opttablecol (boolean)	444	\LWR@rotstyle	105
\LWR@origcolspec	6767	\LWR@rowHTMLcolor	7146
\LWR@originname	298	\LWR@ruleHTMLcolor	7148
\LWR@originnames	305	\LWR@sanitize	4630
\LWR@origmathjax (boolean)	224	\LWR@sanitized	4629
\LWR@overline	10537	\LWR@scalestyle	108
\LWR@parseaftercolumn	6970	\LWR@section	5701
\LWR@parseatcolumn	6910	\LWR@sectionnumber	5676
\LWR@parsebangcolumn	6936	\LWR@select@html@hspace	10639
\LWR@parsebarcolumn	6981	\LWR@select@html@nohspase	10669
\LWR@parsebeforecolumn	6959	\LWR@select@print@hspase	10672
\LWR@parsecoloncolumn	7006	\LWR@setexpararray	4586
\LWR@parseDcolumn	7052	\LWR@setlatestname	8220
\LWR@parsedrequirepackagenames ..	826	\LWR@setOSWindows	594
\LWR@parsenormalcolumn	7031	\LWR@shellescapecmd	1327
\LWR@parsepcolumn	7043	\LWR@sidetoc	8718
\LWR@parsesemicoloncolumn	7027	\LWR@singledollar	9197
\LWR@parsetablecols	7058	\LWR@skipatbang (boolean)	428
\LWR@parsewcolumn	7047	\LWR@skippingmcolrowcell (boolean) ..	428
\LWR@patcherror	560	\LWR@skippingmrowcell (boolean) ..	428
\LWR@patchlists	6675	\LWR@spandepth (counter)	351
\LWR@pdfencoding	516	\LWR@splabel	8254
\LWR@PreloadedPackage	9795	\LWR@starredlongtable (boolean) ..	472
\LWR@prevFileDepth (counter)	379	\LWR@startedrow (boolean)	427
\LWR@previousautopagelabel (counter)	370	\LWR@startingequation (counter) ..	556
\LWR@printatbang	7197	\LWR@startingequationtag	9590
\LWR@printbartag	7187	\LWR@startnewdepth	5689
\LWR@printcloselist	6506	\LWR@startpars	5193
\LWR@PrintLatexCmd	1374	\LWR@startref	8295
\LWR@printlength	808	\LWR@stoppars	5211
\LWR@printmcoldata	7646	\LWR@stripperiod	8219
\LWR@printmcoltypes	7589	\LWR@strresult	6765
\LWR@printopenlist	6507	\LWR@subaddcmidruletrim	7374
\LWR@printpendingfootnotes	5362	\LWR@subaddtabularcellcolor	7559
\LWR@printpendingmpfootnotes	5381	\LWR@subcdashline	7338
\LWR@printthetitle	6221	\LWR@subcmidrule	7320
\LWR@providelength	556	\LWR@subhtmelementclass	4964
\LWR@ProvidesPackageDrop	938	\LWR@subhyperref	8367
\LWR@ProvidesPackagePass	925	\LWR@subhyperrefclass	8380
\LWR@pushoneclose	5685	\LWR@subhyperreftext	8375
\LWR@quickfile	954	\LWR@subinlinenimage	8441
\LWR@ref@ignoresstar	8341	\LWR@sublabel	8259
\LWR@remembertag	9595	\LWR@subnewref	8327
\LWR@replaceEPSSVG	122	\LWR@subsingledollar	9017
\LWR@requeststoc	6142	\LWR@subtableofcontents	8638
\LWR@requirepackagenames	825	\LWR@subtabularhtmlcolumns	7866
\LWR@restoreorigaccents	1283	\LWR@syncmathjax	9292

\LWR@tablecols	429	\LWRsetnextfloat	8543
\LWR@tablecolsindex (counter)	429	\LyX	10825
\LWR@tablecolswidth (counter)	429		
\LWR@tabledatacolumntag	7957	M	
\LWR@tabledatasinglecolumntag	7236	Mac OS (program)	128, 222
\LWR@tableLaTeXcolindex (counter)	429	Madcap (program)	86
\LWR@tableparcell (boolean)	428	magaz (package)	817
\LWR@tabletotallaTeXcols (counter)	429	\mainmatter	5666
\LWR@tabletotallaTeXcolsnext (counter)	429	make (program)	182
\maketitle	10176		
\LWR@tabularcelladded (boolean)	427	makeidx (package)	158, 818, 958, 959
\LWR@tabularDepth (counter)	428	\MakeIndex	10807
\LWR@tabularendofline	6880	makeindex (option)	116, 120, 227
\LWR@tabularfinishrow	6834	makeindex (program)	156
\LWR@tabularhtmlcolumns	7876	makeindexStyle (option)	116, 156, 225
\LWR@tabularleftedge	7227	\makelabel	415
\LWR@tabularmutemods (boolean)	479	\maketitle	47, 130, 6281
\LWR@tabularpardepth (counter)	428	marginal (package)	818
\LWR@tdaddstyle	7362	marginfit (package)	818
\LWR@tdendstyles	7368	marginfix (package)	819
\LWR@tdstartstyles	7361	marginnote (package)	819
\LWR@tempcolor	10543	\marginpar	135, 366, 5393
\LWR@tempcolortwo	10543	\marginparBlock	135, 366, 5404, 5421
\LWR@tempheight (length)	601	\markboth	4786
\LWR@tempraise (length)	601	\markright	4787
\LWR@tempwidth (length)	601	math (environment)	9257
\LWR@textcurrentcolor	19, 10548	\mathimagename	125, 8968
\LWR@textcurrentfont	10477, 10554	MathJax (program)	160, 161, 527
\LWR@thisautoid (counter)	505	MathJax (program) [requirement]	89
\LWR@thisautoidWP (counter)	505	mathjax (boolean)	224
\LWR@thiscmidrulewidth (length)	454	mathjax (option)	116, 120, 225
\LWR@thiscols	7235	mathsvg (option)	116, 120, 225
\LWR@thisfilename	5430	\mbox	10173
\LWR@thisnewfilename	5431	\mcaption (package)	820
\LWR@titlingmaketitle	83, 6320	\mcolrowcell	8032
\LWR@topnavigation	4731	\mdframed (package)	139, 820
\LWR@traceinfo	1023	\mdseries	10505
\LWR@tracingl warp (boolean)	243	memhfixc (package)	830
\LWR@triml rules	454	memoir (class)	174, 175, 981
\LWR@trimr rules	454	metalogo (package)	830
\LWR@unknownenGINE	1339	mhchem (package)	831
\LWR@validtablecol (boolean)	444	microtype (package)	231, 833
\LWR@verbtags (boolean)	411	midfloat (package)	834
\LWR@vspace	10676	midpage (package)	834
\LWR@WPcell	7433	\MiKTeX	10824
\LWR@writeconf	1476	minipage (environment)	10071
\LWR@xcolorrowHTMLcolor	7144	\minipagewidth	10063
\LWRopquote	589	morefloats (package)	835
\LWRopseq	590	moreverb (package)	835
\LWRPrintStack	4739	\mparhack (package)	836

