



# The **lwarp** package

L<sup>A</sup>T<sub>E</sub>X to HTML5

v0.36 — 2017/08/17

© 2016–2017 Brian Dunn  
bd@BDTechConcepts.com

## Abstract

The **lwarp** package allows L<sup>A</sup>T<sub>E</sub>X to directly produce HTML5 output, using external utility programs only for the final conversion of text and images. Math may be represented by SVG files or MathJax.

Documents may be produced by pdfL<sup>A</sup>T<sub>E</sub>X, LuaL<sup>A</sup>T<sub>E</sub>X, or XeL<sup>A</sup>T<sub>E</sub>X. A **texlua** script removes the need for system utilities such as **make** and **gawk**, and also supports **xindy** and **latexmk**. Configuration is automatic at the first manual compile.

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

A modular package-loading system uses the **lwarp** version of a package for HTML when available. More than a hundred L<sup>A</sup>T<sub>E</sub>X packages are supported with these high-level source compatibility replacements, and many others work as-is.

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

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

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

## License:

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

# Contents

<b>1</b>	<b>lwarp.sty</b>	<b>23</b>
<b>1</b>	<b>Updates</b>	<b>23</b>
<b>2</b>	<b>Introduction</b>	<b>28</b>
2.1	Supported packages and features . . . . .	29
<b>3</b>	<b>Alternatives</b>	<b>34</b>
3.1	Internet class . . . . .	34
3.2	TeX4ht . . . . .	34
3.3	Translators . . . . .	34
3.4	AsciiDoc . . . . .	35
3.5	Pandoc . . . . .	35
3.6	Word processors . . . . .	35
3.7	Commercial systems . . . . .	36
3.8	Comparisons . . . . .	36
<b>4</b>	<b>Installation</b>	<b>37</b>
4.1	Installing the lwarp package . . . . .	37
4.2	Installing the lwarpmk utility . . . . .	40
4.2.1	Using a local copy of lwarpmk . . . . .	41
4.3	Installing additional utilities . . . . .	42
<b>5</b>	<b>Tutorial</b>	<b>45</b>
5.1	Starting a new project . . . . .	45
5.2	Compiling the print version with lwarpmk . . . . .	49
5.3	Compiling the HTML version with lwarpmk . . . . .	50
5.4	Generating the SVG images . . . . .	51
5.5	Using MathJax for math . . . . .	52
5.6	Changing the CSS style . . . . .	53
5.7	Customizing the HTML output . . . . .	54
5.8	Using latexmk . . . . .	59
5.9	Using XeLaTeX or LuaLaTeX . . . . .	60
5.10	Using a glossary . . . . .	61
5.11	Cleaning auxiliary files . . . . .	62
5.12	Cleaning auxiliary and output files . . . . .	62
5.13	Processing multiple projects in the same directory . . . . .	62
5.14	Using the make utility . . . . .	62
<b>6</b>	<b>Additional details</b>	<b>63</b>
6.1	Font and UTF-8 support . . . . .	63
6.1.1	Indexes and UTF-8 . . . . .	64
6.2	lwarp package loading and options . . . . .	64
6.3	Selecting the operating system . . . . .	66

6.4	Selecting actions for print or HTML output . . . . .	66
6.5	Commands to be placed into the <code>warpprint</code> environment . . . . .	67
6.6	Title page . . . . .	67
6.7	HTML page meta descriptions . . . . .	68
6.8	HTML page meta author . . . . .	69
6.9	CSS . . . . .	69
6.10	Modifying <code>xindy</code> index processing . . . . .	70
<b>7</b>	<b>Special cases and limitations</b>	<b>70</b>
7.1	Formatting . . . . .	71
7.1.1	Text formatting . . . . .	71
7.1.2	Horizontal space . . . . .	71
7.1.3	Text alignment . . . . .	71
7.1.4	Superscripts and other non-math uses of math mode . . . . .	71
7.1.5	Empty <code>\item</code> followed by a new line of text or a nested list: . . . . .	71
7.1.6	Filenames in lists . . . . .	71
7.2	Boxes and minipages . . . . .	72
7.2.1	Save Boxes . . . . .	72
7.2.2	Minipages . . . . .	72
7.2.3	Side-by-side minipages . . . . .	73
7.2.4	Framed minipages and other environments . . . . .	73
7.2.5	<code>fancybox</code> package . . . . .	74
7.2.6	<code>mdframed</code> package . . . . .	75
7.3	Cross-references . . . . .	76
7.3.1	Page references . . . . .	76
7.3.2	<code>cleveref</code> and <code>varioref</code> packages . . . . .	76
7.3.3	Hyperlinks, <code>hyperref</code> , and <code>url</code> . . . . .	76
7.3.4	Footnotes and page notes . . . . .	77
7.4	Math . . . . .	77
7.4.1	<code>ntheorem</code> package . . . . .	79
7.4.2	<code>siunitx</code> package . . . . .	79
7.4.3	<code>units</code> and <code>nicefrac</code> packages . . . . .	79
7.4.4	<code>newtxmath</code> package . . . . .	79
7.5	Graphics . . . . .	79
7.5.1	<code>xcolor</code> package . . . . .	80
7.6	Tabular . . . . .	81
7.6.1	<code>longtable</code> package . . . . .	82
7.6.2	<code>float</code> , <code>trivfloat</code> , and/or <code>algorithmicx</code> together . . . . .	82
7.6.3	<code>caption</code> and <code>subcaption</code> packages . . . . .	83
7.7	Floats . . . . .	83
7.7.1	<code>subfig</code> package . . . . .	83
7.7.2	<code>floatrow</code> package . . . . .	83
7.8	Miscellaneous . . . . .	84
7.8.1	<code>abstract</code> package . . . . .	84
7.8.2	<code>verse</code> and <code>memoir</code> . . . . .	84
7.8.3	<code>newclude</code> package . . . . .	84

7.8.4	babel package . . . . .	85
7.8.5	glossaries package . . . . .	85
<b>8</b>	<b>Epub conversion</b>	<b>86</b>
<b>9</b>	<b>Word-processor conversion</b>	<b>88</b>
<b>10</b>	<b>Modifying lwarp</b>	<b>89</b>
10.1	Creating an lwarp version of a package . . . . .	89
10.2	Testing lwarp . . . . .	90
10.3	Modifying lwarpmk . . . . .	90
<b>11</b>	<b>Troubleshooting</b>	<b>91</b>
11.1	Using the lwarp.sty package . . . . .	91
11.1.1	Debug tracing output . . . . .	93
11.2	Compiling the lwarp.dtx file . . . . .	93
<b>12</b>	<b>Implementation</b>	<b>94</b>
<b>13</b>	<b>Stack depths</b>	<b>94</b>
<b>14</b>	<b>Source Code</b>	<b>96</b>
<b>15</b>	<b>Detecting the T<sub>E</sub>X Engine — pdf<sub>l</sub>atex, lua<sub>l</sub>atex, xel<sub>l</sub>atex</b>	<b>97</b>
<b>16</b>	<b>Unicode Input Characters</b>	<b>97</b>
<b>17</b>	<b>Early package requirements</b>	<b>98</b>
<b>18</b>	<b>Operating-System portability</b>	<b>99</b>
18.1	Common portability code . . . . .	99
18.2	Unix, Linux, and Mac OS . . . . .	99
18.3	MS-Windows . . . . .	99
<b>19</b>	<b>Package options</b>	<b>100</b>
<b>20</b>	<b>Misplaced packages</b>	<b>103</b>
<b>21</b>	<b>Required packages</b>	<b>107</b>
<b>22</b>	<b>Loading packages</b>	<b>111</b>
<b>23</b>	<b>Copying a file</b>	<b>114</b>
<b>24</b>	<b>Debugging messages</b>	<b>114</b>
<b>25</b>	<b>HTML-conversion output modifications</b>	<b>115</b>
<b>26</b>	<b>Remembering original formatting macros</b>	<b>116</b>

<b>27</b>	<b>Textcomp and Slunitx</b>	<b>118</b>
<b>28</b>	<b>Configuration Files</b>	<b>119</b>
28.1	project_html.tex . . . . .	119
28.2	lwarpmk.conf . . . . .	119
28.3	project.lwarpmkconf . . . . .	120
28.4	lwarp.css . . . . .	121
28.5	lwarp_sagebrush.css . . . . .	146
28.6	lwarp_formal.css . . . . .	151
28.7	sample_project.css . . . . .	155
28.8	lwarp.xdy . . . . .	156
28.9	lwarp_mathjax.txt . . . . .	156
28.10	lwarpmk option . . . . .	158
<b>29</b>	<b>Stacks</b>	<b>169</b>
29.1	Assigning depths . . . . .	169
29.2	Closing actions . . . . .	170
29.3	Closing depths . . . . .	170
29.4	Pushing and popping the stack . . . . .	171
<b>30</b>	<b>Data arrays</b>	<b>172</b>
<b>31</b>	<b>HTML entities</b>	<b>173</b>
<b>32</b>	<b>HTML filename generation</b>	<b>174</b>
<b>33</b>	<b>Homepage link</b>	<b>176</b>
<b>34</b>	<b>\PrintStack diagnostic tool</b>	<b>177</b>
<b>35</b>	<b>Closing stack levels</b>	<b>178</b>
<b>36</b>	<b>PDF pages and styles</b>	<b>178</b>
<b>37</b>	<b>HTML tags, spans, divs, elements</b>	<b>179</b>
37.1	Mapping L <sup>A</sup> T <sub>E</sub> X Sections to HTML Sections . . . . .	179
37.2	Babel-French . . . . .	179
37.3	HTML tags . . . . .	181
37.4	Block tags and comments . . . . .	183
37.5	Div class and element class . . . . .	184
37.6	Single-line elements . . . . .	185
37.7	HTML5 semantic elements . . . . .	186
37.8	High-level block and inline classes . . . . .	186
37.9	Closing HTML tags . . . . .	187
<b>38</b>	<b>Paragraph handling</b>	<b>188</b>
<b>39</b>	<b>Paragraph start/stop handling</b>	<b>191</b>

<b>40</b>	<b>Page headers and footers</b>	<b>194</b>
<b>41</b>	<b>CSS</b>	<b>195</b>
<b>42</b>	<b>HTML meta description and author</b>	<b>196</b>
<b>43</b>	<b>Footnotes</b>	<b>196</b>
43.1	Regular page footnotes . . . . .	197
43.2	Minipage footnotes . . . . .	197
43.3	Titlepage thanks . . . . .	197
43.4	Regular page footnote implementation . . . . .	197
43.5	Minipage footnote implementation . . . . .	199
<b>44</b>	<b>Marginpars</b>	<b>200</b>
<b>45</b>	<b>Splitting HTML files</b>	<b>200</b>
<b>46</b>	<b>Sectioning</b>	<b>206</b>
46.1	Book class commands . . . . .	206
46.2	Sectioning support macros . . . . .	206
46.3	\section and friends . . . . .	211
<b>47</b>	<b>Starting a new file</b>	<b>213</b>
<b>48</b>	<b>Starting HTML output</b>	<b>216</b>
<b>49</b>	<b>Ending HTML output</b>	<b>219</b>
<b>50</b>	<b>Titles and the titling package</b>	<b>221</b>
50.1	Setting the title, etc. . . . .	222
50.2	Changes to HTML titlepage and titlingpage . . . . .	224
50.3	Printing the title, etc. in HTML . . . . .	225
50.4	Printing the title, etc. in print form . . . . .	228
50.5	\maketitle for print output . . . . .	229
50.6	\maketitle for HTML output . . . . .	231
<b>51</b>	<b>Abstract</b>	<b>234</b>
<b>52</b>	<b>Quote and verse</b>	<b>235</b>
52.1	Citations and attributions . . . . .	235
52.2	Quotes, quotations . . . . .	236
52.3	Verse . . . . .	236
<b>53</b>	<b>Verbatim</b>	<b>237</b>
<b>54</b>	<b>Theorems</b>	<b>239</b>
<b>55</b>	<b>Lists</b>	<b>240</b>

55.1	Itemize . . . . .	241
55.2	Enumerate . . . . .	242
55.3	Description . . . . .	242
<b>56</b>	<b>Tabular</b>	<b>244</b>
56.1	Token lookahead . . . . .	245
56.2	Booleans . . . . .	246
56.3	Handling &, @, and ! . . . . .	247
56.3.1	Localizing & catcodes . . . . .	248
56.3.2	Handling & . . . . .	248
56.4	Handling \\ . . . . .	249
56.5	Variables . . . . .	250
56.6	Parsing @, >, <, ! columns . . . . .	251
56.7	Parsing 'l', 'c', or 'r' columns . . . . .	254
56.8	Parsing 'p', 'm', or 'b' columns . . . . .	254
56.9	Parsing 'D' columns . . . . .	255
56.10	Parsing the column specifications . . . . .	255
56.11	Starting a new row . . . . .	259
56.12	Printing at or bang tags . . . . .	259
56.13	Data opening tag . . . . .	260
56.14	Midrules . . . . .	261
56.15	Multicolumns . . . . .	262
56.15.1	Parsing multicolumns . . . . .	262
56.15.2	High-level multicolumn interface . . . . .	265
56.15.3	Longtable captions . . . . .	266
56.15.4	Counting HTML tabular columns . . . . .	269
56.15.5	\tabledatamulticolumntag . . . . .	270
56.16	Multirow . . . . .	270
56.17	Utility macros inside a table . . . . .	271
56.18	Checking for a new table cell . . . . .	271
56.19	\mrowcell . . . . .	273
56.20	New \tabular definition . . . . .	273
56.21	Array . . . . .	276
<b>57</b>	<b>Cross-references</b>	<b>276</b>
57.1	Setup . . . . .	277
57.2	Zref setup . . . . .	277
57.3	Labels . . . . .	280
57.4	References . . . . .	281
57.5	Hyper-references . . . . .	283
<b>58</b>	<b>Floats</b>	<b>286</b>
58.1	Float captions . . . . .	287
58.1.1	Caption inside a float environment . . . . .	288
58.1.2	Caption and LOF linking and tracking . . . . .	289

<b>59</b>	<b>Table of Contents, LOF, LOT</b>	<b>292</b>
59.1	Reading and printing the TOC . . . . .	292
59.2	High-level TOC commands . . . . .	295
59.3	Side TOC . . . . .	295
59.4	Low-level TOC line formatting . . . . .	296
<b>60</b>	<b>Index and glossary</b>	<b>299</b>
<b>61</b>	<b>Math</b>	<b>301</b>
61.1	Inline and display math . . . . .	303
61.2	MathJax support . . . . .	306
61.3	Equation environment . . . . .	309
61.4	AMS Math environments . . . . .	311
61.4.1	Support macros . . . . .	311
61.4.2	Environment patches . . . . .	312
<b>62</b>	<b>Lateximages</b>	<b>319</b>
<b>63</b>	<b>center, flushleft, flushright</b>	<b>325</b>
<b>64</b>	<b>Siunitx</b>	<b>326</b>
<b>65</b>	<b>Graphics</b>	<b>326</b>
65.1	<code>\graphicspath</code> . . . . .	327
65.2	Length conversions and graphics options . . . . .	328
65.3	<code>\includegraphics</code> . . . . .	331
65.4	<code>\rotatebox</code> , <code>\scalebox</code> , <code>\reflectbox</code> . . . . .	335
65.5	Null functions . . . . .	337
<b>66</b>	<b>Xcolor defaults</b>	<b>338</b>
66.1	Defaults without xcolor . . . . .	338
66.2	xcolor print-mode additions . . . . .	338
<b>67</b>	<b>Cleveref</b>	<b>342</b>
<b>68</b>	<b>Picture</b>	<b>343</b>
<b>69</b>	<b>Boxes and Minipages</b>	<b>344</b>
69.1	Counters and lengths . . . . .	344
69.2	Footnote handling . . . . .	345
69.3	Minipage handling . . . . .	345
69.4	<code>Parbox</code> , <code>makebox</code> , <code>framebox</code> , <code>fbox</code> , <code>raisebox</code> . . . . .	348
<b>70</b>	<b>Direct formatting</b>	<b>353</b>
<b>71</b>	<b>Skips, spaces, font sizes</b>	<b>358</b>
<b>72</b>	<b><code>\phantomsection</code></b>	<b>365</b>