\mrowcell	8029
MS-Windows (program)	128, 222
multicol (package)	837
multicolrule (package)	838
\multicolumnrow	39, 7899
\multirow	839
multirow (package)	838
multitoc (package)	842
multiline (environment)	20
multiline* (environment)	47
musicography (package)	842
N	
\Nameref	8359
\nameref	8350
nameref (package)	844
natbib (package)	844
nccfancyhdr (package)	845
needspace (package)	846
newclude (package)	176
\newfloat	4
newfloat (package)	234
\newline	10610
\newpage	10607
\newtheorem	414
newtxmath (package)	163
newunicodechar (package)	114
nextpage (package)	846
nicefrac (package)	163, 847
\noalign	8050
\nolinebreak	10680
\nolinkurl	8407
nonfloat (package)	848
nonumonpart (package)	848
\nopagebreak	10684
\nopagecolor	72
nopageno (package)	848
\normalfont	10519
\normalmarginpar	5418
notespaces (package)	848
nowidow (package)	849
ntheorem (package)	162, 849
\numberline	8739
numindex (option) [tocbibind]	158, 959
O	
octave (package)	862
OpenOffice (program)	86
option:	
--shell-escape	113
[lwarpmk]:	
htmlglossary	144, 761
printglossary	144, 761
[tocbibind]:	
numindex	158, 959
[tocloft]:	
titles	143
BaseJobname	119, 225
dvipdfm	116, 228
dvipdfmx	116, 228
dvips	116, 228
GlossaryCmd	119, 144, 228, 761
HomeHTMLFilename	116, 120, 227
HTMLFilename	116, 120, 227
HTMLIndexCmd	118, 227
HTMLLatexCmd	116, 179, 227
latexmk	116, 120, 228
LatexmkIndexCmd	119, 227
lwarpmk	119, 226, 308
makeindex	116, 120, 227
makeindexStyle	116, 156, 225
mathjax	116, 120, 225
mathsvg	116, 120, 225
OSWindows	119, 128, 222, 226
pdftotextEnc	119, 226
PrintIndexCmd	118, 227
PrintLatexCmd	116, 179, 227
warpHTML	119, 225
warpprint	119, 224
xindy	116, 120, 228
xindyCodepage	118, 226
xindyLanguage	118, 226
xindyStyle	116, 157, 226
\OSPathSymbol	593
OSWindows (option)	119, 128, 222, 226
overpic (package)	168, 863
P	
package:	
2in1	615
2up	615
a4	615
a4wide	616
a5comb	616
abstract	143, 616
accsupp	619
acro	619
acronym	621
addlines	623
adjmulticol	622

afterpage	623	chemfig	670
algorithm2e	624	chemformula	672
algorithmicx	628	chemgreek	677
algorithmx	172	chemmacros	677
alltt	629	chemnum	697
amsmath	629	chkfloat	698
amsthm	636	chngpage	669
anonchap	641	cite	698
any size	641	CJK	699
appendix	143, 642	CJKutf8	699
arabicfront	642	cleveref	140, 574
array	643	clrdblp	700
arydshln	643	cm-super	113
asymptote	167, 645	cmap	114
atbegshi	646	cmdtrack	700
attachfile	647	color	165, 700
attachfile2	648	colortbl	700
authblk	143, 651	comment	230
axessibility	651	continue	702
axodraw2	652	copyrightbox	703
babel	176	crop	703
backref	652	cuted	704
balance	652	cutwin	704
biblatex	653	dblfloatfix	705
bibunits	653	dblfnote	705
bigdelim	172, 654	dcolumn	706
bigstrut	655	dejavu	113
bitpattern	655	diagbox	706
blowup	656	ditaa	183
booklet	656	dprogress	708
bookmark	657	draftcopy	708
booktabs	657	draftfigure	708
bophook	659	draftwatermark	709
boundddvi	659	easy-todo	709
boxedminipage2e	659	ebook	710
breakurl	660	ellipsis	711
breqn	661	embrac	711
bsheaders	662	emptypage	712
bxpapersize	662	endfloat	712
bytefield	663	endheads	713
calc	233	endnotes	144, 713
cancel	663	enumerate	714
canoniclayout	664	enumitem	714
capt-of	510	environ	234
caption	173, 240, 510, 664	epigraph	715
cases	667	epsfig	716
changebar	668	epstopdf	166, 717
change page	669	epstopdf-base	717
chappg	670	errata	719
chapterbib	670	eso-pic	720