---

<b>73</b>	<b>\LaTeX and other logos</b>	<b>365</b>
73.1	HTML logos . . . . .	366
73.2	Print logos . . . . .	367
<b>74</b>	<b>\AtBeginDocument, \AtEndDocument</b>	<b>368</b>
<b>75</b>	<b>Trademarks</b>	<b>369</b>
<b>2</b>	<b>lwarp-abstract.sty</b>	<b>370</b>
76	Abstract	370
<b>3</b>	<b>lwarp-afterpage.sty</b>	<b>372</b>
77	Afterpage	372
<b>4</b>	<b>lwarp-algorithmicx.sty</b>	<b>373</b>
78	Algorithmicx	373
<b>5</b>	<b>lwarp-alltt.sty</b>	<b>373</b>
79	Alltt	374
<b>6</b>	<b>lwarp-amsthm.sty</b>	<b>375</b>
80	AMSthm	375
<b>7</b>	<b>lwarp-balance.sty</b>	<b>378</b>
81	Balance	378
<b>8</b>	<b>lwarp-bookmark.sty</b>	<b>379</b>
82	Bookmark	379
<b>9</b>	<b>lwarp-booktabs.sty</b>	<b>379</b>
83	Booktabs	379

---

<b>10</b>	<b>lwarp-boxedminipage.sty</b>	<b>380</b>
84	Boxedminipage	380
<b>11</b>	<b>lwarp-boxedminipage2e.sty</b>	<b>380</b>
85	Boxedminipage2e	380
<b>12</b>	<b>lwarp-ccaption.sty</b>	<b>381</b>
86	Ccaption	381
<b>13</b>	<b>lwarp-changepage.sty</b>	<b>381</b>
87	Changepage	381
<b>14</b>	<b>lwarp-crop.sty</b>	<b>382</b>
88	Crop	382
<b>15</b>	<b>lwarp-cutwin.sty</b>	<b>383</b>
89	Cutwin	383
<b>16</b>	<b>lwarp-dcolumn.sty</b>	<b>383</b>
90	Dcolumn	384
<b>17</b>	<b>lwarp-draftwatermark.sty</b>	<b>385</b>
91	Draftwatermark	385
<b>18</b>	<b>lwarp-ellipsis.sty</b>	<b>385</b>
92	Ellipsis	385
<b>19</b>	<b>lwarp-emptypage.sty</b>	<b>386</b>
93	Emptypage	386

---

<b>20</b>	<b>lwarp-endnotes.sty</b>	<b>387</b>
94	Endnotes	387
<b>21</b>	<b>lwarp-enumerate.sty</b>	<b>388</b>
95	Enumerate	388
<b>22</b>	<b>lwarp-enumitem.sty</b>	<b>389</b>
96	Enumitem	389
<b>23</b>	<b>lwarp-epigraph.sty</b>	<b>390</b>
97	Epigraph	390
<b>24</b>	<b>lwarp-eso-pic.sty</b>	<b>391</b>
98	Eso-pic	391
<b>25</b>	<b>lwarp-everypage.sty</b>	<b>392</b>
99	Everypage	392
<b>26</b>	<b>lwarp-everyshi.sty</b>	<b>393</b>
100	Everyshi	393
<b>27</b>	<b>lwarp-extramarks.sty</b>	<b>393</b>
101	Extramarks	393
<b>28</b>	<b>lwarp-fancybox.sty</b>	<b>394</b>
102	Fancybox	394
<b>29</b>	<b>lwarp-fancyhdr.sty</b>	<b>397</b>
103	Fancyhdr	397

---

<b>30</b>	<b>lwarp-fancyvrb.sty</b>	<b>399</b>
104	Fancyvrb	399
<b>31</b>	<b>lwarp-figcaps.sty</b>	<b>405</b>
105	Figcaps	405
<b>32</b>	<b>lwarp-float.sty</b>	<b>406</b>
106	Float and \newfloat	406
<b>33</b>	<b>lwarp-floatflt.sty</b>	<b>408</b>
107	Floatflt	408
<b>34</b>	<b>lwarp-floatpag.sty</b>	<b>409</b>
108	Floatpag	409
<b>35</b>	<b>lwarp-floatrow.sty</b>	<b>410</b>
109	Floatrow	410
<b>36</b>	<b>lwarp-flushend.sty</b>	<b>416</b>
110	Flushend	416
<b>37</b>	<b>lwarp-fontenc.sty</b>	<b>417</b>
111	Fontenc	417
<b>38</b>	<b>lwarp-fontspec.sty</b>	<b>417</b>
112	Fontspec	417
<b>39</b>	<b>lwarp-footmisc.sty</b>	<b>418</b>
113	Footmisc	418

---

<b>40</b>	<b>lwarp-footnote.sty</b>	<b>420</b>
114	Footnote	420
<b>41</b>	<b>lwarp-footnotehyper.sty</b>	<b>421</b>
115	Footnotehyper	421
<b>42</b>	<b>lwarp-framed.sty</b>	<b>422</b>
116	Framed	422
<b>43</b>	<b>lwarp-ftnright.sty</b>	<b>425</b>
117	Ftnright	425
<b>44</b>	<b>lwarp-fullpage.sty</b>	<b>425</b>
118	Fullpage	425
<b>45</b>	<b>lwarp-geometry.sty</b>	<b>426</b>
119	Geometry	426
<b>46</b>	<b>lwarp-glossaries.sty</b>	<b>427</b>
120	Glossaries	427
<b>47</b>	<b>lwarp-graphics.sty</b>	<b>428</b>
121	Graphics	428
<b>48</b>	<b>lwarp-graphicx.sty</b>	<b>428</b>
122	Graphicx	428
<b>49</b>	<b>lwarp-hyperref.sty</b>	<b>429</b>
123	Hyperref	429

---

<b>50</b>	<b>lwarp-hyperxmp.sty</b>	<b>432</b>
124	Hyperxmp	432
<b>51</b>	<b>lwarp-idxlayout.sty</b>	<b>433</b>
125	Idxlayout	433
<b>52</b>	<b>lwarp-indentfirst.sty</b>	<b>434</b>
126	Indentfirst	434
<b>53</b>	<b>lwarp-inputenc.sty</b>	<b>434</b>
127	Inputenc	434
<b>54</b>	<b>lwarp-keyfloat.sty</b>	<b>435</b>
128	Keyfloat	435
<b>55</b>	<b>lwarp-layout.sty</b>	<b>437</b>
129	Layout	437
<b>56</b>	<b>lwarp-letterspace.sty</b>	<b>437</b>
130	Letterspace	437
<b>57</b>	<b>lwarp-lettrine.sty</b>	<b>438</b>
131	Lettrine	438
<b>58</b>	<b>lwarp-lips.sty</b>	<b>439</b>
132	Lips	439
<b>59</b>	<b>lwarp-listings.sty</b>	<b>440</b>
133	Listings	440

---

<b>60</b>	<b>lwarp-longtable.sty</b>	<b>444</b>
134	Longtable	444
<b>61</b>	<b>lwarp-lscape.sty</b>	<b>446</b>
135	Lscape	446
<b>62</b>	<b>lwarp-ltcaption.sty</b>	<b>446</b>
136	Ltcaption	446
<b>63</b>	<b>lwarp-marginfit.sty</b>	<b>447</b>
137	Marginfit	447
<b>64</b>	<b>lwarp-marginfix.sty</b>	<b>447</b>
138	Marginfix	447
<b>65</b>	<b>lwarp-marginnote.sty</b>	<b>448</b>
139	Marginnote	448
<b>66</b>	<b>lwarp-mcaption.sty</b>	<b>448</b>
140	Mcaption	448
<b>67</b>	<b>lwarp-mdframed.sty</b>	<b>449</b>
141	Mdframed	449
<b>68</b>	<b>lwarp-microtype.sty</b>	<b>458</b>
142	Microtype	458
<b>69</b>	<b>lwarp-moreverb.sty</b>	<b>459</b>
143	Moreverb	459

---

<b>70</b>	<b>lwarp-mparhack.sty</b>	<b>461</b>
144	Mparhack	461
<b>71</b>	<b>lwarp-multicol.sty</b>	<b>462</b>
145	Multicol	462
<b>72</b>	<b>lwarp-multirow.sty</b>	<b>464</b>
146	Multirow	464
<b>73</b>	<b>lwarp-nameref.sty</b>	<b>464</b>
147	Nameref	464
<b>74</b>	<b>lwarp-needspace.sty</b>	<b>465</b>
148	Needspace	465
<b>75</b>	<b>lwarp-newclude.sty</b>	<b>465</b>
149	Newclude	465
<b>76</b>	<b>lwarp-newunicodechar.sty</b>	<b>466</b>
150	Newunicodechar	466
<b>77</b>	<b>lwarp-nextpage.sty</b>	<b>466</b>
151	Nextpage	466
<b>78</b>	<b>lwarp-nowidow.sty</b>	<b>466</b>
152	Nowidow	467
<b>79</b>	<b>lwarp-ntheorem.sty</b>	<b>468</b>
153	Ntheorem	468



---

<b>80</b>	<b>lwarp-pagenote.sty</b>	<b>479</b>
154	Pagenote	479
<b>81</b>	<b>lwarp-paralist.sty</b>	<b>479</b>
155	Paralist	479
<b>82</b>	<b>lwarp-parskip.sty</b>	<b>480</b>
156	Parskip	480
<b>83</b>	<b>lwarp-pdfscape.sty</b>	<b>480</b>
157	Pdfscape	480
<b>84</b>	<b>lwarp-pdfsync.sty</b>	<b>481</b>
158	Pdfsync	481
<b>85</b>	<b>lwarp-placeins.sty</b>	<b>481</b>
159	Placeins	481
<b>86</b>	<b>lwarp-prelim2e.sty</b>	<b>482</b>
160	Prelim2e	482
<b>87</b>	<b>lwarp-ragged2e.sty</b>	<b>483</b>
161	Ragged2e	483
<b>88</b>	<b>lwarp-rotating.sty</b>	<b>484</b>
162	Rotating	484
<b>89</b>	<b>lwarp-rotfloat.sty</b>	<b>485</b>
163	Rotfloat	485

---

<b>90</b>	<b>lwarp-savetrees.sty</b>	<b>486</b>
164	Savetrees	486
<b>91</b>	<b>lwarp-setspace.sty</b>	<b>487</b>
165	Setspace	487
<b>92</b>	<b>lwarp-shadow.sty</b>	<b>488</b>
166	Shadow	488
<b>93</b>	<b>lwarp-showidx.sty</b>	<b>489</b>
167	Showidx	489
<b>94</b>	<b>lwarp-showkeys.sty</b>	<b>489</b>
168	Showkeys	489
<b>95</b>	<b>lwarp-sidecap.sty</b>	<b>490</b>
169	Sidecap	490
<b>96</b>	<b>lwarp-sidenotes.sty</b>	<b>491</b>
170	Sidenotes	491
<b>97</b>	<b>lwarp-soul.sty</b>	<b>493</b>
171	Soul	493
<b>98</b>	<b>lwarp-subfig.sty</b>	<b>495</b>
172	Subfig	495
<b>99</b>	<b>lwarp-syntonly.sty</b>	<b>501</b>
173	Syntonly	501

---

<b>100</b>	<b>lwarp-tabularx.sty</b>	<b>502</b>
174	Tabularx	502
<b>101</b>	<b>lwarp-tabulary.sty</b>	<b>503</b>
175	Tabulary	503
<b>102</b>	<b>lwarp-textpos.sty</b>	<b>504</b>
176	Textpos	504
<b>103</b>	<b>lwarp-theorem.sty</b>	<b>505</b>
177	Theorem	505
<b>104</b>	<b>lwarp-threeparttable.sty</b>	<b>509</b>
178	Threeparttable	509
<b>105</b>	<b>lwarp-tikz.sty</b>	<b>510</b>
179	Tikz	510
<b>106</b>	<b>lwarp-titleps.sty</b>	<b>512</b>
180	Titleps	512
<b>107</b>	<b>lwarp-titleref.sty</b>	<b>514</b>
181	Titleref	514
<b>108</b>	<b>lwarp-titlesec.sty</b>	<b>515</b>
182	Titlesec	515
<b>109</b>	<b>lwarp-titletoc.sty</b>	<b>516</b>
183	Titletoc	516

<b>110</b>	<b>lwarp-titling.sty</b>	<b>517</b>
184	Titling	517
<b>111</b>	<b>lwarp-tocloft.sty</b>	<b>518</b>
185	Tocloft	518
<b>112</b>	<b>lwarp-transparent.sty</b>	<b>523</b>
186	Transparent	523
<b>113</b>	<b>lwarp-trivfloat.sty</b>	<b>524</b>
187	Trivfloat	524
187.1	Combining <code>\newfloat</code> , <code>\trivfloat</code> , and <code>algorithmicx</code> . . . . .	524
<b>114</b>	<b>lwarp-ulem.sty</b>	<b>526</b>
188	Ulem	526
<b>115</b>	<b>lwarp-upref.sty</b>	<b>528</b>
189	Upref	528
<b>116</b>	<b>lwarp-verse.sty</b>	<b>529</b>
190	Verse	529
<b>117</b>	<b>lwarp-wallpaper.sty</b>	<b>531</b>
191	Wallpaper	531
<b>118</b>	<b>lwarp-wrapfig.sty</b>	<b>532</b>
192	Wrapfig	532

---

<b>119</b>	<b>lwarp-xcolor.sty</b>	<b>534</b>
<b>193</b>	<b>Xcolor</b>	<b>534</b>
193.1	Xcolor definitions: location and timing . . . . .	534
193.2	Code . . . . .	536
<b>120</b>	<b>lwarp-xfrac.sty</b>	<b>543</b>
<b>194</b>	<b>Xfrac</b>	<b>543</b>
<b>121</b>	<b>lwarp-xmpincl.sty</b>	<b>546</b>
<b>195</b>	<b>Xmpincl</b>	<b>546</b>
	<b>Change History and Index</b>	<b>547</b>

## List of Figures

1	tutorial.tex listing . . . . .	46
---	--------------------------------	----

## List of Tables

1	L <sup>A</sup> T <sub>E</sub> X–HTML generation — lwarp package — Supported functions	29
2	Additional supported packages . . . . .	33
3	Required software programs . . . . .	38
4	Files created along with the print version . . . . .	48
5	Package options . . . . .	65
6	Section depths and HTML headings . . . . .	95
7	Tabular baseline . . . . .	256
8	Tabular HTML column conversions . . . . .	256
9	Cross-referencing data structures . . . . .	278
10	Float data structures . . . . .	286

## Package 1

# lwarp.sty

## 1 Updates

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

For a detailed list of changes, see the Change History on page 547.

### v0.36:

- Recorganized the documentation section regarding special cases and limitations. (Section 7)
- Improved source formatting.
- boxes and frames
  - `\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  $\text{\LaTeX}$  standards, and avoids problems with space between arguments.