etoolbox	206	grffile	165, 777
everyhook	233	grid	777
everypage	721	grid-system	778
everyshi	721	gridset	778
expl3	233	hang	779
extramarks	722	hanging	780
fancybox	137, 722	hypcap	781
fancyhdr	728	hypdestopt	782
fancyref	729	hypernat	782
fancytabs	730	hyperref	140, 498, 782
fancyvrb	730	hyperxmp	788
figcaps	737	hyphenat	788
figsize	737	idxlayout	789
filecontents	233	ifoddpage	790
fix2col	738	ifplatform	206
fixme	177, 738	imakeidx	790
fixmetodonotes	739	indentfirst	795
flafter	740	index	795
flippdf	740	inputenc	114
float	172, 741	inputenx	114
floatflt	742	intopdf	797
floatpag	743	keyfloat	174, 797
floatrow	173, 743	kvoptions	224
fltrace	749	layaureo	799
flushend	749	layout	800
fnbreak	750	leading	800
fncychap	750	letltxmacro	206
fnlineno	751	letterspace	800
fnpos	751	lettrine	801
fontenc	114, 751	lineno	802
fontspec	114	lips	804
footmisc	752	listings	805
footnote	753	lltjext	809
footnotehyper	754	lmodern	113, 114
footnoterange	754	longtable	171, 810
footnpag	755	lscape	812
forest	755	ltablex	813
framed	756	ltcaption	813
ftnright	758	ltxgrid	813
fullminipage	758	ltxtable	814
fullpage	759	lua-check-hyphen	814
fullwidth	759	lua-visual-debug	814
fwlw	759	luacolor	815
gentombow	760	luatodonotes	177, 815
geometry	232, 760	lwarp	116
getttitlestring	233	lwarp-patch-komascript	1013
glossaries	144, 761	lwarp-patch-memoir	1015
gmeometric	763	magaz	817
graphics	163, 763	makeidx	158, 818, 958, 959
graphicx	163, 777	marginal	818

marginfit	818	pdfsync	872
marginfix	819	pdftricks	166, 873
marginnote	819	pdfx	873
mcaption	820	perltex	181
mdframed	139, 820	pfnote	874
memhfixc	830	phfqit	874
metalogo	830	placeins	875
mhchem	831	plarydshln	875
microtype	231, 833	plex	875
midfloat	834	plexarydshln	876
midpage	834	plextcolortbl	876
morefloats	835	polyglossia	176
moreverb	835	prelim2e	877
mparhack	836	prettyref	877
multicol	837	preview	877
multicolrule	838	printlen	235
multirow	838	psfrag	167, 878
multitoc	842	psfragx	878
musicography	842	pst-eps	879
nameref	844	pstool	167, 879
natbib	844	pstricks	166, 880
nccfancyhdr	845	pxatbegshi	880
needsspace	846	pxeveryshi	881
newclude	176	pxftnright	881
newfloat	234	pxjahyper	881
newtxmath	163	pythontex	181
newunicodechar	114	quotchap	882
nextpage	846	quoting	883
nicefrac	163, 847	ragged2e	883
nonfloat	848	realscripts	884
nonumonpart	848	refcheck	885
nopageno	848	refcount	234
notespages	848	register	885
nowidow	849	relsize	134, 886
ntheorem	162, 849	repeatindex	887
octave	862	resizegather	888
overpic	168, 863	rmpage	889
pagegrid	864	romanbar	889
pagenote	143, 864	romanbarpagenumber	889
pagesel	865	rotating	890
paralist	865	rotfloat	890
parnotes	866	rterface	182
parskip	867	rviewport	891
pbox	867	sagetex	181
pdfcomment	868	savetrees	892
pdflandscape	868	scalefnt	892
pdfmarginpar	868	schemata	892
pdfpages	869	scrextend	893
pdfprivacy	872	scrhack	896
pdfrender	872	scrlayer	896

scrlayer-notecolumn	898	tocbibind	158, 958, 959
scrlayer-scrpage	898	tocenter	960
scrpage2	899	tocloft	143, 159, 641, 961
section	901	tocstyle	967
sectionbreak	902	todo	968
sectsty	902	todonotes	177, 969
semantic-markup	903	tram	971
setspace	904	transparent	971
shadow	905	trimclip	972
showidx	906	trivfloat	172, 972
showkeys	906	turnthepage	974
sidecap	906	twoup	974
sidenotes	907	typearea	974
SIunits	909	ulem	975
siunitx	163, 567, 909	underscore	977
soul	915	units	163, 977
soulpos	917	unitsdef	978
soulutf8	917	upref	979
splitidx	918	url	140, 979
srcltx	919	uspace	979
srctex	920	varioref	140
stabular	920	varwidth	235
stffloats	921	verbatim	234
subcaption	173	verse	175, 980, 981
subfig	173, 921	versionotes	982
subfigure	926	verbbars	982
supertabular	171, 927	vmargin	983
sympytex	182	vowel	983
syntonly	929	vpe	984
tabls	929	vwcol	984
tabularx	930	wallpaper	986
tabulary	930	watermark	987
tascmac	931	widows-and-orphans	987
textarea	932	wrapfig	988
textcomp	114, 134, 933	xbmks	989
textfit	936	xcolor	165, 570, 989
textpos	937	xexchangebar	1001
theorem	937	xellipsis	1001
thinsp	941	xetexko-vertical	1002
threadcol	942	xfrac	1002
threeparttable	942	xifthen	234
thumb	943	xltabular	1005
thumbs	943	xltxtra	1005
tikz	165, 944	xmpincl	1006
titleps	945	xparse	178, 233
titleref	948	xpatch	206
titlesec	949	xpiano	1006
titletoc	951	xpinyin	1007
titling	143, 953	xstring	234
tocbasic	958	xtab	171, 1007

xunicode	1009	Plastex (program)	85
xurl	1010	plext (package)	875
xy	1010	plextrydshln (package)	876
zhlineskip	1011	plextcolortbl (package)	876
zref	235	\PN@parnotes@auto	5125
zwpagelayout	1011	polyglossia (package)	176
\packagediagramname	125, 8969	\popclose	4559
\pagebreak	10681	\postchaptername	5869
\pagecolor	71	\postpartname	5867
pagegrid (package)	864	\postsectionname	5871
pagenote (package)	143, 864	\prechaptername	5869
\pagenumbering	4792	prelim2e (package)	877
\pageref	8345	\prepartname	5867
\pagerefPageFor	8344	\presectionname	5871
pagesel (package)	865	prettyref (package)	877
\pagestyle	4784	preview (package)	877
Pandoc (program)	86	\printauthor	399, 6229, 6248
\paragraph	5950	\printdate	399, 6240, 6250
paralist (package)	865	printglossary (option) [lwarpmk]	144, 761
\parbox	10164	\printindex	2
parnotes (package)	866	PrintIndexCmd (option)	118, 227
\parsemulticolumnalignment	7681	PrintLatexCmd (option)	116, 179, 227
parskip (package)	867	printlen (package)	235
\part	5885	\printthanks	398, 401
pbox (package)	867	\printtitle	399, 6213, 6247
pdfcomment (package)	868	program:	
pdfcrop (program) [requirement]	89	[requirement]:	
pdflaTeX (program) [requirement]	89	LuaLaTeX	89
pdflscape (package)	868	MathJax	89
pdfmarginpar (package)	868	pdfcrop	89
pdfpages (package)	869	pdfLaTeX	89
pdfprivacy (package)	872	pdfseparate	89, 94
pdfrender (package)	872	pdftocairo	89, 94
pdfseparate (program) [requirement]	89, 94	pdftotext	89, 94
pdfsync (package)	872	perl	95
pdftocairo (program)	164, 568	XeLaTeX	89
pdftocairo (program) [requirement]	89, 94	Adobe	86
pdftotext (program) [requirement]	89, 94	AsciiDoc	86
pdftotextEnc (option)	119, 226	AsciiDoctor	86
pdftricks (package)	166, 873	Asciidoc-LaTeX	86
pdfx (package)	873	epstopdf	164, 568
perl (program) [requirement]	95	Flare	86
perltex (package)	181	FrameMaker	86
pfnote (package)	874	GELLMU	85
\phantomsection	10742	GladTeX	85
phfqt (package)	874	Hevea	85
picture (environment)	578, 10052	htlatex	85
placeins (package)	875	InDesign	86
plarydshln (package)	875	LaTeX2HTML	85
		latexmk	181

LaTeXML	85	\raggedright	9784
LibreOffice	86	\raisebox	10297
Linux	128, 222	realscripts (package)	884
lwarpmk	196, 308	\ref	8333
lwarpmk epstopdf	164, 568	refcheck (package)	885
lwarpmk pdftosvg	164, 568	refcount (package)	234
Mac OS	128, 222	\reflectbox	351
Madcap	86	register (package)	885
make	182	relsize (package)	134, 886
makeindex	156	repeatindex (package)	887
MathJax	160, 161, 527	[requirement]:	
MS-Windows	128, 222	LuaTeX (program)	89
OpenOffice	86	MathJax (program)	89
Pandoc	86	pdfcrop (program)	89
pdftocairo	164, 568	pdfLaTeX (program)	89
Plastex	85	pdfseparate (program)	89, 94
TeX2page	85	pdftocairo (program)	89, 94
TeX4ht	85	pdftotext (program)	89, 94
TeXMaths	189	perl (program)	95
TtH	85	XeLaTeX (program)	89
Unix	128, 222	\RequirePackage	889
Windows	128, 222	\resizebox	360
Word	86	\resizegather (package)	888
xindy	156	\ResumeTabular	7944
project.css (file)	127	\reversemarginpar	5417
project.lwarpmkconf (file)	264	\rightline	9792
project_html.tex (file)	256	\rmfamily	10507
psfrag (package)	167, 878	rmpage (package)	889
psfrags (package)	878	romanbar (package)	889
pst-eps (package)	879	romanbarpagenumber (package)	889
pstool (package)	167, 879	\rotatebox	317
pstricks (package)	166, 880	rotating (package)	890
\pushclose	4532	rotfloat (package)	890
pxatbegshi (package)	880	\rowcolor	7149
pxeveryshi (package)	881	\rownum	7141
pxftnright (package)	881	rterface (package)	182
pxjahyper (package)	881	\rule	10688
pythontex (package)	181	rviewport (package)	891

Q	S		
\quadquad	600, 10630	sagetex (package)	181
\quadquad	600, 10625	sample_project.css (file)	127, 303
quotchap (package)	882	savetrees (package)	892
quote (environment)	6389	\sb	10521
quoting (package)	883	\scalebox	336
R		scalefont (package)	892
ragged2e (package)	883	schemata (package)	892
\raggedbottom	4788	scrextend (package)	893
\raggedleft	9778	scrhack (package)	896
		scrlayer (package)	896