## ⚠ Optional arguments

spans with minipages

framing minipages

lateximage, SVG math,  
tabular

eqnarray

verbatim packages

framing packages

list packages

babel-french

- Likewise, `\InlineClass`'s optional argument now comes before the mandatory arguments:

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

- Improved compatibility between spans, minipages, lists, frames, and math. Handles minipages and lists inside an HTML span, such as an `\fbox` containing a minipage, although with minimal HTML formatting. See section 7.2.2. `\fboxBlock` is added to frame minipages, tables, and lists with full HTML formatting but no longer inline, and behaves as `\fbox` for print output. The `fminipage` environment is added for framed minipages, as an environment with full HTML formatting, and draws a framed minipage in print output. See section 7.2.4. `\fbox` and minipages now often work in SVG math and `lateximages`. MathJax supports `\fbox`, but not `\fboxBlock` nor `fminipage`.
- Improved compatibility between `lateximage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, `booktabs`.
- Improved font control for `lateximage`s and SVG math.
- Added the `eqnarray` environments.
- `fancyvrb` is no longer required (preloaded), but is still supported.
- Added `verbatim` and `moreverb`.
- Added `fancybox`, `boxedminipage2e` and `shadow`.
- `enumitem` is no longer required, but is still supported.
- Added `enumerate` and `paralist`.
- `titleps` is no longer required, but is still supported.
- Added `crop`.
- Added `rotfloat`, `marginfit`, and several minor packages; see the change log.
- Adds fixed-width HTML spaces around punctuation when using `babel-french`. Lua<sub>T</sub><sub>E</sub>X does not yet use the extra punctuation spacing.

### v0.33:

- Tabular `@` and `!` columns now have their own HTML columns.
- `&` catcode changes are localized, perhaps causing errors about the tab alignment character `&`, so any definitions of macros or environments which themselves contain `tabular` and `&` must be enclosed within `\StartDefiningTabulars` and `\EndDefiningTabulars`. See section 56.3.1. This change is not required for the routine use of tables, but only when a table is defined inside another macro or environment, and while also using the `&` character inside the definition. This may include the use inside conditional expressions.



- Several math environments were incorrectly placed inline. Also, for `amsmath` with SVG math, the `fleqn` option has been removed, resulting in improved spacing for aligned equations.
- Bug fixes; see the changelog.

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

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

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

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

**v0.30:**

 **lwarp-newproject**

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

To modify existing documents, remove from the document source:

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

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

- 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:

 **HTML setup changes.**

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

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

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

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

```

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

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

**v0.29:**

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

**v0.28:**

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

## 2 Introduction

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

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

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

SVG images may be used for math, and are also used for `picture`, and `Tikz` environments, as this format has better browser and e-book support than MathML (as of this writing), while still allowing for the high-quality display and printing of images (again, subject to potentially bug-ridden<sup>1</sup> 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  $\text{\LaTeX}$  source for the expression, allowing it to be copy/pasted into other documents.<sup>2</sup> Custom  $\text{\LaTeX}$  macros may be used as-is in math expressions, since the math is evaluated entirely inside  $\text{\LaTeX}$ .

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

---

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

---

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

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

a L<sup>A</sup>T<sub>E</sub>X-generated PDF file which happens to have HTML5 tags, to a number of HTML5 plain-text files and accompanying images.

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

When requesting packages during HTML conversion, `lwarp` first looks to see if it has its own modified version to use instead of the usual L<sup>A</sup>T<sub>E</sub>X version. These `lwarp-packageName.sty` files contain code used to emulate or replace functions for HTML output.

Enough functionality is provided to convert a typical article containing technical content. Not every package has been tested, but many of the most useful ones are known to work, either as-is or through emulation with replacement code. (See table 1 on page 29.)

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

## 2.1 Supported packages and features

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

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

Table 1: L<sup>A</sup>T<sub>E</sub>X–HTML generation — `lwarp` package — Supported functions

Category	Status
Engines:	pdfL <sup>A</sup> T <sub>E</sub> X, X <sub>Y</sub> L <sup>A</sup> T <sub>E</sub> X, LuaL <sup>A</sup> T <sub>E</sub> X
Classes:	<code>book</code> , <code>report</code> , or <code>article</code> . <code>memoir</code> is planned.

## lwarp Supported Functions — continued

Category	Status
Sectioning:	Supported, with hyperlinks. Honors <code>tocdepth</code> and <code>secnumdepth</code> . Adds <code>filedepth</code> for splitting the HTML output. Files may be numbered sequentially or named according to section name. Common short words and punctuation is removed from the filenames. Emulates <code>titlesec</code> .
Page layout:	Emulates <code>fancyhdr</code> , <code>titleps</code> .
Table of contents, figures, tables:	Supported, with hyperlinks. Emulates <code>titletoc</code> and <code>tocloft</code> .
Title page:	<code>\maketitle</code> , <code>titlepage</code> , <code>titling</code> . Optional <code>titling</code> -based commands for published and subtitle.
<code>abstract</code> :	Supported
Cross-references:	Emulated, with hyperlinks.
<code>hyperref</code> :	Emulated. HTML hyperlinks are generated for TOC, LOF, LOT, <code>\nameref</code> , <code>\ref</code> , the <code>cleveref</code> commands, and index entries.
<code>url</code> :	Supported, without hyperlinks so far.
Margin notes:	<code>marginfit</code> , <code>marginfix</code> .
Footnotes:	<code>footnote</code> , <code>footmisc</code> , <code>marginnote</code> , <code>sidenote</code> , <code>pagenote</code> , <code>endnotes</code> .
Indexing:	<code>texindy</code> is used, with hyperlinks. <code>idxlayout</code> is emulated.
Glossary:	<code>glossaries</code> and <code>xindy</code> are used.
Bibliography:	Supported, without hyperlinks so far.
Math:	Supported. Converted to SVG images with HTML <code>ALT</code> tags containing the $\text{\LaTeX}$ source for the math expression. MathJax supported as an alternative. $\mathcal{AMS}$ environments are supported. User-defined macros are available during conversion, due to native $\text{\LaTeX}$ processing.
Theorems:	Support for native $\text{\LaTeX}$ theorems, plus <code>theorem</code> , <code>amsthm</code> , <code>ntheorem</code> .

## lwarp Supported Functions — continued

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

## lwarp Supported Functions — continued

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



Table 2: Additional supported packages

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

lwarp Supported Functions — continued

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

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

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

### 3 Alternatives

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

#### 3.1 Internet class

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

#### 3.2 TeX4ht

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

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

#### 3.3 Translators

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

Prog **Hevea** **H<sup>E</sup>v<sup>E</sup>a**: <http://hevea.inria.fr/> (not on CTAN)

Prog **TtH** **T<sub>T</sub>H**: <http://hutchinson.belmont.ma.us/tth/>

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

Prog **LaTeXML**  **$\text{\LaTeX}$ XML**: <http://dlmf.nist.gov/LaTeXML/>

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

Prog **LaTeX2HTML**  **$\text{\LaTeX}$ 2HTML**: <http://www.latex2html.org/>  
and <http://ctan.org/pkg/latex2html>.

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

Finally, GladTeX may be used to directly insert L<sup>A</sup>T<sub>E</sub>X math into HTML:

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

### 3.4 AsciiDoc

AsciiDoc is one of the most capable markup languages, providing enough features to produce the typical technical-writing document with cross-references, and it writes L<sup>A</sup>T<sub>E</sub>X and HTML.

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

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

The Asciidoctor-LaTeX project is adding additional L<sup>A</sup>T<sub>E</sub>X-related features.

**Asciidoctor-LaTeX:**

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

### 3.5 Pandoc

Prog Pandoc

A markup system which also reads and writes L<sup>A</sup>T<sub>E</sub>X and HTML.

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

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

### 3.6 Word processors

Prog Word It should be noted that the popular word processors have advanced through the years in their abilities to represent math with a L<sup>A</sup>T<sub>E</sub>X-ish input syntax, unicode math fonts, and high-quality output, and also generate HTML with varying success.

Prog LibreOffice

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

### 3.7 Commercial systems

Prog	Adobe	Likewise, several professional systems exist whose abilities have been advancing in the areas of typesetting, cross-referencing, and HTML generation. See Adobe® FrameMaker®, Adobe® InDesign®, and Madcap Flare™.
Prog	FrameMaker	
Prog	InDesign	
	Flare	
Prog	Madcap	3.8 Comparisons

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

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

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

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

## 4 Installation

Table 3 shows the tools which are used for the L<sup>A</sup>T<sub>E</sub>X to HTML conversion. In most cases, these will be available via the standard package-installation tools.

### 4.1 Installing the lwarp package

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

**Pre-installed:** Try entering into a command line:

```
Enter ⇒ kpsewhich lwarp.sty
```

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

**T<sub>E</sub>X Live:** If using a T<sub>E</sub>X Live distribution, try installing via tlmgr:

```
Enter ⇒ tlmgr install lwarp
```

**MiK<sub>T</sub>E<sub>X</sub>:** If using MiK<sub>T</sub>E<sub>X</sub>, try using the package installer to install the package lwarp. Also update the package miktex-misc, which will install the lwarpmk executable.

**Operating-system package:** The operating-system package manager may already have lwarp, perhaps as part of a set of T<sub>E</sub>X-related packages.

**CTAN TDS archive:** lwarp may be downloaded from the Comprehensive T<sub>E</sub>X Archive:

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

**T<sub>E</sub>X Live:**

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

**MiK<sub>T</sub>E<sub>X</sub>:**

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

This should be something like:

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

4. Unpack the archive in the TDS local directory.

Table 3: Required software programs

---

**Provided by your L<sup>A</sup>T<sub>E</sub>X distribution:**

From T<sub>E</sub>XLive: <http://tug.org/texlive/>.

**L<sup>A</sup>T<sub>E</sub>X:** pdf<sub>l</sub>atex, xelatex, or lualatex.

**The lwarp package:** This package.

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

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

**xindy:** The xindy package is used by lwarp to create indexes. On a MiK<sub>T</sub>E<sub>X</sub> 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<sup>A</sup>T<sub>E</sub>X code. On a MiK<sub>T</sub>E<sub>X</sub> system, Perl may need to be installed first.

**pdfcrop:** Used to pull images out of the L<sup>A</sup>T<sub>E</sub>X PDF.

**Poppler PDF utilities:**

**pdftotext:** Used to convert PDF to text.

**pdfseparate:** Used to pull images out of the L<sup>A</sup>T<sub>E</sub>X PDF.

**pdftocairo:** Used to convert images to SVG.

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

From Poppler: [poppler.freedesktop.org](http://poppler.freedesktop.org).

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

Enter ⇒ `brew install poppler`

For Windows, see:

<https://sourceforge.net/projects/poppler-win32/> and:

<http://blog.alivate.com.au/poppler-windows/>

**Perl:**

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

[perl.org](http://perl.org), [strawberryperl.com](http://strawberryperl.com)

**Automatically downloaded from the internet as required:**

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

From: [mathjax.org](http://mathjax.org)

---

5. Renew the cache:

Enter  $\Rightarrow$  `mktextlsr`

— or —

Enter  $\Rightarrow$  `texhash`

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

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

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

Enter  $\Rightarrow$  `pdflatex lwarp.dtx`

(several times)

6. Create the .sty files:

Enter  $\Rightarrow$  `pdflatex lwarp.ins`

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

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

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

`<texlocal>/doc/local/lwarp`

9. Renew the cache:

Enter  $\Rightarrow$  `mktextlsr`

— or —

Enter  $\Rightarrow$  `texhash`

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

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

Just testing!

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

## 4.2 Installing the lwarpmk utility

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

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

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

Otherwise, continue with the following:

3. Locate the file lwarpmk.lua, which should be in the scripts directory of the TDS tree. On a T<sub>E</sub>X Live or MiKTeX system you may use

```
Enter ⇒ kpsewhich lwarpmk.lua
```

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

4. Create lwarpmk:

**Unix:** Create a symbolic link and make it executable:

- (a) Locate the T<sub>E</sub>X Live binaries:

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

This will be something like:

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

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

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

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

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

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



- (c) Make the link executable:

Enter  $\Rightarrow$  `chmod 0755 lwarpmk`

**Windows T<sub>E</sub>X Live:** Create a new `lwarpmk.exe` file:

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

**Windows MiK<sub>T</sub>E<sub>X</sub>:** Create a new `lwarpmk.bat` file:

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

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

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

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

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

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

#### 4.2.1 Using a local copy of `lwarpmk`

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

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

`\usepackage[lwarpmk]{lwarp}`

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

**Unix:**

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

Enter  $\Rightarrow$  `chmod 0755 lwarpmk.lua`

- (b) Compile documents with

Enter  $\Rightarrow$  `./lwarpmk.lua html`

Enter  $\Rightarrow$  `./lwarpmk.lua print`

etc.

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

**Windows:**

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

Enter  $\Rightarrow$  `texlua lwarpmk.lua html`

Enter  $\Rightarrow$  `texlua lwarpmk.lua print`

etc.

Or:

Enter  $\Rightarrow$  `lwarpmk html`

Enter  $\Rightarrow$  `lwarpmk print`

etc.

### 4.3 Installing additional utilities

**To test for the existence of the additional utilities:**

Enter the following in a command line. If each programs' version is displayed, then that utility is already installed. See table 3 on page 38.

Enter  $\Rightarrow$  `luatex -version`

Enter  $\Rightarrow$  `xindy -version`

Enter  $\Rightarrow$  `latexmk -version`

Enter  $\Rightarrow$  `perl -version`

Enter  $\Rightarrow$  `pdftocrop -version`

Enter  $\Rightarrow$  `pdftotext -v`

Enter  $\Rightarrow$  `pdfseparate -version`

Enter  $\Rightarrow$  `pdftocairo -v`

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

The T<sub>E</sub>X utilities xindy, latexmk, and pdfcrop may be provided by your operating system's package manager, and are also provided by the CTAN archive:

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

Prog pdftotext  
 Prog pdfseparate  
 Prog pdftocairo

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

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

**To install the Poppler utilities to a MacOS machine:**

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

Enter ⇒

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

2. Install the Poppler utilities:

Enter ⇒ `brew install poppler`

**To install the Poppler utilities to a Windows machine:**

1. See table 3 on page 38.
2. Download and extract the Poppler utilities pdftotext, pdfseparate, and pdftocairo to a directory, such as Poppler.
3. In the Start window, type "Path" to search for results related to Path. Or, open the control panel and search for "Path".
4. Choose "Edit the system environment variables" in the control panel.
5. Choose the "Environment Variables" button.
6. Choose the "Path" variable, then the "Edit" button.
7. Choose the "New" button to make an additional entry.
8. Enter the bin directory of the Poppler utilities, such as:

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

Be sure to include \bin.

9. Click "Ok" when done.

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

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

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

`C:\Strawberry\perl\bin`

Be sure to include `\bin`.

4. Click "Ok" when done.

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

## 5 Tutorial

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

### 5.1 Starting a new project

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

File `tutorial.tex`

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

#### Copy from the documentation PDF:

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


#### Copy from the `lwarp` documentation directory:

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

Enter  $\Rightarrow$  `texdoc -l lwarp_tutorial.txt`

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

File `lwarp_tutorial.txt`

 Note: `.txt` suffix!

 Bad formatting!

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

3. Compile the project:

Enter  $\Rightarrow$  `pdflatex tutorial.tex`

(several times)

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

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

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

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

Figure 1: tutorial.tex listing

Note: There are two pages!

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

\documentclass{book}

\usepackage{iftex}

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

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

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

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

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

---

```

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

\begin{document}

\maketitle                % Or titlepage/titlingpage environment.

% An article abstract would go here.