	T
scrlayer-notecolumn (package)	<i>898</i>
scrlayer-scrpage (package)	<i>898</i>
scrpage2 (package)	<i>899</i>
\scshape	<i>10512</i>
\section	<i>5924</i>
section (package)	<i>901</i>
sectionbreak (package)	<i>902</i>
sectsty (package)	<i>902</i>
semantic-markup (package)	<i>903</i>
\SetHTMLFileNumber	<i>4660</i>
setspace (package)	<i>904</i>
\sffamily	<i>10508</i>
\sfrac	<i>1003</i>
shadow (package)	<i>905</i>
showidx (package)	<i>906</i>
showkeys (package)	<i>906</i>
sidecap (package)	<i>906</i>
sidenotes (package)	<i>907</i>
SideTOCDepth (counter)	<i>122, 515</i>
\sidetocname	<i>122, 8715</i>
\simplechapterdelim	<i>5696</i>
\sishape	<i>10513</i>
SIunits (package)	<i>909</i>
siunitx (package)	<i>163, 567, 909</i>
\sloppy	<i>4790</i>
\slshape	<i>10518</i>
soul (package)	<i>915</i>
soulpos (package)	<i>917</i>
soulutf8 (package)	<i>917</i>
\sp	<i>10520</i>
splitidx (package)	<i>918</i>
\srcitx (package)	<i>919</i>
\srctex (package)	<i>920</i>
stabular (package)	<i>920</i>
\StartDefiningMath	<i>4607</i>
\StartDefiningTabulars	<i>4597</i>
stfloats (package)	<i>921</i>
\StopDefiningMath	<i>4611</i>
\StopDefiningTabulars	<i>4601</i>
subcaption (package)	<i>173</i>
subfig (package)	<i>173, 921</i>
subfigure (package)	<i>926</i>
\subparagraph	<i>5958</i>
\subsection	<i>5933</i>
\subsubsection	<i>5941</i>
supertabular (package)	<i>171, 927</i>
sympytex (package)	<i>182</i>
syntonly (package)	<i>929</i>
tabbing (environment)	<i>6473</i>
\tableofcontents	<i>124, 8657</i>
tbls (package)	<i>929</i>
tabular (environment)	<i>8093</i>
\TabularMacro	<i>7942</i>
tabularx (package)	<i>930</i>
tabulary (package)	<i>930</i>
tascmac (package)	<i>931</i>
\TeX	<i>10750</i>
\TeX2page (program)	<i>85</i>
\TeX4ht (program)	<i>85</i>
\TeXMaths (program)	<i>189</i>
textarea (package)	<i>932</i>
\textbf	<i>10324</i>
\textcolor	<i>48</i>
\textcomp (package)	<i>114, 134, 933</i>
\textfit (package)	<i>936</i>
\textgreater	<i>4650</i>
\textit	<i>10369</i>
\textless	<i>4650</i>
\textmd	<i>10315</i>
\textnormal	<i>10416</i>
\textpos (package)	<i>937</i>
\textrm	<i>10333</i>
\textsc	<i>10378</i>
\textsf	<i>10342</i>
\textsi	<i>10390</i>
\textsl	<i>10407</i>
\textsubscript	<i>10524</i>
\textsuperscript	<i>10522</i>
\texttt	<i>10351</i>
\textup	<i>10360</i>
\tfl@chapter@fix	<i>973</i>
\thanks	<i>131</i>
\thanksmarkseries	<i>91</i>
\theauthor	<i>399</i>
\thebibliography (environment)	<i>8889</i>
\thedate	<i>399</i>
\theHTMLSection	<i>5997</i>
\theHTMLTitleSection	<i>5996</i>
\theHTMLTitleSeparator	<i>5973</i>
\theindex (environment)	<i>8801</i>
\thempfootnote	<i>5359</i>
\theorem (package)	<i>937</i>
\thetitle	<i>399</i>
\thinsp (package)	<i>941</i>
\thispagestyle	<i>4785</i>
\threadcol (package)	<i>942</i>
\threepartable (package)	<i>942</i>

thumb (package)	943	
thumbs (package)	943	
tikz (package)	165, 944	
\title	130, 5260	
titlepage (environment)	130, 6197	
titleps (package)	945	
titleref (package)	948	
titles (option) [tocloft]	143	
titlesec (package)	949	
titletoc (package)	951	
titling (package)	143, 953	
titlingpage (environment)	14, 130	
tocbasic (package)	958	
[tocbibind]:		
numindex (option)	158, 959	
tocbibind (package)	158, 958, 959	
tocdepth (counter)	122	
toccenter (package)	960	
[tocloft]:		
titles (option)	143	
tocloft (package)	143, 159, 641, 961	
tocstyle (package)	967	
todo (package)	968	
todonotes (package)	177, 969	
\tracingl warp	201, 1022	
tram (package)	971	
transparent (package)	971	
trimclip (package)	972	
trivfloat (package)	172, 972	
\ttfamily	10509	
TtH (program)	85	
turnthepage (package)	974	
tutorial.tex (file)	96	
twoup (package)	974	
typearea (package)	974	
		W
U		
ulem (package)	975	
\underline	10532	
underscore (package)	977	
units (package)	163, 977	
unitsdef (package)	978	
Unix (program)	128, 222	
\up	10530	
upref (package)	979	
\upshape	10510	
\url	8424	
url (package)	140, 979	
usingOSWindows (boolean)	223	
uspace (package)	979	
		X
xbmks (package)	989	
xcolor (package)	165, 570, 989	
xchangebar (package)	1001	
\XeLaTeX	10795	
XeLaTeX (program) [requirement]	89	
xellipsis (package)	1001	
\XeTeX	10795	
xetexko-vertical (package)	1002	

xfrac (package)	<i>1002</i>	xpatch (package)	<i>206</i>
\xfracHTMLfontsize	<i>3</i>	xpiano (package)	<i>1006</i>
xifthen (package)	<i>234</i>	xpinyin (package)	<i>1007</i>
xindy (option)	<i>116, 120, 228</i>	xstring (package)	<i>234</i>
xindy (program)	<i>156</i>	xtab (package)	<i>171, 1007</i>
xindyCodepage (option)	<i>118, 226</i>	xunicode (package)	<i>1009</i>
xindyLanguage (option)	<i>118, 226</i>	xurl (package)	<i>1010</i>
xindyStyle (option)	<i>116, 157, 226</i>	xy (package)	<i>1010</i>
xltabular (package)	<i>1005</i>		Z
xltxtra (package)	<i>1005</i>	zhlineskip (package)	<i>1011</i>
xmpincl (package)	<i>1006</i>	zref (package)	<i>235</i>
xparse (package)	<i>178, 233</i>	zwpagelayout (package)	<i>1011</i>

General Index

This is an index of instructions and concepts. Look here when wondering how to do something, and check the Troubleshooting Index when something goes wrong.

Symbols	
\,	133
~	133
A	
accents	
in section & file names	376
adapting	
class	195
document	112
package	194
affiliation	
multiple authors	142
algorithmic	
with newfloat, trivfloat	973
array	
mhchem	831
author	
HTML meta tag	131, 359
multiple	142
B	
baseline	
tabular	442
bibliography	
HTML page and TOC	142
bitmapped fonts	113
bugs	197
C	
<i>Calibre</i>	184
chemistry	
Greek symbols	677
class	
modifying for lwrap	195
code listings	133
compiling	
custom	179
Computer Modern	113
D	
converting	
class	195
document	112
package	194
CSS	
file selection	127
lwrap.css	127
per HTML page	127
project-specific changes	127
D	
danger icon	205
debugging	
HTML debug comments	243
tracing log	243
Deja Vu	113
description	
HTML meta tag	131, 360
display math	
complicated objects	162
document	
convert existing	112
DVI LATEX	108, 113
dynamic math	162
dynamic math expressions	331
E	
endnotes	
HTML page and TOC	142
EPS image	
converting	110
using	163, 568
EPUB	
conversion software	184
HTML conversion settings	184, 249
error messages	197
export	
to word processor	186

	F	
FAQ	197	
file		
accent in name	376	
hashed name	531, 558	
underscore in name	120, 134	
unique names	123	
font		
Computer Modern	113	
Deja Vu	113	
ligatures	114	
packages	114	
selection	113	
size		
xfrac	1003	
latextimage	160, 526, 553	
math, SVG	160, 526, 553	
type 1 vector	113	
type 3 bitmapped	113	
footnotes	360	
framed objects	136	
Frequently Asked Questions	197	
	G	
generator		
HTML meta tag	391	
GIF images	164, 569	
glossaries		
HTML page and TOC	142	
glossary		
language	144	
options	144	
processing	109	
graphics		
formats	163, 568	
Greek		
chemistry symbols	677	
	H	
hash		
SVG image filename	531, 558	
heading, word processor	189	
horizontal space	133	
between minipages	600	
HTML		
conversion settings	120	
debug comments	243	
EPUB	184, 249	
word processor	186, 249	
conversion suggestions	132	
	I	
entities, conversion	133	
filename generation	126	
headings	204	
meta tag		
author	131, 359	
description	131, 360	
generator	391	
title	124, 131, 359	
viewport	391	
selecting print/HTML definitions . . .	244	
tabular column conversion	443	
verbatim, in	133	
hyperref		
and xindy	146	
	I	
icon		
warning	205	
image		
GIF	164, 569	
graphicx package	568	
hashed filename	531, 558	
PDF or EPS		
converting	110, 164, 569	
using	163, 568	
PNG and JPG	164, 569	
processing	308	
\includegraphics		
using	163, 568	
index		
HTML page and TOC	142	
custom makeindex style file	156	
custom xindy style file	156	
custom display styles	157	
letter headings	790	
placement and TOC options	158, 958	
processing	101, 102, 146	
see, seealso, ranges	146	
source code	146	
table of contents	158, 958	
UTF-8	115	
xindy and hyperref	146	
inline math		
complicated objects	162	
item		
empty	415	
	J	
JAVASCRIPT		
MATHJAX	159, 525	

JPG images	164, 569
L	
label	
in HTML	394
math environment	548
language	
glossary	144
language HTML metadata	389
latextimage	
font size	160, 526, 553
processing	308
Latin Modern font	113
LIBREOFFICE	
conversion recommendations	189
import into	186
section headings	189
ligatures	114, 231
line numbers	205
list	
empty item	415
filename	134
listings, program code	133
locale	182
LuaL ^A T _E X	
detection	206
file & section names	376
l warp	
loading	116
options	116
l warp.ist	
customizing	156
l warp.xdy	
customizing	156
l warpmk	
customizing	196
M	
make utility	182
makeindex	
customizing	156
margin	
numbers	205
tags	205
markup languages	86
math	
mathjax option	225
mathsvg option	225
mhchem	831
display with complicated objects	162
dynamic	162, 331
font size — SVG	160, 526, 553
inline with complicated objects	162
MATHJAX custom functions	160, 527
MATHJAX summary	160, 527
SVG summary	159, 526
word processor conversion	189
MATHJAX	
custom functions	160, 527
mathjax option	225
summary	160, 527
MD5 hash	
SVG image filename	531, 558
memoir	
verse	981
meta tag, HTML	
author	131, 359
description	131, 360
generator	391
title	131, 359
viewport	391
minipage	
framed	136
horizontal space between	600
modifying	
class	195
document	112
package	194
\multicolumn	
with \multirow	841
\multirow	
with \multicolumn	841
N	
newfloat	
with trivfloat, algorithmic	973
numbers	
left margin	205
P	
package	
modifying for l warp	194
required	231
\par	
hooks	356
PDF image	
converting	110
using	163, 568
PERL	95
PNG images	164, 569