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

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

\section{Some math}

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

\printindex

\end{document}

```

Table 4: Files created along with the print version

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

---

When the lwarpmk option is given to the lwarp package:

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

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

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

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



## 5.2 Compiling the print version with `lwarpmk`

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

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

1. Re-compile the print version:

Enter  $\Rightarrow$  `lwarpmk print`

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

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

Enter  $\Rightarrow$  `lwarpmk print`

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

4. Force a recompile to occur.

Enter  $\Rightarrow$  `lwarpmk again`

Enter  $\Rightarrow$  `lwarpmk print`

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

5. Process the index.<sup>4</sup>

Enter  $\Rightarrow$  `lwarpmk printindex`

6. Recompile again to include the index.

Enter  $\Rightarrow$  `lwarpmk print`

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

---

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

<sup>4</sup>A `lwarpmk printglossary` command is also available to process a glossary produced with the `glossaries` package. See section 7.8.5.

### 5.3 Compiling the HTML version with lwarpmk

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

1. Compile the HTML version:

Enter  $\Rightarrow$  `lwarpmk html`

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

2. View the homepage in a web browser.

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

math

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

3. Force a recompile:

Enter  $\Rightarrow$  `lwarpmk again`

Enter  $\Rightarrow$  `lwarpmk html`

Enter  $\Rightarrow$  `lwarpmk print`

4. Process the HTML index and recompile:<sup>7</sup>

Enter  $\Rightarrow$  `lwarpmk htmlindex`

Enter  $\Rightarrow$  `lwarpmk html`

`_Index.html`, is updated for the new  $\text{\LaTeX}$  index.

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

<sup>5</sup>`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`.

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

<sup>7</sup>A `lwarpmk htmlglossary` command is also available to process a glossary produced with the `glossaries` package. See section 7.8.5.

## 5.4 Generating the SVG images

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


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


1. Create the SVG images:


Enter  $\Rightarrow$  `lwarpmk limages`

Enter  $\Rightarrow$  `lwarpmk html`

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

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

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

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

## 5.5 Using MathJax for math

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

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

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

2. Recompile

```
Enter ⇒ lwarpmk html
```

3. Reload the math page.



### MathJax requirements

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

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

## 5.6 Changing the CSS style

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

The styles provided by `lwarp` include:

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

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

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

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

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




**Rename it!**

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

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

## 5.7 Customizing the HTML output

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

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

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

### Options for the `lwarp` package:

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

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

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

Ctr	<code>tocdepth</code>	<p><b>tocdepth:</b> Sectioning depth of the table of contents. See section 13 for a list of L<sup>A</sup>T<sub>E</sub>X stack depths.</p>
-----	-----------------------	---

Ctrl **SideTOCDepth**

[sideTOC](#)

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

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

It is recommended to set:

`SideTOCDepth=FileDepth`

or

`SideTOCDepth=FileDepth+1`

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



Ctrl **FileDepth**

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

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

`\setcounter{FileDepth}{-5}`

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

`\setcounter{FileDepth}{1}`



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

`tocdepth >= FileDepth`

Bool **CombineHigherDepths**

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

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

Inaccessible pages!

Lost in an old page!

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

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

Bool `FileSectionNames`

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

HTML filenames

Example HTML filenames:

**Numbered html nodes:**

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

---

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

---

**Named html sections, no prefix:**

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

---

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

---

**Named html sections, with prefix:**

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

---

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

---

`\abstractname`

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

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

`\CSSFilename`

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




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

The CSS styles of the web pages are set by the `\CSSFilename` command. If `\CSSFilename` is not used, a default plain style is used to mimic printed L<sup>A</sup>T<sub>E</sub>X 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 pagess of the website.

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

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

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

**Placed in the document wherever necessary:**

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

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

Env **warpHTML**

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

**\warpprintonly**

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

**\warpHTMLonly**

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

## 5.8 Using latexmk

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

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

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

2. Recompile the printed version of the document.

```
Enter ⇒ lwarpmk print
```

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

3. Recompile the document.

```
Enter ⇒ lwarpmk print
```

and/or

```
Enter ⇒ lwarpmk html
```

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

## 5.9 Using XeLaTeX or LuaLaTeX

X<sub>Y</sub>LaTeX or LuaLaTeX may be used instead of L<sup>A</sup>T<sub>E</sub>X.

1. Remove the auxiliary files for the project:

```
Enter ⇒  lwarpmk cleanall
```

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

```
Enter ⇒  xelatex tutorial.tex
```

-or-

```
Enter ⇒  lualatex tutorial.tex
```

When the recompile occurs, the configuration files for **lwarpmk** are modified to remember which T<sub>E</sub>X engine was used. X<sub>Y</sub>LaTeX or LuaLaTeX will be used for future runs of **lwarpmk**.

3. To recompile the document:

```
Enter ⇒  lwarpmk print
```

-and-

```
Enter ⇒  lwarpmk html
```

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

## 5.10 Using a glossary

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

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

---

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

---

To process the glossary for the print version:

Enter  $\Rightarrow$  `lwarpmk printglossary`

To process the glossary for the HTML version:

Enter  $\Rightarrow$  `lwarpmk htmlglossary`

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

## 5.11 Cleaning auxiliary files

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

```
Enter ⇒ lwarpmk clean
```

## 5.12 Cleaning auxiliary and output files

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

```
Enter ⇒ lwarpmk cleanall
```

## 5.13 Processing multiple projects in the same directory

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

To create each project:

```
Enter ⇒ pdflatex project_a
```

```
Enter ⇒ pdflatex project_b
```

Each project is given its own configuration file:

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

To compile each project with `lwarkmk`:

```
Enter ⇒ lwarpmk print project_a
```

```
Enter ⇒ lwarpmk html project_b
```

## 5.14 Using the make utility

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

```
lwarpmk pdftohtml [project]
```

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

## 6 Additional details

### 6.1 Font and UTF-8 support

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

vector fonts  
Computer Modern



`\usepackage{lmodern}`

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

`\usepackage{dejavu}`

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



X<sub>Y</sub>LaTeX and LuaLaTeX users must use the `fontspec` package. Do NOT use `fontenc`!

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

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

- For X<sub>Y</sub>LaTeX or LuaLaTeX:

Pkg `fontspec`

- `fontspec` and font choices

ligatures

lwarp sets the following to turn off T<sub>E</sub>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 `pdflatex`:

Pkg `lmodern`

- `lmodern` or other font-related packages

Pkg `fontenc`

- `fontenc`

Pkg `inputenc`

- `inputenc`

Pkg `newunicodechar`

- `newunicodechar`

File `glyphtounicode`

- `\input glyphtounicode.tex`

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

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

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

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

### 6.1.1 Indexes and UTF-8

lwarp uses the xindy program to processes indexes.

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

While using pdflatex, the texindy program is used with the `-C utf8` option, which is newly supported in recent distributions of L<sup>A</sup>T<sub>E</sub>X. This option correctly sorts index entries into headings while using Latin languages, but will not work well with others. XeL<sup>A</sup>T<sub>E</sub>X or LuaL<sup>A</sup>T<sub>E</sub>X are recommended for non-Latin languages.

For an older distribution of L<sup>A</sup>T<sub>E</sub>X, it may be necessary to generate a local version of `lwarpmk.lua` and modify it to remove the `-C utf8` option from the texindy call. See section 10.3.

## 6.2 lwarp package loading and options

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

Pkg	lwarp	Load the lwarp package immediately after the font and UTF-8 setup commands.
Opt	warpprint	Select the warpprint option to generate print output (default), or the warpHTML option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual pdflatex, etc. When lwarp is loaded in print mode, it creates <code>&lt;project&gt;_html.tex</code> , which sets the warpHTML option before calling the user's source code <code>&lt;project&gt;.tex</code> . In this way, <code>&lt;project&gt;.tex</code> can <code>\usepackage{lwarp}</code> without any options to create a printed version, while <code>&lt;project&gt;_html.tex</code> will create an HTML version.
Opt	warpHTML	
Opt	mathsvg	For math display, select mathsvg (default), or mathjax. For more information about the math options, see section 7.4.
Opt	mathjax	

See table 5 for the full list of options.



Table 5: Package options

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

### 6.3 Selecting the operating system

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

### 6.4 Selecting actions for print or HTML output

The following environments and macros are used to select actions which only apply to either traditional L<sup>A</sup>T<sub>E</sub>X print-formatted PDF generation, or to HTML generation.

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

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

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

---

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

---

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

---

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

---

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

---

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

---

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

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

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

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

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

## 6.5 Commands to be placed into the `warpprint` environment

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

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

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

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

## 6.6 Title page



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

---

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

---

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

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

## 6.7 HTML page meta descriptions

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

- limitations** Each page of HTML output should have its own HTML meta description, which usually shows up in web search results, is limited to around 150 characters in length, and should not include the ASCII double quote character (").
- placement** Use `\HTMLDescription` just before `\begin{document}` to set the description of the home page, and also just before each sectioning command such as `\chapter` or `\section` where a new file will be generated, depending on `FileDepth`. For example, if `FileDepth` is 1, use `\HTMLDescription` just before each `\section` command, and that description will be placed inside the HTML page for that `\section`. The same description will be used for all following HTML files as well, until reset by a new `\HTMLDescription`. It is best to use a unique description for each HTML file.
- disabling** To disable the generation of HTML description meta tags, use:  
`\HTMLDescription{}`

## 6.8 HTML page meta author

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

## 6.9 CSS

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

File `project.css`

File `sample_project.css`

---

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

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

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

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

---

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

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

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

## 6.10 Modifying xindy index processing

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

To customize index processing:

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

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

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

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

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

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

## 7 Special cases and limitations

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

Some commonly-used L<sup>A</sup>T<sub>E</sub>X expressions should be modified to allow for a smooth conversion to both HTML and print-formatted outputs:

## 7.1 Formatting

### 7.1.1 Text formatting

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

### 7.1.2 Horizontal space

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

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

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

### 7.1.3 Text alignment

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

### 7.1.4 Superscripts and other non-math uses of math mode

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

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

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

### 7.1.6 Filenames in lists



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

## 7.2 Boxes and minipages

### 7.2.1 Save Boxes

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

### 7.2.2 Minipages

-  **inline** A line of text with an inline minipage or parbox will have the minipage or parbox placed onto its own line, because a paragraph is a block element and cannot be made `inline-block`.
- placement** Minipages and parboxes will be placed side-by-side in HTML unless you place a `\newline` between them.
- side-by-side** Side-by-side minipages may be separated by `\quad`, `\qquad`, `\enskip`, `\hspace`, `\hfill`, or a `\rule`. When inside a `center` environment, the result is similar in print and HTML. Paragraph tags are suppressed between side-by-side minipages and these spacing commands, but not at the start or end of the paragraph.
- in a span** There is limited support for minipages inside an HTML `<span>`. An HTML `<div>` cannot appear inside a `<span>`. While in a `<span>`, 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<sup>A</sup>T<sub>E</sub>X PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.




### 7.2.3 Side-by-side minipages

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

### 7.2.4 Framed minipages and other environments

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

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

- colored boxes and frames: **To create colored frames and boxes:** See section 193 for xcolor's `\colorbox` and `\fcolorbox`, and lwarp'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 `\EndDefiningTabulars` afterwards. Also see the lwarp documentation for the fancybox package.
- To frame equations:** See section 102 for the fancybox package.
- For fancy framed minipages:** See packages `boxedminipage`, `shadow`, `fancybox`, `framed`, `mdframed`.
- Custom environments:** Use a custom environment to create a sidebar, containing a `BlockClass` environment with custom CSS formatting, and `\warpprintonly{\hrule}` command:

---

```

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

```

---

### 7.2.5 fancybox package

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

```

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

```

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

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

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


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

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

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

### 7.2.6 mdframed package

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

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

**font** For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

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

source). Since `lwarp` does not support `\bfseries` and friends, only one font selection may be made at a time.

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

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

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

## 7.3 Cross-references

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

### 7.3.1 Page references

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

### 7.3.2 `cleveref` and `varioref` packages

`cleveref` and `varioref` are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used instead. See section [7](#) to redefine the message which is printed for page number references.

### 7.3.3 Hyperlinks, `hyperref`, and `url`


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

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

### 7.3.4 Footnotes and page notes

lwarp uses native L<sup>A</sup>T<sub>E</sub>X footnote code, although with its own `\box` to avoid the L<sup>A</sup>T<sub>E</sub>X output routine. The usual functions work as-is.

## 7.4 Math


-  **Equation numbering** `ntheorem` has a bug with equation numbering in  $\mathcal{AMS}$  environments when the option `thref` is used. lwarp does not share this bug, so equations with `\split`, etc, are numbered correctly with lwarp's HTML output, but not with the print output. It is recommended to use `cleveref` instead of `ntheorem`'s `thref` option.
- Math rendering** Math may be rendered as SVG graphics or using the MATHJAX JavaScript display engine.
- SVG math option** For SVG math, math is rendered as usual by L<sup>A</sup>T<sub>E</sub>X into the initial PDF file using the current font<sup>8</sup>, 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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X code which generated the math, allowing copy/paste of the L<sup>A</sup>T<sub>E</sub>X math expression into other documents.
- SVG image font size** The size of the math and text used in the SVG image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, for ex:  


$$\renewcommand{\LateximageFontSizeName}{\large}$$
- SVG files** As currently implemented, each instance of math creates a new SVG file. In text with many references to math variables, this can result in a large number of files with duplicate content. In the future, some method of content-based naming and checksumming may be used to remove the need for duplicate files.
- SVG inline** Another approach would be to in-line the SVG files directly into the HTML. This avoids having a large number of files and potentially speeds loading the images, but dis-allows the possibility of sharing one file among many instances without user intervention.
- PNG files** Others have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but SVG files are also larger as well.
- MathML** Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 8 regarding EPUB output with MathJax.

<sup>8</sup>See section 194 regarding fonts and fractions.

MathJax math option	The popular MathJax alternative ( <a href="https://mathjax.org">mathjax.org</a> ) may be used to display math.
<div> <div>Prog</div> <div>MathJax</div> </div>	When MathJax is enabled, math is rendered twice: <ol style="list-style-type: none"> <li>1. As regular <math>\text{\LaTeX}</math> PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of <math>\text{\LaTeX}</math>, and</li> <li>2. As detokenized printed <math>\text{\LaTeX}</math> 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 <math>\text{\LaTeX}</math> values, and the MathJax cross-referencing system is ignored in favor of the <math>\text{\LaTeX}</math> internal system, seamlessly integrating with the rest of the <math>\text{\LaTeX}</math> code.</li> </ol>
MathJax limitations	Limitations when using MathJax include:
<div> <div>Prog</div> <div>MathJax</div> </div>	
chapter numbers	<ul style="list-style-type: none"> <li>• In document classes which have chapters, <code>\tagged</code> equations have the chapter number prepended in HTML output, unlike <math>\text{\LaTeX}</math>. <code>\tag*</code> equations (correctly) do not. This may be improved with future versions of the MathJax support script.</li> </ul> <p><a href="https://groups.google.com/forum/#!topic/mathjax-users/jUteWUcE2bY">https://groups.google.com/forum/#!topic/mathjax-users/jUteWUcE2bY</a></p>
subequations	<ul style="list-style-type: none"> <li>• MathJax itself does not support subequations. This may be improved by parsing the <math>\text{\LaTeX}</math> math expression to manually insert tags, but this has not yet been done.</li> </ul>
footnotes in math	<ul style="list-style-type: none"> <li>• Footnotes inside equations are not yet supported while using MathJax.</li> </ul>
lateximage	<ul style="list-style-type: none"> <li>• Math appearing inside a <code>lateximage</code>, and therefore also inside a <code>Tikz</code> or <code>picture</code> environment, is rendered as SVG math even if MathJax is used in the rest of the document.</li> </ul>
siunitx	<ul style="list-style-type: none"> <li>• Usage of <code>siunitx</code> inside a math equation is supported via a third-party MathJax extension. While inside a math expression, do not use <code>\SI</code> or <code>\si</code> inside <code>\text</code>, where it will be rendered as normal text.</li> </ul> <p><a href="https://github.com/mathjax/MathJax-third-party-extensions/tree/master/siunitx">https://github.com/mathjax/MathJax-third-party-extensions/tree/master/siunitx</a></p>
<div> <div>△</div> <div>siunitx inside an equation</div> </div>	
$\text{\LaTeX}$ macros	<ul style="list-style-type: none"> <li>• MathJax does not automatically support custom <math>\text{\LaTeX}</math> macros, but they may be set up by the user.</li> </ul>
custom MathJax macros	For an example of using custom $\text{\LaTeX}$ macros with MathJax, see page 303.

### 7.4.1 ntheorem package

 **Font control** This conversion is not total. Font control is via CSS, and the custom L<sup>A</sup>T<sub>E</sub>X font settings are ignored.

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

### 7.4.2 siunitx package

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

 **per-mode**

Place the `\SI` expression inside math mode for anything which requires math mode units.

### 7.4.3 units and nicefrac packages

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


### 7.4.4 newtxmath package





Pkg `newtxmath` The proper load order is:

 **loading sequence**

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

## 7.5 Graphics

 **graphics vs. graphicx** If using the older `graphics` syntax, use both optional arguments for `\includegraphics`.

-  **viewports** A single optional parameter is interpreted as the newer **graphicx** syntax. Note that viewports are not supported by **warp**; the entire image will be shown.
-  **\graphicspath** **\graphicspath** only works for a single directory; all graphics must be in this directory.
- units** For **\includegraphics**, avoid **px** and **%** units for width and height, or enclose them inside **warpHTML** environments. For font-proportional image sizes, use **ex** or **em**. For fixed-sized images, use **cm**, **mm**, **in**, **pt**, or **pc**. Using the keys **width=.5\linewidth**, or similar for **\textwidth** or **\textheight** to give fixed-sized images proportional to a 6 by 9 inch text area.
- options** **\includegraphics** accepts **width** and **height**, **origin**, **rotate** and **scale**, plus a new **class** key.
- HTML class** With HTML output, **\includegraphics** accepts an optional **class=xyz** keyval combination, and if this is given then the HTML output will include that class for the image. The class is ignored for print output.
-  **image file types** For **\includegraphics** the user should provide both **.pdf** and **.svg** images, but always refer to **.pdf** images in the document source. All **\includegraphics** references to **.pdf** will automatically be changed to **.svg** for HTML output, and will be left as **.pdf** for print output. Images may also be **.jpg** and **.png**, and will be used as-is for either output.
- \rotatebox** **\rotatebox** accepts the optional **origin** key.
-  **browser support** **\rotatebox**, **\scalebox**, and **\reflectbox** depend on modern browser support. The CSS3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike L<sup>A</sup>T<sub>E</sub>X, so expect some ugly results for scaling and rotating.

### 7.5.1 xcolor package

- \colorboxBlock** and **\fcolorboxBlock** **\colorboxBlock** and **\fcolorboxBlock** are provided for increased HTML compatibility, and they are identical to **\colorbox** and **\fcolorbox** in print mode. In HTML mode they place their contents into a **<div>** instead of a **<span>**. These **<div>**s are set to **display: inline-block** so adjacent **\colorboxBlocks** appear side-by-side in HTML, although text is placed before or after each.
- Print-mode definitions for **\colorboxBlock** and **\fcolorboxBlock** are created by **lwarp**'s core if **xcolor** is loaded.
- background: none** **\fcolorbox** and **\fcolorboxBlock** allow a background color of **none**, in which case only the frame is drawn, which can be useful for HTML.
- color support** Color definitions, models, and mixing are fully supported without any changes




required.

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

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

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

## 7.6 Tabular


 **misplaced alignment  
alignment tab character &**

- When defining environments or macros which include **tabular** plus instances of the **&** character, it may be necessary to make **&** active before the environment or macro is defined, then restore **&** to its default catcode after, using the following commands:

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

**column types**

- Vertical rules are not yet supported.
- \*** in a column specification is not used (so far). Repeat the column type the correct number of times.
- Only one each of **@**, **!**, **>**, and **<** may be used at each column, and they are used in that order.
- `\newcolumnntype` is ignored; unknown column types are set to **l**.
- tabularx** ignores the width, but **X** columns do produce paragraph columns or multicolumns.

 **`\multirow &`  
`\multicolumn`**

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

 **`\multirow`**

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

**`\multirow with rules`**

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

**rule at last row**

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

 **paragraphs**

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

**`\cmidrule width, trim`**

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

- `\cmidrule` and `\multicolumn`
  - `\cmidrule` borders are generated by the individual cells on the following row, and so do not necessarily work correctly when the following row has `\multicolumn` cells below `\cmidrule` borders.
- longtable headings
  - For longtable, place headings and footings which do not apply to HTML inside `\warpprintonly{}`.
- ⚠ `\warpprintonly`
  - For `\toprule` and `\bottomrule`, when combined with a `warpprint` or `warppHTML` environment, if a “misplaced `\noalign`” error occurs, change
 