POPPLER	89, 94	tracing log	243
print		trivfloat	
selecting print/HTML definitions	244	with newfloat, algorithmic	973
problems	197	troubleshooting	197
program listings	133	HTML debug comments	243
programs		tracing log	243
utility	88	type 1 vector fonts	113
\published	406	type 3 bitmapmed fonts	113
S			
section		underscore	
depths	204	filename	120, 134
heading, word processor	189	Unicode	
settings		enhanced coverage	113
CSS project-specific	127	file & section names	376
CSS selection	127	input characters	220
HTML conversion	120	selection	113
lwrap package options	116	UTF-8	
selecting output	128	enhanced coverage	113
title page	130	file & section names	376
shell escape	113	index	115
siunitx		locale	182
with <i>TeXMaths</i>	189	selection	113
space		utility	
horizontal	133	programs	88
between minipages	600		
stack depths	204	V	
\subtitle	406	vector fonts	113
SVG		verbatim	
mathsvg option	225	code and HTML tags	133
converting from PDF or EPS	110	viewport	
dynamic math	331	HTML meta tag	391
image processing	308		
math summary	159, 526	W	
		warning icon	205
T			
tabular		word processor	
baseline	442	conversion recommendations	189
HTML colummn conversion	443	HTML conversion settings	186, 249
macros inside	170, 424	section headings	189
\multicolumn with \multirow	841		
tikz		X	
catcodes	944	XeL ^A T _E X	
dollar redefined	944	detection	206
title		file & section names	376
HTML meta tag	131, 359	xindy	
titlepage		and hyperref	146
\subtitle and \published	406	customizing	156
		xparse	
		warnings	178

Troubleshooting Index

This index is a sorted reference of problems and solutions. In order to make it easier to locate a solution, the same issue may be addressed by more than one entry.

Entries with higher page numbers are often duplicates of entries with lower page numbers, as the same warning may occur within the user manual and again within the source code for a given package.

A	B	C
abstract		MATHJAX 661
missing toc 143, 616		
acronym		<i>Calibre</i>
multiply-defined labels 621		EPUB conversion 184
\AddSubtitlePublished 407		caption
affiliation 399		numbering 173
alt tags 162, 198		options 173
AMSmath		cases
ntheorem		MATHJAX 668
numbering 162, 850		chemfig 177
appendix		chemformula
incorrect toc link 143, 642		MATHJAX 177, 672
array		chemgreek
chemformula 177		fontspec mapping 677
authblk		text-mode symbols 677
\theauthor 398, 399		chemmacros
titling 143, 398, 651, 953		\makepolymerdelims 678
author		redox reactions 678
affiliation 399		colortbl 170, 425, 700
formatting 130		cross reference
		incorrect link 199
		missing 199
babel		CSS 200
French 176		
backref		D
. 652		ditaa 183
backref		duplicate filename 368
. 140		
bibliography		E
HTML page and toc 142		endnotes
bibtex		HTML page and toc 142
\etalchar 144		\ensuremath 198
Improper \prevdepth 144		epstopdf 166, 718
bigdelim		EPUB
. 172, 654		encoding 185
boxes		
. 135		
breqn		
darray 661		

page order	184	frames	136
search order	184		
section breaks	185		
\etalchar	144	G	
		glossaries	
		HTML page and toc	142
		makeglossaries not found . . .	144, 761
		page numbers	145, 762
		style	145, 762
		graphics	
		\rotatebox, \scalebox, \reflectbox	165, 569
		image format priorities	164, 569
		images out of order	103, 200
		optional arguments	164, 569
		scaled	764
		viewport	164, 569
		Greek packages	677
		grffile	165, 777
F		H	
fancybox		horizontal spacing	173, 921
\VerbatimFootnotes . . .	141, 723, 730	HTML	
fancyvrb		&, <, >	133, 588
\VerbatimFootnotes . . .	141, 723, 730	alt tags	162
figure		author	360
macro in name	198	corrupted	130, 135, 197, 200
file		entities	133, 588
inaccessible	122	image appear as HTML code .	103, 200
filename		inaccessible pages	122, 124
accents	376	starred section	142
duplicate	368	missing pages	
Missing \$ inserted	197	filename not unique	123
fixme	177, 738	recompile	200
float		SideTOCDepth and FileDepth .	122
alignment	133, 172	SideTOCDepth and tocdepth .	123
out of sequence	199	page did not update	123, 200
floatrow		settings	
\FBwidth and \FBheight . . .	173, 744	changed	120
with subfig	173, 744	undefined	197
floatrow		\HTMLAuthor	360
\ttabbox	168, 423	HTMLIndexCmd	
font		filenames	118
ligatures	114	hyperref	
missing symbols	134, 933	backref	140
monospace	114	comments between arguments .	140
package conflicts	114	incorrect link	199
selection	113		
UTF-8	113		
fontspec	113		
footnote			
displaymath	141, 723, 730	I	
in math	141	image	
in sectioning command	141	appears as HTML code	103, 200
memoir	141, 360		
misnumbered	141		
paragraph tags	141, 723, 730		
sectioning, footmisc	141, 360		
sectioning, verbatim	141, 723, 730		
title	398		
verbatim	141		
\VerbatimFootnotes . . .	141, 723, 730		
forest	755		
formatting			
\bfseries etc.	133, 588		