```
This & That \endhead
```

 to
 

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

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

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

### 7.6.1 longtable package

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

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

 instead of
 

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

 Doing so helps avoid “Misplaced `\noalign`.” when using `\begin{warpprint}`.  
 Keep the `\endfirsthead` row, which is still relevant to HTML output.
- ⚠ `\kill` is ignored, place a `\kill` line inside
 

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

 or place it inside `\warpprintonly`.

### 7.6.2 float, trivfloat, and/or algorithmicx together

- ⚠ package conflicts If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 187.1.

### 7.6.3 caption and subcaption packages


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

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

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


## 7.7 Floats


### 7.7.1 subfig package

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

**horizontal spacing** In the document source, use `\hfill` and `\hspace*` between subfigures to spread them apart horizontally. The use of other forms of whitespace may cause paragraph tags to be generated, resulting in subfigures appearing on the following lines instead of all on a single line.

### 7.7.2 floatrow package

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

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

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

## 7.8 Miscellaneous

### 7.8.1 abstract package

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

### 7.8.2 verse and memoir

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

---

```
\begin{warpHTML}

\let\attrib\attribution


\end{warpHTML}
```

---

<p>Len <code>\leftskip</code></p> <p>Len <code>\leftmargini</code></p> <p>Len <code>\TMLvleftskip</code></p> <p>Len <code>\TMLleftmargini</code></p>	<p>These lengths are used by <code>verse</code> and <code>memoir</code> to control the left margin, and they may already be set by the user for print output. New lengths <code>\HTMLvleftskip</code> and <code>\HTMLleftmargini</code> are provided to control the margins in HTML output. These new lengths may be set by the user before any <code>verse</code> environment, and persist until they are manually changed again. One reason to change <code>\HTMLleftmargini</code> is if there is a wide <code>\flagverse</code> in use, such as the word “Chorus”, in which case the value of <code>\HTMLleftmargini</code> should be set to a wide enough length to contain “Chorus”. The default is wide enough for a stanza number.</p>
--	--

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

### 7.8.3 newclude package

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

---

*Ex:*

---

```
\documentclass{article}
... <font setup>
\usepackage{newclde}
\usepackage[warpHTML]{lwrap}
...
```

---

#### 7.8.4 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, lwrap patches \FBcolonspace and \FBthinspace to create fixed-width HTML spaces around punctutation:

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

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

#### 7.8.5 glossaries package

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

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

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

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

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

## 8 EPUB conversion

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

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

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

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

To help convert lwarp HTML output to EPUB, add

```
\booltrue{FormatEPUB}
```

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

```
Enter ⇒ lwarpmk cleanall
```


then

```
Enter ⇒ lwarpmk html
```

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


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

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

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

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

Check the box Add linked files in breadth first order.

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

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

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

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

Math HTML tags with tag name:

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

`//h:h4` — or — `//h:h3`

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

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

---

#### *MathJax support*

---

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

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

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

---

## 9 Word-processor conversion

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

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

Bool    **FormatWordProcessor**

---

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

---

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

```
\booltrue{FormatWordProcessor}
```

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

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

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

Bool    **HTMLMarkFloats**

---

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

---

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



## 10 Modifying lwarp

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

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

An environment may be converted to a `lateximage` then displayed with an image of the resulting L<sup>A</sup>T<sub>E</sub>X output. See section 68 for an example of the `picture` environment.

To create a custom HTML block or inline CSS class, see section 37.8.

### 10.1 Creating an lwarp version of a package

When creating HTML, `lwarp` redefines the `\usepackage` and `\RequirePackage` macros such that it first looks to see if a `lwarp-⟨packagename⟩.sty` version exists. If so, the `lwarp` version is used instead. This modular system allows users to create their own versions of packages for `lwarp` to use for HTML, simply by creating a new package with a `lwarp-` prefix. If placed in the local directory along with the source code, it will be seen by that project alone. If placed alongside the other `lwarp-` packages where T<sub>E</sub>X can see it, then the user’s new package will be seen by any documents using `lwarp`. (Remember `mktextlsr` or `texhash`.)

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

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

Each `lwarp-` package should first call either `\LWR@ProvidesPackageDrop` or `\LWR@ProvidesPackagePass`. If `Dropped`, the original print-version package is ignored, and only the `lwarp-` version is used. Use this where the original print version is useless for HTML. If `Passed`, the original package is loaded first, with the user-supplied options, then the `lwarp-` version continues loading as well. See section 153 (Ntheorem) for an example of selectively disabling user options for a package. Use this when HTML output only requires some modifications of the

original package. For a case where the original package is usable without changes, there is no need to create a `lwarp-` version.

## 10.2 Testing lwarp

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

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

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

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

## 10.3 Modifying lwarpmk

Prog `lwarpmk`  
File `lwarpmk.lua`

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

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

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

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

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

## 11 Troubleshooting

### 11.1 Using the `lwarp.sty` package

Also see:

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

Section 7: Special cases and limitations

Section 7: Special cases and limitations

#### Text is not converting:

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

#### Undefined HTML settings:

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

#### Misplaced alignment character `&`:

- If using `tabular` inside a `\fbox` or other macro, use `\StartDefiningTabulars` before the definition, and `\EndDefiningTabulars` after. See section 7.6

#### Obscure error messages:

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

#### Missing sections:

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

#### Missing HTML files:

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

#### Missing / incorrect cross-references:

- Use `lwarpmk` again followed by `lwarpmk html` or `lwarpmk print` to compile the document one more time.
- `\nameref` refers to the most recently-used section where the `\label` was defined. If no section has been defined before the `\label`, the link will be empty. Index entries also use `\nameref` and have the same limitation.

- `cleveref` and `varioref` are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used instead. See section 7 to redefine the message which is printed for page number references.

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

Use  $\XeTeX$  or  $\LuaTeX$ .

### Floats out of sequence:

**Mixed “Here” and floating:** Floats [H]ere and regular floats may become out of order. `\clearpage` if necessary.

**Caption setup:** With `\captionsetup` set the positions for the captions above or below to match their use in the source code.

### Print document contains html tags:

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

### HTML document contains a single unformatted print document:

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

### Images are appearing in strange places:

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

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

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

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

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

### Plain-looking document:

⚠ Adding/removing

⚠ HTML instead of images

⚠ Lots of files!

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

#### Broken fragments of HTML:

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

#### Changes do not seem to be taking effect:

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

#### Un-matched conditional compiles:

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

#### 11.1.1 Debug tracing output

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

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

### 11.2 Compiling the `lwarp.dtx` file

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

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

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

**Un-nested environments:**

Be sure to properly nest:

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

## 12 Implementation

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

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

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

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

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

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

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

## 13 Stack depths

Stacks are created to track depth inside the L<sup>A</sup>T<sub>E</sub>X document structure. This depth is translated to HTML headings as shown in table 6. “Depth” here is not depth in

Table 6: Section depths and HTML headings

Section	L <sup>A</sup> T <sub>E</sub> X depth	HTML headings
title of the entire website		<code>&lt;h1&gt;</code>
none	-5	new for this package
book	-2	<b>not yet used</b>
part	-1	<code>&lt;h2&gt;</code>
chapter	0	<code>&lt;h3&gt;</code>
section	1	<code>&lt;h4&gt;</code>
subsection	2	<code>&lt;h5&gt;</code>
subsubsection	3	<code>&lt;h6&gt;</code>
paragraph	4	<code>&lt;span class = "paragraph"&gt;</code>
subparagraph	5	<code>&lt;span class = "subparagraph"&gt;</code>
listitem	7	new for this package, used for list items

the traditional computer-science stack-usage sense, but rather a representation of the nesting depth inside the L<sup>A</sup>T<sub>E</sub>X 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.

## 14 Source Code


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

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

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

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

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

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

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

lwarp source code begins on the following page.



## 15 Detecting the T<sub>E</sub>X Engine — pdf<sub>l</sub>atex, lua<sub>l</sub>-tex, xelatex

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

## 16 Unicode Input Characters

**for HTML & PRINT:** If using pdf<sub>l</sub>atex, convert a minimal set of Unicode characters. Additional characters may be defined by the user, as needed.

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

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

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

In PDF<sub>T</sub>E<sub>X</sub>, preserve upright quotes in verbatim text:

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

## 17 Early package requirements

Pkg	etoolbox	Provides \ifbool and other functions.  22 \RequirePackage{etoolbox}[2011/01/03] 23 % requires v2.6 for \BeforeBeginEnvironment, etc.
Pkg	ifplatform	Provides \ifwindows to try to automatically detect Windows OS.  24 \RequirePackage{ifplatform}% sense op-system platform
Pkg	comment	Provides conditional code blocks.  25 \RequirePackage{comment} 26 \excludecomment{testing}
Pkg	letltxmacro	Used to redefine \textbf and friends.  27 \RequirePackage{letltxmacro}

## 18 Operating-System portability

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

### 18.1 Common portability code

Bool usingOSWindows Set if the OSWindows option is used.

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

### 18.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

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

### 18.3 MS-Windows

For MS-Windows:

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

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

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

```

36 \ifwindows
37 \LWR@setOSWindows
38 \fi

```

## 19 Package options

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

```

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

```

**Bool** **warpingprint**

**Bool** **warpingHTML**

**Bool** **mathjax**

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

```

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

```

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

Only process the contents if producing printed output.

```

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

```

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

Only process the contents if producing HTML output.

```

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

```

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

```

46 \includecomment{warpall}

```

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

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

```

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

```

```

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

```

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

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

```

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

```

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

```

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

```

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

```

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

```

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

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

```

69 \DeclareStringOption[\jobname]{BaseJobname}

```

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

```

70 \DeclareStringOption[english]{IndexLanguage}

```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

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

```

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

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

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

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

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

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

Defaults unless already over-ridden by the user:

```

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

```

## 20 Misplaced packages

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

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

`lwarp-<packagename>.sty`

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

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

```

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

```

```

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

```

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

```

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

```

`\LWR@loadnever`  $\{\langle bad package name \rangle\} \{\langle replacement pkg name \rangle\}$

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

```

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

```

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

The following packages must be loaded after lwarp:

```

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

```



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

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

## 21 Required packages

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

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

**for HTML & PRINT:** 236 `\begin{warpall}`

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

Detects  $\text{\TeX}$  and  $\text{\LuaTeX}$ :

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

248 \end{warpall}
```

**for HTML output:** 249 `\begin{warpHTML}`

```
250 \ifxetexorluatex
251 % ^^A \usepackage[no-math]{fontspec}
```

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

```
252 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
253 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
254 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
255 \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<sup>A</sup>T<sub>E</sub>X dash and quote ligatures, which may fail on older browsers but at least won't corrupt written words.

```

256 \RequirePackage {microtype}
257
258 \microtypesetup{
259     protrusion=false,
260     expansion=false,
261     tracking=false,
262     kerning=false,
263     spacing=false}
264
265 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}

266 \fi

267 \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:**

```

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

```

**for HTML & PRINT:**

```

276 \begin{warpall}

```

**Pkg xparse**

L<sup>A</sup>T<sub>E</sub>X3 command argument parsing

```

277 \RequirePackage{xparse}

278 \end{warpall}

```

```
for HTML output: 279 \begin{warpHTML}

    Pkg  expl3

    LATEX3 programming

    280 \RequirePackage{expl3}

Pkg  gettitlestring

    Used to emulate \nameref.

    281 \RequirePackage{gettitlestring}

Pkg  everyhook

    everyhook is used to patch paragraph handling.

    282 \RequirePackage{everyhook}
    283 \end{warpHTML}

for HTML & PRINT: 284 \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

    285 \RequirePackage{filecontents}
    286
    287 \@ifpackagelater{filecontents}{2011/10/09}%
    288 {}
    289 {
    290 \newwrite\fcwrite
    291 \let\LWR@origfilec@ntents\filec@ntents
    292 \def\filec@ntents{\def\chardef##1\write{\let\reserved@c\fcwrite}\LWR@origfilec@ntents}
    293 }

    294 \end{warpall}

for HTML output: 295 \begin{warpHTML}

Pkg  xifthen

    296 \RequirePackage{xifthen}
```

Pkg **xstring**

297 \RequirePackage{xstring}

Pkg **xstring**

298 \RequirePackage{verbatim}

Pkg **makeidx**

299 \RequirePackage{makeidx}

300 \makeindex

Pkg **calc**

301 \RequirePackage{calc}

Pkg **refcount**

302 \RequirePackage{refcount}

Pkg **newfloat**

303 \RequirePackage{newfloat}

Pkg **caption**

304 \RequirePackage{caption}

305 \end{warpHTML}

**for HTML & PRINT:** 306 \begin{warpall}

Pkg **titling**

Used for \maketitle and the title page. See section 50.

307 \RequirePackage{titling}

Pkg **environ**

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

308 \RequirePackage{environ}

309 \end{warpall}

**for HTML output:** 310 \begin{warpHTML}

Pkg **zref**

Used for cross-references.

```
311 \RequirePackage{zref}
```

Pkg **amsmath**

Equation numbers are placed to the left for HTML.

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

```
312 \PassOptionsToPackage{leqno}{amsmath}
```

```
313 \RequirePackage{amsmath}
```

Pkg **xfrac**

Patched for HTML use. See section 194.

```
314 \RequirePackage{xfrac}
```

Used to convert lengths for image width/height options.

```
315 \RequirePackage{printlen}
```

```
316 \end{warpHTML}
```

## 22 Loading packages

for HTML output: 317 \begin{warpHTML}

Remember the original `\RequirePackage`:

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

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

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

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

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

```

320 \newcommand*\LWR@findword[3][,]{%
321     \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
322 }

```

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

```

323 \newcommand*\LWR@lookforpackagename[1]{%

```

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

```

324 \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`.

```

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

```

See if the package name was found:

```

326 \IfStrEq{\LWR@strresulttwo}{}%
327 {}% no filename
328 {}% yes filename

```

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

```

329 \IfFileExists{lwarp-\LWR@strresulttwo.sty}%
330 {}% lwarp-* file found
331     \StrSubstitute%
332         {\LWR@requirepackagenames}%
333         {\LWR@strresult}%
334         {lwarp-\LWR@strresulttwo}[\LWR@requirepackagenames]%
335 }%
336 {}% no lwarp-* file
337 }% yes filename
338 }

```

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

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

```

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

```

Redirect up to nine names:

```

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

```



```

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

```

\RequirePackage depending on the options and version:

```

350 \IfValueTF{#1}
351 {% options given
352   \IfValueTF{#3}% version given?
353   {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}[#3]}
354   {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}
355 }
356 {% no options given
357   \IfValueTF{#3}% version given?
358   {\LWR@origRequirePackage{\LWR@requirepackagenames}[#3]}
359   {\LWR@origRequirePackage{\LWR@requirepackagenames}}
360 }
361 }
362 \LetLtxMacro{\usepackage}{\RequirePackage}

```

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

Uses the original package, including options.

```

363 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
364 \PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
365 \IfValueTF{#2}
366 {\ProvidesPackage{lwarp-#1}[#2]}
367 {\ProvidesPackage{lwarp-#1}}
368 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
369 \ProcessOptions\relax
370
371 \IfValueTF{#2}
372 {\LWR@origRequirePackage{#1}[#2]}
373 {\LWR@origRequirePackage{#1}}
374 }

```

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

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

```

375 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
376 \PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
377 \IfValueTF{#2}
378 {\ProvidesPackage{lwarp-#1}[#2]}
379 {\ProvidesPackage{lwarp-#1}}
380 \DeclareOption*{}
381 \ProcessOptions\relax
382 }

383 \end{warpHTML}

```

## 23 Copying a file

for HTML output: 384 \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.

```

385 \newcommand*{\LWR@copyfile}[2]{%
386 \newwrite\copyfile % open the file to write to
387 \immediate\openout\copyfile=#2
388 \newread\file % open the file to read from
389 \openin\file=#1
390 \begingroup\endlinechar=-1
391 \makeatletter
392 \loop\unless\ifeof\file
393 \read\file to\fileline % Read one line and store it into \fileline
394 % \fileline\par % print the content into the pdf
395 % print the content:
396 \immediate\write\copyfile{\unexpanded\expandafter{\fileline}}%
397 \repeat
398 \closeout\copyfile
399 \endgroup
400 }

401 \end{warpHTML}

```

## 24 Debugging messages

```

402 \begin{warpall}

```

```

Bool  LWR@tracinglwarp  True if tracing is turned on.

403 \newbool{LWR@tracinglwarp}

\tracinglwarp  Turns on the debug tracing messages.

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

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

405 \newcommand{\LWR@traceinfo}[1]{%
406 \ifbool{LWR@tracinglwarp}%
407 {%
408     \typeout{*** lwarp: #1}%
409     % \PackageInfo{lwarp}{#1 : }%
410 }%
411 {}%
412 }

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

413 \newbool{HTMLDebugComments}
414 \boolfalse{HTMLDebugComments}

If \tracinglwarp, show where preamble hooks occur:

415 \AfterEndPreamble{
416 \LWR@traceinfo{AfterEndPreamble}
417 }
418
419 \AtBeginDocument{
420 \LWR@traceinfo{AtBeginDocument}
421 }

```

## 25 HTML-conversion output modifications

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

```

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

422 \newbool{FormatEPUB}
423 \boolfalse{FormatEPUB}

```

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

```
424 \newbool{FormatWordProcessor}
425 \boolfalse{FormatWordProcessor}
```

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

```
426 \newbool{HTMLMarkFloats}
427 \booltrue{HTMLMarkFloats}
```

```
428 \end{warpall}
```

## 26 Remembering original formatting macros

for HTML output: 429 \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.

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

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

## 27 Textcomp and SIunitx

Mimic the textcomp package if it not loaded, then create HTML versions.

for HTML output: 496 \begin{warpHTML}

```

497 \AtBeginDocument{% in case textcomp was not loaded
498
499 \providecommand*\textdegree{\textdegree}{DEG}
500 \providecommand*\textcelsius{\textcelsius}{DEGC}
501 \providecommand*\textohm{\textohm}{OHM}
502 \providecommand*\textmu{\textmu}{MU}
503
504 \let\LWR@origtextdegree\textdegree
505 \let\LWR@origtextcelsius\textcelsius
506 \let\LWR@origtextohm\textohm
507 \let\LWR@origtextmu\textmu
508
509 \ifPDFTeX
510 \renewcommand*\textdegree{\HTMLentity{deg}}
511 \renewcommand*\textcelsius{\HTMLentity{deg}\,C}
512 \renewcommand*\textohm{\HTMLunicode{2126}}
513 \renewcommand*\textmu{\HTMLunicode{00B5}}
514 \else
515 \renewcommand*\textdegree{\textdegree}{}
516 \renewcommand*\textcelsius{\textcelsius}{\textdegree C}
517 \renewcommand*\textohm{\textohm}{\Omega}
518 \renewcommand*\textmu{\textmu}{\mu}
519 \fi
520
521 \ifpackageloaded{siunitx}
522 {
523 \DeclareSIUnit\bohr{\textit{a}\textsubscript{0}}
524 \DeclareSIUnit\clight{\textit{c}\textsubscript{0}}
525 \DeclareSIUnit\elementarycharge{\textit{e}}
526 \DeclareSIUnit\electronmass{\textit{m}\textsubscript{e}}
527 \DeclareSIUnit\hartree{\textit{E}\textsubscript{h}}
528 \DeclareSIUnit\planckbar{\HTMLunicode{210F}}
529 }% siunitx loaded
530 {}
531
532 }% AtBeginDocument
533
534 % \begin{macrocode}
535 \end{warpHTML}

```

## 28 Configuration Files

```
536 \begin{warpprint}
537 \typeout{lwarp: generating configuration files}
538 \end{warpprint}
```

### 28.1 project\_html.tex

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

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

```
539 \begin{warpprint}
540 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
541 \immediate\openout{LWR@file}=\jobname_html.tex
542 \immediate\write{LWR@file}{%
543 \detokenize{\PassOptionsToPackage}%
544 {warpHTML,BaseJobname=\jobname}{lwarp}%
545 }
546 \immediate\write{LWR@file}{%
547 \detokenize{\input}\string{\jobname.tex}\string }%
548 }
549 \immediate\closeout{LWR@file}
550 \end{warpprint}
```

### 28.2 lwarpmk.conf

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

An example `lwarpmk.conf`:

---

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

---

for PRINT output:

```
551 \begin{warpprint}
552 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
553 \immediate\openout{LWR@file}=lwarpmk.conf
554 \ifbool{usingOSWindows}{
```

```

555 \immediate\write\LWR@file{opsystem = "Windows"}
556 }{
557 \immediate\write\LWR@file{opsystem = "Unix"}
558 }
559 \ifPDFTeX
560 \immediate\write\LWR@file{latexname = "pdflatex"}
561 \fi
562 \ifXeTeX
563 \immediate\write\LWR@file{latexname = "xelatex"}
564 \fi
565 \ifLuaTeX
566 \immediate\write\LWR@file{latexname = "lualatex"}
567 \fi
568 \immediate\write\LWR@file{sourcename = "\jobname"}
569 \immediate\write\LWR@file{%
570 homehtmlfilename = "\HomeHTMLFilename"%
571 }
572 \immediate\write\LWR@file{htmlfilename = "\HTMLFilename"}
573 \immediate\write\LWR@file{latexmk = "\ifbool{LWR@latexmk}{true}{false}}"}
574 \immediate\write\LWR@file{language = "\LWR@IndexLanguage"}
575 \immediate\write\LWR@file{xdyfile = "\LWR@xdyFilename"}
576 \immediate\closeout\LWR@file
577 \end{warpprint}

```

### 28.3 project.lwarpmkconf

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

```

578 \begin{warpprint}
579 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
580 \immediate\openout\LWR@file=\jobname.lwarpmkconf
581 \ifbool{usingOSWindows}{
582 \immediate\write\LWR@file{opsystem = "Windows"}
583 }{
584 \immediate\write\LWR@file{opsystem = "Unix"}
585 }
586 \ifPDFTeX
587 \immediate\write\LWR@file{latexname = "pdflatex"}
588 \fi
589 \ifXeTeX
590 \immediate\write\LWR@file{latexname = "xelatex"}
591 \fi
592 \ifLuaTeX
593 \immediate\write\LWR@file{latexname = "lualatex"}
594 \fi
595 \immediate\write\LWR@file{sourcename = "\jobname"}
596 \immediate\write\LWR@file{%
597 homehtmlfilename = "\HomeHTMLFilename"%

```



```

598 }
599 \immediate\write\LWR@file{htmlfilename = "\HTMLFilename"}
600 \immediate\write\LWR@file{latexmk = "\ifbool{LWR@latexmk}{true}{false}}"}
601 \immediate\write\LWR@file{language = "\LWR@IndexLanguage"}
602 \immediate\write\LWR@file{xdyfile = "\LWR@xdyFilename"}
603 \immediate\closeout\LWR@file
604 \end{warpprint}

```

## 28.4 lwarp.css

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

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

```

605 \begin{warpprint}
606 \begin{filecontents*}{lwarp.css}
607 /*
608  CSS stylesheet for the LaTeX lwarp package
609  Copyright 2016-2017 Brian Dunn -- BD Tech Concepts LLC
610 */
611
612
613 /* a fix for older browsers: */
614 header, section, footer, aside, nav, main,
615     article, figure { display: block; }
616
617
618 A:link {color:#000080 ; text-decoration: none ; }
619 A:visited {color:#800000 ; }
620 A:hover {color:#000080 ; text-decoration: underline ;}
621 A:active {color:#800000 ; }
622
623 a.tocpart {display: inline-block ; margin-left: 0em ;
624     font-weight: bold ;}
625 a.tocchapter {display: inline-block ; margin-left: 0em ;
626     font-weight: bold ;}
627 a.tocsection {display: inline-block ; margin-left: 1em ;
628     text-indent: -.5em ; font-weight: bold ; }
629 a.tocsubsection {display: inline-block ; margin-left: 2em ;
630     text-indent: -.5em ; }
631 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
632     text-indent: -.5em ; }
633 a.tocparagraph {display: inline-block ; margin-left: 4em ;
634     text-indent: -.5em ; }
635 a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
636     text-indent: -.5em ; }