format priorities	164, 569	<i>makeglossaries</i>	
incorrect	103, 199, 200	not found	144, 761
viewport	164, 569	\makelabel	416
Improper \prevdepth		\marginpar	135, 366
bibtex	144	math	
boxes	198	Tikz	162
\includegraphics		alt tags	198
optional arguments	164, 569	appears as HTML code	103, 200
index		chemformula	177
HTML page and toc	142	custom macros	198
empty	157	dynamic	331
empty link	139, 199	equation numbering	
missing entries	146	ntheorem	162, 850
reference ranges	157	footnote	141
see and seealso	157	in T _E X boxes	160, 526
styling references	157	incorrect	103, 200, 331
xindy and hyperref	157	non-math contents	198
		slow or failed compile	198
		subequations	161, 527
		tabbing	168, 414
		MATHJAX	
		breqn	661
		cases	668
		chemformula	177, 672
		errors	162, 198, 331
		mhchem	831
		siunitx	161, 163, 528, 567, 910
		subequations	161, 527
		unsupported packages	161, 198, 528
		memoir	
		options clash	174, 1015
		mhchem	
		MATHJAX	831
		nested dollar signs	831
		minipage	
		alignment	136, 579
		horizontal space between	600
		in a span	135, 578
		inline	135, 578
		Misplaced \noalign	171, 810
		tabular	
		rules	170, 425
		Misplaced \omit	
		tabular	246
		Misplaced alignment tab character &	
		floatrow	173, 744
		frames	137
		supertabular	171, 927, 1008
		tabular	
		macros	168, 330, 423
		M	
MAC OS	128, 222		

Missing \$ inserted	
filename or URL	197
morewrites	197
morewrites	157
MS-WINDOWS	128, 222
multiply-defined labels	
acronym	621
\multirow and \multicolumn	169, 424, 840
multirow	169, 424
N	
newclude	176
newtxmath	163
nicefrac	163
No room for a new \write	197
ntheorem	850
font	162, 850
numbering	162, 850
O	
operating system	128, 222
options clash	
memoir	174, 1015
overpic	168, 863
P	
page	
inaccessible	122
page counter	
references	139
SVG images	103, 200
page numbers	140, 199, 574
pdftricks	166, 873
perltex	181
pfnote	
numbering	141, 874
polyglossia	
Undefined control sequence	176
PrintIndexCmd	118
psfrag	167
pstool	167, 879
pstricks	166, 880
pythontex	182
R	
reference	
% character between arguments	199
empty link	139, 199
incorrect link	199
label characters	139, 199
S	
sample_projects.css	
overwritten	127
sectioning	
accents	376
duplicate name	368
macro in name	198
missing	122
starred section	142
word processor import	189
siunitx	
MATHJAX	163, 567, 910
with TeXMaths	189
siunitx	
MATHJAX	161, 528
S column	171, 425
tabular S and s columns	163, 567, 910
splitidx	
\thepage and \AtWriteToIndex	148, 918
subfig	
options	173, 921
with floatrow	173, 744
SVG image	
appears as HTML code	103, 200
incorrect	103, 199, 200
math incorrect	331
out of order	103, 200
sympytex	182
T	
tabbing	
math	168, 414
table	
macro in name	198
numbering	814
Table of Contents	
missing	124
tabular	
* column specification	170, 424
baselines	442
corrupt rows	168, 423, 478
\endhead, \endfoot, \endlastfoot	171, 810
inside an environment	168, 423, 478
\kill	171, 811

macro inside	170, 424	Unicode	
Misplaced \noalign	170, 425	fonts	113
Misplaced alignment tab character &		UTF-8 locale	182
macros	168, 423	units	163
rules	330	UNIX	128, 222
\multicolumn with \multirow	841	URL	
\multirow	482, 483	Missing \$ inserted	197
multirow	169, 424	UTF-8	
\newcolumntype	170, 425	locale	182
numbering	1005		
row corruption	170, 424	V	
rules	170, 425	verbatim	
S column	171, 425	footnote	141
TeXMaths	189	framed	138, 723
text		VerbatimFootnotes	141, 723, 730
&, <, >	133, 588	verse	
\bfseries etc.	133, 588	spacing	175, 410, 980
corrupted	113	viewport	164, 569
textcomp			
missing symbols	134, 933	W	
\theauthor and authblk	398, 399	warpall	197
tikz		warpHTML	128, 197
in math	162	warpprint	129, 197
matrices, &	165, 944	WINDOWS	128, 222
title		word processor	
affiliation	399	import	186
newlines	130	sectioning headings	189
\thanks	398		
titling		X	
authblk	143, 398, 651, 953	xeL ^A TEX	113
hooks	407	xfrac	1002
tocloft		xindy	
chapter titles	143, 159, 641, 961	hyperref	157
todonotes	177, 815, 969	options	
tracing l warp	201	HTMLIndexCmd	118
tram	971	LateXmkIndexCmd	119
transparent	971	PrintIndexCmd	118
		xltabular	
		numbering	1005
U		xparse	178
Undefined control sequence		xy	1010
polyglossia	176		

Index of Indexes

C		I	
Change History	1043	Index of Objects	1070
G		T	
General Index	1091	Troubleshooting Index	1095