```

```
637 a.tocfigure {margin-left: 0em}
638 a.tocsubfigure {margin-left: 2em}
639 a.tocatable {margin-left: 0em}
640 a.tocsubtable {margin-left: 2em}
641 a.toctheorem {margin-left: 0em}
642 a.toclstlisting {margin-left: 0em}
643
644
645 body {
646     font-family: "DejaVu Serif", "Bitstream Vera Serif",
647         "Lucida Bright", Georgia, serif;
648     background: #FAF7F4 ;
649     color: black ;
650     margin: 0em ;
651     padding: 0em ;
652     font-size: 100% ;
653     line-height: 1.2 ;
654 }
655
656 p {margin: 1.5ex 0em 1.5ex 0em ;}
657
658 /* Holds a section number to add space between it and the name */
659 span.sectionnumber { margin-right: .6em }
660
661 /* Inserted in front of index lines */
662 span.indexitem {margin-left: 0em}
663 span.indexsubitem {margin-left: 2em}
664 span.indexsubsubitem {margin-left: 4em}
665
666 div.hidden, span.hidden { display: none ; }
667
668 kbd {
669     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
670         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
671         "Courier New", monospace;
672     font-size: 100% ;
673 }
674
675 pre { padding: 3pt ; }
676
677 span.strong { font-weight: bold; }
678
679 span.textmd { font-weight: normal; }
680
681 span.textsc { font-variant: small-caps; }
682
683 span.textsl { font-style: oblique; }
684
685 span.textup { font-variant: normal; }
686
```

```
687 span.textrm {
688     font-family: "DejaVu Serif", "Bitstream Vera Serif",
689     "Lucida Bright", Georgia, serif;
690 }
691
692 span.textsf {
693     font-family: "DejaVu Sans", "Bitstream Vera Sans",
694     Geneva, Verdana, sans-serif ;
695 }
696
697 span.attribution {
698     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
699 }
700
701 span.citetitle {
702     margin-left: 1em ; font-size: 80% ; font-style: oblique;
703 }
704
705 span.poemtitle {
706     font-size: 120% ; font-weight: bold;
707 }
708
709 blockquote {
710     margin-left: 0px ;
711     margin-right: 0px ;
712 }
713
714 blockquote p {
715     line-height: 1.5;
716     text-align: left ;
717     font-size: .85em ;
718     margin-left: 3em ;
719     margin-right: 3em ;
720 }
721
722 blockquotation {
723     margin-left: 0px ;
724     margin-right: 0px ;
725 }
726
727 blockquotation p {
728     line-height: 1.5;
729     text-align: left ;
730     font-size: .85em ;
731     margin-left: 3em ;
732     margin-right: 3em ;
733 }
734
735 div.epigraph {
736     line-height: 1.2;
```

```
737     text-align: left ;
738     padding: 3ex 1em 0ex 1em ;
739 /*     margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
740     margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
741 /*     margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
742     font-size: .85em ;
743     max-width: 27em ;
744 }
745
746
747
748 div.epigraphsource{
749     text-align:right ;
750     margin-left:auto ;
751 /*     max-width: 50% ; */
752     border-top: 1px solid #A0A0A0 ;
753     padding-bottom: 3ex ;
754     line-height: 1.2;
755 }
756
757 div.epigraph p { padding: .5ex ; margin: 0ex ;}
758 div.epigraphsource p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
759
760
761 /* lettrine package: */
762 span.lettrine { font-size: 3ex ; float: left ; }
763 span.lettrinetext { font-variant: small-caps ; }
764
765 /* ulem and soul packages: */
766 span.uline {
767     text-decoration: underline ;
768     text-decoration-skip ;
769 }
770
771 span.uuline {
772     text-decoration: underline ;
773     text-decoration-skip ;
774     text-decoration-style: double ;
775 }
776
777 span.uwave {
778     text-decoration: underline ;
779     text-decoration-skip ;
780     text-decoration-style: wavy ;
781 }
782
783 span.sout {
784     text-decoration: line-through ;
785 }
786
```

```
787 span.xout {
788     text-decoration: line-through ;
789 }
790
791 span.dashuline {
792     text-decoration: underline ;
793     text-decoration-skip ;
794     text-decoration-style: dashed ;
795 }
796
797 span.dotuline {
798     text-decoration: underline ;
799     text-decoration-skip ;
800     text-decoration-style: dotted ;
801 }
802
803 span.letterspacing { letter-spacing: .2ex ; }
804
805 span.capsspacing {
806     font-variant: small-caps ;
807     letter-spacing: .1ex ;
808 }
809
810 span.highlight { background: #F8E800 ; }
811
812
813
814
815 html body {
816     margin: 0 ;
817     line-height: 1.2;
818 }
819
820
821 body div {
822     margin: 0ex;
823 }
824
825
826 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
827 {
828     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
829         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
830         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
831         "Times New Roman", serif;
832     font-style: normal ;
833     font-weight: bold ;
834     text-align: left ;
835 }
836
```

```
837 h1 { /* title of the entire website, used on each page */
838     text-align: center ;
839     font-size: 2.5em ;
840     padding: .4ex 0em 0ex 0em ;
841 }
842 h2 { font-size: 2.25em }
843 h3 { font-size: 2em }
844 h4 { font-size: 1.75em }
845 h5 { font-size: 1.5em }
846 h6 { font-size: 1.25em }
847 span.paragraph {font-size: 1em ; font-variant: normal ;
848     margin-right: 1em ; }
849 span.subparagraph {font-size: 1em ; font-variant: normal ;
850     margin-right: 1em ; }
851
852
853
854 /* Title of the file */
855 h1 {
856     margin: 0ex 0em 0ex 0em ;
857     line-height: 1.3;
858     text-align: center ;
859 }
860
861 /* Part */
862 h2 {
863     margin: 1ex 0em 1ex 0em ;
864     line-height: 1.3;
865     text-align: center ;
866 }
867
868 /* Chapter */
869 h3 {
870     margin: 3ex 0em 1ex 0em ;
871     line-height: 1.3;
872 }
873
874 /* Section */
875 h4 {
876     margin: 3ex 0em 1ex 0em ;
877     line-height: 1.3;
878 }
879
880 /* Sub-Section */
881 h5 {
882     margin: 3ex 0em 1ex 0em ;
883     line-height: 1.3;
884 }
885
886 /* Sub-Sub-Section */
```

```
887 h6 {
888     margin: 3ex 0em 1ex 0em ;
889     line-height: 1.3;
890 }
891
892
893 div.titlepage {
894     text-align: center ;
895 }
896
897 .footnotes {
898     font-size: .85em ;
899     margin: 3ex 1em 0ex 1em ;
900     padding-bottom: 1ex ;
901     border-top: 1px solid silver ;
902 }
903
904 .marginpar {
905     max-width:50%;
906     float:right;
907     text-align:left;
908     margin: 1ex 0.5em 1ex 1em ;
909     padding: 1ex 0.5em 1ex 0.5em ;
910     font-size: 85% ;
911     border-top: 1px solid silver ;
912     border-bottom: 1px solid silver ;
913     overflow-x: auto;
914 }
915
916 .marginpar br { margin-bottom: 2ex ; }
917
918 div.marginblock {
919     max-width:50%;
920     float:right;
921     text-align:left;
922     margin: 1ex 0.5em 1ex 1em ;
923     padding: 1ex 0.5em 1ex 0.5em ;
924     overflow-x: auto;
925 }
926
927 div.marginblock div.minipage {
928     display: block ;
929     margin: 0pt auto 0pt auto ;
930 }
931
932 div.marginblock div.minipage p { font-size: 85%}
933
934 div.marginblock br { margin-bottom: 2ex ; }
935
936
```

```
937 section.textbody div.footnotes{
938     margin: 3ex 0em 0ex 0em ;
939     border-bottom: 2px solid silver ;
940 }
941
942 .footnoteheader {
943     border-top: 2px solid silver ;
944     margin-top: 3ex ;
945     padding-top: 1ex ;
946     font-weight: bold ;
947 }
948
949 .mpfootnotes {
950     text-align: left ;
951     font-size: .85em ;
952     margin-left: 1em ;
953     border-top: 1px solid silver ;
954 }
955
956 /* Remove footnote top border in the title page. */
957 div.titlepage div.mpfootnotes {
958     border-top: none ;
959 }
960
961
962
963 ol {
964     margin: 1ex 1em 1ex 0em;
965     line-height: 1.2;
966 }
967
968 ul, body dir, body menu {
969     margin: 1ex 1em 1ex 0em;
970     line-height: 1.2;
971 }
972
973 li { margin: 0ex 0em 1ex 0em; }
974
975 html {
976     margin: 0;
977     padding: 0;
978 }
979
980 .programlisting {
981     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
982         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
983         "Courier New", monospace;
984     margin: 1ex 0ex 1ex 0ex ;
985     padding: .5ex 0pt .5ex 0pt ;
986     overflow-x: auto;
```



```
987 }
988
989 section.textbody>pre.programlisting {
990 border-top: 1px solid silver ;
991 border-bottom: 1px solid silver ;
992 }
993
994
995 .inlineprogramlisting {
996   font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
997               "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
998               "Courier New", monospace;
999   overflow-x: auto;
1000 }
1001
1002 span.listinglabel {
1003   display: inline-block ;
1004   font-size: 70% ;
1005   width: 4em ;
1006   text-align: right ;
1007   margin-right: 2em ;
1008 }
1009
1010 div.abstract {
1011   margin: 2em 5% 2em 5% ;
1012   padding: 1ex 1em 1ex 1em ;
1013   /* font-weight: bold ; */
1014   font-size: 90% ;
1015 }
1016
1017 div.abstract dl {line-height:1.5;}
1018 div.abstract dt {color:#304070;}
1019
1020 div.abstracttitle{
1021   font-family: "URW Classico", Optima, "Linux Biolinum O",
1022               "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1023               "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1024   font-weight:bold;
1025   font-size:1.25em;
1026   text-align: center ;
1027 }
1028
1029 span.abstracrunintitle{
1030   font-family: "URW Classico", Optima, "Linux Biolinum O",
1031               "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1032               "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1033   font-weight:bold;
1034 }
1035
1036
```

```
1037 .verbatim {
1038     overflow-x: auto ;
1039 }
1040
1041 .alltt {
1042     overflow-x: auto ;
1043 }
1044
1045
1046 .bverbatim {
1047     margin: 1ex Opt 1ex Opt ;
1048     padding: .5ex Opt .5ex Opt ;
1049     overflow-x: auto ;
1050 }
1051
1052 .lverbatim {
1053     margin: 1ex Opt 1ex Opt ;
1054     padding: .5ex Opt .5ex Opt ;
1055     overflow-x: auto ;
1056 }
1057
1058 .fancyvrb {
1059     font-size:.85em ;
1060     margin: 3ex Opt 3ex Opt
1061 }
1062
1063 .fancyvrblabel {
1064     font-weight:bold;
1065     text-align: center ;
1066 }
1067
1068
1069 .verse {
1070     font-family: "Linux Libertine Mono O", "Lucida Console",
1071                 "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1072                 "Liberation Mono", "FreeMono", "Andale Mono",
1073                 "Nimbus Mono L", "Courier New", monospace;
1074     margin-left: 1em ;
1075 }
1076
1077
1078 div.singlespace { line-height: 1.2 ; }
1079 div.onehalfspace { line-height: 1.5 ; }
1080 div.doublespace { line-height: 2 ; }
1081
1082
1083
1084
1085
1086 /* Minipage environments, vertically aligned to top, center, bottom: */
```

```
1087 .minipage, .fminipage, .fcolorminipage {
1088     /* display: inline-block ; */
1089     /* Mini pages which follow each other will be tiled. */
1090     margin: .25em .25em .25em .25em;
1091     padding: .25em .25em .25em .25em;
1092     display: inline-flex;
1093     flex-direction: column ;
1094     overflow: auto;
1095 }
1096
1097 /* Paragraphs in the flexbox did not collapse their margins. */
1098 /* Have not yet researched this. */
1099 .minipage p {margin: .75ex 0em .75ex 0em ;}
1100
1101 .fboxBlock .minipage, .colorbox .minipage, .colorboxBlock .minipage,
1102 .fcolorbox .minipage, .fcolorboxBlock .minipage
1103     {border: none ; background: none;}
1104
1105 .fbox, .fboxBlock { border: 1px solid black ; }
1106
1107 .fbox, .fboxBlock, .fcolorbox, .fcolorboxBlock, .colorbox, .colorboxBlock,
1108 .fminipage, .fcolorminipage
1109     {display: inline-block}
1110
1111 .shadowbox, .shabox {
1112     border: 1px solid black;
1113     box-shadow: 3px 3px 3px #808080 ;
1114     border-radius: 0px ;
1115     padding: .4ex .3em .4ex .3em ;
1116     margin: 0pt .3ex 0pt .3ex ;
1117     display: inline-block ;
1118 }
1119
1120 .doublebox {
1121     border: 3px double black;
1122     border-radius: 0px ;
1123     padding: .4ex .3em .4ex .3em ;
1124     margin: 0pt .3ex 0pt .3ex ;
1125     display: inline-block ;
1126 }
1127
1128 .ovalbox, .Ovalbox {
1129     border: 1px solid black;
1130     border-radius: 1ex ;
1131     padding: .4ex .3em .4ex .3em ;
1132     margin: 0pt .3ex 0pt .3ex ;
1133     display: inline-block ;
1134 }
1135
1136 .Ovalbox { border-width: 2px ; }
```

```
1137
1138 .framebox {
1139     border: 1px solid black;
1140     border-radius: 0px ;
1141     padding: .3ex .2em 0ex .2em ;
1142     margin: 0pt .1ex 0pt .1ex ;
1143     display: inline-block ;
1144 }
1145
1146
1147 .mdframed {
1148 /*     padding: 0ex ; */
1149 /*     border: 1px solid black; */
1150 /*     border-radius: 0px ; */
1151     padding: 0ex ;
1152     margin: 3ex 5% 3ex 5% ;
1153 /*     display: inline-block ; */
1154 }
1155
1156 .mdframed p { padding: 0ex .5em 0ex .5em ; }
1157
1158 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
1159
1160 .mdframedtitle {
1161     padding: .5em ;
1162     display: block ;
1163     font-size: 130%
1164 }
1165
1166 .mdframedsubtitle {
1167     padding: 0ex .5em 0ex .5em ;
1168     display: block ;
1169     font-size: 115% ;
1170 }
1171
1172 .mdframedsubsubtitle {
1173     padding: 0ex .5em 0ex .5em ;
1174     display: block ;
1175 }
1176
1177 .mdtheorem {
1178     padding: 0ex .5em 0ex .5em ;
1179     margin: 3ex 5% 3ex 5% ;
1180 /*     display: inline-block ; */
1181 }
1182
1183
1184 /* framed package */
1185 .framed, pre.boxedverbatim, fcolorbox {
1186     margin: 3ex 0em 3ex 0em ;
```

```
1187     border: 1px solid black;
1188     border-radius: 0px ;
1189     padding: .3ex 1em 0ex 1em ;
1190     display: block ;
1191 }
1192
1193 .snugframed {
1194     margin: 3ex 0em 3ex 0em ;
1195     border: 1px solid black;
1196     border-radius: 0px ;
1197     display: block ;
1198 }
1199
1200 .framedleftbar {
1201     margin: 3ex 0em 3ex 0em ;
1202     border-left: 3pt solid black;
1203     border-radius: 0px ;
1204     padding: .3ex .2em .3ex 1em ;
1205     display: block ;
1206 }
1207
1208 .framedtitle {
1209     margin: 0em ;
1210     padding: 0em ;
1211     font-size: 130%
1212 }
1213
1214 .framedtitle p { padding: .3em }
1215
1216
1217
1218 dl {
1219     margin: 1ex 2em 1ex 0em;
1220     line-height: 1.3;
1221 }
1222
1223 dl dt {
1224     margin-top: 1ex;
1225     margin-left: 1em ;
1226     font-weight: bold;
1227 }
1228
1229 dl dd p { margin-top: 0em; }
1230
1231
1232 nav.toc, nav.lof, nav.lot, nav.lol, nav.lothm {
1233     font-family: "URW Classico", Optima, "Linux Biolinum O",
1234     "DejaVu Sans", "Bitstream Vera Sans",
1235     Geneva, Verdana, sans-serif ;
1236     margin-bottom: 4ex ;
```

```
1237 }
1238
1239 nav.toc p, nav.lof p, nav.lot p, nav.lol p, nav.lothm p {
1240     line-height: 1.2 ;
1241     margin-top:.5ex ;
1242     margin-bottom:.5ex;
1243     font-size: .9em ;
1244 }
1245
1246
1247
1248 img, img.hyperimage, img.borderimage {
1249     max-width: 600px;
1250     border: 1px solid silver;
1251     box-shadow: 3px 3px 3px #808080 ;
1252     padding: .5% ;
1253     margin: .5% ;
1254     background: none ;
1255 }
1256
1257 img.inlineimage{
1258     padding: 0px ;
1259     box-shadow: none ;
1260     border: none ;
1261     background: none ;
1262     margin: 0px ;
1263     display: inline-block ;
1264     border-radius: 0px ;
1265 }
1266
1267 img.logoimage{
1268     max-width: 300px ;
1269     box-shadow: 3px 3px 3px #808080 ;
1270     border: 1px solid black ;
1271     background:none ;
1272     padding:0 ;
1273     margin:.5ex ;
1274     border-radius: 10px ;
1275 }
1276
1277
1278 .section {
1279 /*
1280     To have each section float relative to each other:
1281 */
1282 /*
1283     display: block ;
1284     float: left ;
1285     position: relative ;
1286     background: white ;
```

```
1287     border: 1px solid silver ;
1288     padding: .5em ;
1289 */
1290     margin: 0ex .5em 0ex .5em ;
1291     padding: 0 ;
1292 }
1293
1294
1295 figure {
1296     margin: 3ex auto 3ex auto ;
1297     padding: 1ex 1em 1ex 1em ;
1298     overflow-x: auto ;
1299 }
1300
1301
1302 /* To automatically center images in figures: */
1303 /*
1304 figure img.inlineimage {
1305     margin: 0ex auto 0ex auto ;
1306     display: block ;
1307 }
1308 */
1309
1310 /* To automatically center minipages in figures: */
1311 /*
1312 figure div.minipage, figure div.minipage div.minipage {
1313     margin: 1ex auto 1ex auto ;
1314     display: block ;
1315 }
1316 */
1317
1318 figure div.minipage p { font-size: 85% ; }
1319
1320 figure.subfigure, figure.subtable {
1321     display: inline-block ; margin: 3ex 1em 3ex 1em ;
1322 }
1323
1324 figcaption .minipage { margin:0 ; padding: 0 }
1325
1326 div.floatrow { text-align: center; }
1327
1328 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
1329
1330 div.floatfoot { font-size: .85em ;
1331     border-top: 1px solid silver ; line-height: 1.2 ; }
1332
1333 figcaption , .lstlistingtitle {
1334     font-size: .85em ;
1335     text-align: center ;
1336     font-weight: bold ;
```

```

1337     margin-top: 1ex ;
1338     margin-bottom: 1ex ;
1339 }
1340
1341 figure.subfigure figcaption, figure.subtable figcaption {
1342     border-bottom: none ; background: none ;
1343 }
1344
1345 div.nonfloatcaption {
1346     margin: 1ex auto 1ex auto ;
1347     font-size: .85em ;
1348     text-align: center ;
1349     font-weight: bold ;
1350 }
1351
1352 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
1353 figure div.floatrow div.minipage figcaption {
1354     border: none ;
1355     background: none ;
1356 }
1357
1358
1359 table {
1360     margin: 1ex auto 1ex auto ;
1361     border-collapse: collapse ;
1362     border-spacing: 0px ;
1363     line-height: 1.3 ;
1364 }
1365
1366 tr.hline {border-top: 1px solid silver ; margin-top: 0ex ;
1367     margin-bottom: 0ex ; } /* for \hline */
1368
1369 tr.tbrule {border-top: 1px solid black ; margin-top: 0ex ;
1370     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
1371
1372 td {padding: 1ex .5em 1ex .5em ;}
1373
1374 table td.tdl { text-align: left ; vertical-align: middle ; }
1375 table td.tdc { text-align: center ; vertical-align: middle ; }
1376 table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
1377 table td.tdbang { text-align: center ; vertical-align: middle ; }
1378 table td.tdr { text-align: right ; vertical-align: middle ; }
1379 table td.tdp { text-align: left ; vertical-align: bottom ; }
1380 table td.tdm { text-align: left ; vertical-align: middle ; }
1381 table td.tdb { text-align: left ; vertical-align: top ; }
1382 table td.tdP { text-align: center ; vertical-align: bottom ; }
1383 table td.tdM { text-align: center ; vertical-align: middle ; }
1384 table td.tdB { text-align: center ; vertical-align: top ; }
1385 table td.tdlrule { text-align: left ; border-top: 1px solid silver ;
1386     vertical-align: middle ; } /* for cmdrule */

```



```

1387 table td.tdcrule { text-align: center ; border-top: 1px solid silver ;
1388     vertical-align: middle ; }
1389 table td.tdatrule { text-align: center ; border-top: 1px solid silver ;
1390     vertical-align: middle ; padding: 0px ; margin: 0px ; }
1391 table td.tdbangrule { text-align: center ; border-top: 1px solid silver ;
1392     vertical-align: middle ; } /* for cmidrule */
1393 table td.tdrrule { text-align: right ; border-top: 1px solid silver ;
1394     vertical-align: middle ; }
1395 table td.tdprule { text-align: left ; border-top: 1px solid silver ;
1396     vertical-align: bottom ; }
1397 table td.tdmrule { text-align: left ; border-top: 1px solid silver ;
1398     vertical-align: middle ; }
1399 table td.tdbrule { text-align: left ; border-top: 1px solid silver ;
1400     vertical-align: top ; }
1401 table td.tdPrule { text-align: center ; border-top: 1px solid silver ;
1402     vertical-align: bottom ; }
1403 table td.tdMrule { text-align: center ; border-top: 1px solid silver ;
1404     vertical-align: middle ; }
1405 table td.tdBrule { text-align: center ; border-top: 1px solid silver ;
1406     vertical-align: top ; }
1407
1408 /* Margins of paragraphs inside table cells: */
1409 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
1410     padding-bottom: 1ex ; margin: 0ex ; }
1411 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
1412     padding-bottom: 1ex ; margin: 0ex ; }
1413 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
1414     padding-bottom: 1ex ; margin: 0ex ; }
1415
1416 td.tdp , td.tdprule , td.tdP , td.tdPrule
1417     { padding: 0ex .5em 0ex .5em ; }
1418 td.tdm , td.tdmrule , td.tdM , td.tdMrule
1419     { padding: 0ex .5em 0ex .5em ; }
1420 td.tdb , td.tdbrule , td.tdB , td.tdBrule
1421     { padding: 0ex .5em 0ex .5em ; }
1422
1423
1424 /* table notes: */
1425 .tnotes {
1426     margin: 0ex 5% 1ex 5% ;
1427     padding: 0.5ex 1em 0.5ex 1em;
1428     font-size:.85em;
1429     text-align: left ;
1430 }
1431
1432 .tnotes dl dt p {margin-bottom:0px;}
1433
1434 .tnoteitemheader {margin-right: 1em;}
1435
1436

```

```
1437
1438 /* center, flushleft, flushright environments */
1439 div.center{text-align:center;}
1440 div.center table {margin-left:auto;margin-right:auto;}
1441 div.flushleft{text-align:left;}
1442 div.flushleft table {margin-left:0em ; margin-right:auto;}
1443 div.flushright{text-align:right;}
1444 div.flushright table {margin-left:auto ; margin-right: 0em ;}
1445
1446
1447 /* Fancybox */
1448 div.Btrivlist table tr td { padding: .2ex 0em ; }
1449
1450
1451 /* program listing callouts: */
1452 span.callout {
1453     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1454     Geneva, Verdana, sans-serif ;
1455     border-radius: .5em;
1456     background-color:black;
1457     color:white;
1458     padding:0px .25em 0px .25em;
1459     margin: 0 ;
1460     font-weight: bold;
1461     font-size:.72em ;
1462 }
1463
1464 div.programlisting pre.verbatim span.callout{
1465     font-size: .85em ;
1466 }
1467
1468
1469
1470
1471
1472 div.published
1473 {
1474     text-align: center ;
1475     font-variant: normal ;
1476     font-style: italic ;
1477     font-size: 1em ;
1478     margin: 3ex 0em 3ex 0em ;
1479 }
1480
1481 div.subtitle
1482 {
1483     text-align: center ;
1484     font-variant: normal ;
1485     font-style: italic ;
1486     font-size: 1.25em ;
```

```
1487     margin: 3ex 0em 3ex 0em ;
1488 }
1489
1490 div.subtitle p { margin: 1ex ; }
1491
1492 div.author
1493 {
1494     font-variant: normal ;
1495     font-style: normal ;
1496     font-size: 1em ;
1497     margin: 3ex 0em 3ex 0em ;
1498 }
1499
1500 div.author table {
1501     margin: 3ex auto 0ex auto ;
1502     background: none ;
1503 }
1504
1505 div.author table tbody tr td { padding: .25ex ; }
1506
1507 span.affiliation {font-size: .85em ; font-variant: small-caps; }
1508
1509 div.titledate {
1510     text-align: center ;
1511     font-size: .85em ;
1512     font-style: italic;
1513     margin: 6ex 0em 6ex 0em ;
1514 }
1515
1516
1517 nav.topnavigation{
1518     text-align: left ;
1519     padding: 0.5ex 1em 0.5ex 1em ;
1520 /*     margin: 2ex 0em 3ex 0em ; */
1521     margin: 0 ;
1522     border-bottom: 1px solid silver ;
1523     border-top: 1px solid silver ;
1524     clear:right ;
1525 }
1526
1527 nav.botnavigation{
1528     text-align: left ;
1529     padding: 0.5ex 1em 0.5ex 1em ;
1530 /*     margin: 3ex 0em 2ex 0em ; */
1531     margin: 0 ;
1532     border-top: 1px solid silver ;
1533     border-bottom: 1px solid silver ;
1534     clear:right ;
1535 }
1536
```

```
1537
1538 header{
1539     line-height: 1.2 ;
1540     font-size: 1em ;
1541     /*     border-bottom: 2px solid silver ; */
1542     margin: 0px ;
1543     padding: 0ex 1em 0ex 1em ;
1544     text-align:center ;
1545 }
1546
1547 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
1548
1549
1550 footer{
1551     font-size: .85em ;
1552     line-height: 1.2 ;
1553     margin-top: 1ex ;
1554     border-top: 2px solid silver ;
1555     padding: 2ex 1em 2ex 1em ;
1556     clear:right ;
1557     text-align:left ;
1558 }
1559
1560
1561 a.linkhome { font-weight:bold ; font-size: 1em ;}
1562
1563
1564 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
1565
1566 img.lateximage{
1567     padding: 0px 0px 0px 0px ;
1568     box-shadow: none ;
1569     border: none ;
1570     background: none ;
1571     margin: 0px 0px -.15ex 0px ;
1572     /* pdfcrop leaves a slight margin, adjust to baseline */
1573     max-width: 100% ;
1574     border-radius: 0ex ;
1575     border: none ;
1576 }
1577
1578
1579
1580 nav.sidetoc {
1581     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1582         "Lucida Bright", Georgia, serif;
1583     float:right ;
1584     width: 20%;
1585     border-left: 1px solid silver;
1586     border-top: 1px solid silver;
```

```
1587     border-bottom: 1px solid silver;
1588 /*     border-top: 2px solid #808080 ; */
1589     background: #FAF7F4 ;
1590     padding: 2ex 0em 2ex 1em ;
1591     margin: 0ex 0em 2ex 1em ;
1592     font-size:.9em ;
1593     border-radius: 20px 0px 0px 20px ;
1594 }
1595
1596 div.sidetoccontents {
1597 /*     border-top: 1px solid silver ; */
1598     overflow-y: auto ;
1599     width: 100% ;
1600     text-align: left ;
1601 }
1602
1603 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
1604     text-indent: 0 ; }
1605 nav.sidetoc p a {color:black ; font-size: .7em ;}
1606 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
1607     border-bottom: 1px solid silver ; }
1608 nav.sidetoc a:hover {text-decoration: underline ; }
1609
1610
1611
1612 section.textbody { margin: 0ex 1em 0ex 1em ;}
1613
1614
1615 div.multicolsheading { -webkit-column-span: all;
1616     -moz-column-span: all; column-span: all; }
1617 div.multicols { -webkit-columns: 3 380px ;
1618     -moz-columns: 3 380px ; columns: 3 380px ; }
1619 div.multicols p {margin-top: 0ex}
1620
1621
1622
1623 /* Used to support algorithmicx: */
1624 span.floatright { float: right ; }
1625
1626
1627
1628
1629 /* Native LaTeX theorems: */
1630
1631 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
1632 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
1633
1634
1635 /* theorem, amsthm, and ntheorem packages */
1636
```

```
1637 span.theoremheader,
1638 span.theoremheaderplain,
1639 span.theoremheaderdefinition,
1640 span.theoremheaderbreak,
1641 span.theoremheadermarginbreak,
1642 span.theoremheaderchangebreak,
1643 span.theoremheaderchange,
1644 span.theoremheadermargin
1645 {
1646     font-style:normal ; font-weight: bold ; margin-right: 1em ;
1647 }
1648
1649 span.amsthmnameplain,
1650 span.amsthmnamedefinition,
1651 span.amsthmnumberplain,
1652 span.amsthmnumberdefinition
1653 {
1654     font-style:normal ; font-weight: bold ;
1655 }
1656
1657
1658 span.amsthmnameremark,
1659 span.amsthmnumberremark
1660 {font-style:italic ; font-weight: normal ; }
1661
1662
1663 span.amsthmnoteplain,
1664 span.amsthmnotedefinition
1665 {font-style:normal ;}
1666
1667
1668 span.theoremheaderremark,
1669 span.theoremheaderproof,
1670 span.amsthmproofname
1671 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
1672
1673 span.theoremheadersc
1674 {
1675     font-style:normal ;
1676     font-variant: small-caps ;
1677     font-weight: normal ;
1678     margin-right: 1em ;
1679 }
1680
1681 .theoremendmark {float:right}
1682
1683 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
1684 div.theorembodybreak, div.theorembodynonumberbreak,
1685 div.theorembodymarginbreak,
1686 div.theorembodychangebreak,
```

```
1687 div.theorembodychange,
1688 div.theorembodymargin
1689 {
1690     font-style:italic;
1691     margin-top: 3ex ; margin-bottom: 3ex ;
1692 }
1693
1694 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
1695 div.theorembodyplainupright, nonumberplainuprightsc,
1696 div.amsthmbodydefinition, div.amsthmbodyremark,
1697 div.amsthmproof
1698 {
1699     font-style: normal ;
1700     margin-top: 3ex ; margin-bottom: 3ex ;
1701 }
1702
1703 span.amsthmnoteremark {}
1704
1705
1706
1707 /*
1708 For CSS LaTeX and related logos:
1709 Based on:
1710 http://edward.oconnor.cx/2007/08/tex-poshlet
1711 http://nitens.org/taraborelli/texlogo
1712 */
1713
1714 .latexlogofont {
1715     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1716         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1717     font-variant: normal ;
1718 }
1719
1720 .latexlogo {
1721     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1722         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1723     letter-spacing: .03em ;
1724     font-size: 1.1em;
1725 }
1726
1727 .latexlogo sup {
1728     text-transform: uppercase;
1729     letter-spacing: .03em ;
1730     font-size: 0.85em;
1731     vertical-align: 0.15em;
1732     margin-left: -0.36em;
1733     margin-right: -0.15em;
1734 }
1735
1736 .latexlogo sub {
```

```
1737 text-transform: uppercase;
1738 vertical-align: -0.5ex;
1739 margin-left: -0.1667em;
1740 margin-right: -0.125em;
1741 font-size: 1em;
1742 }
1743
1744 .xetexlogo {
1745     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1746         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1747     letter-spacing: .03em ;
1748     font-size: 1.1em;
1749 }
1750
1751 /* A smaller gap between Xe and Tex v.s. LaTeX: */
1752 .xetexlogo sub {
1753     text-transform: uppercase;
1754     vertical-align: -0.5ex;
1755     margin-left: -0.0667em;
1756     margin-right: -0.2em;
1757     font-size: 1em;
1758     letter-spacing: .03em ;
1759 }
1760
1761 /* A large gap between Xe and LaTeX v.s. TeX: */
1762 .xelatexlogo sub {
1763     text-transform: uppercase;
1764     vertical-align: -0.5ex;
1765     margin-left: -0.0667em;
1766     margin-right: -.05em;
1767     font-size: 1em;
1768     letter-spacing: .03em ;
1769 }
1770
1771 .amslogo {
1772     font-family: "TeXGyreChorus", "URW Chancery L",
1773         "Apple Chancery", "ITC Zapf Chancery", "Monotype Corsiva",
1774         "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
1775         "Hoefler Text", Times, "Times New Roman", serif;
1776     font-style: italic;
1777 }
1778
1779 .lyxlogo {
1780     font-family: "URW Classico", Optima, "Linux Biolinum O",
1781         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
1782         Verdana, sans-serif ;
1783 }
1784
1785
1786
```



```
1787
1788 /* Only display top and bottom navigation if a small screen: */
1789 /* Hide the sidetoc if a small screen: */
1790 nav.topnavigation { display:none; }
1791 nav.botnavigation { display:none; }
1792
1793 @media screen and (max-width: 45em) {
1794 /*     nav.sidetoc {display:none;} */
1795     nav.sidetoc {
1796         float: none ;
1797         width: 100% ;
1798         margin: 5ex 0px 5ex 0px ;
1799         padding: 0 ;
1800         border-radius: 0 ;
1801         border-bottom: 1px solid black ;
1802         border-top: 1px solid black ;
1803         box-shadow: none ;
1804     }
1805 /*     nav.topnavigation { display:block } */
1806     nav.botnavigation { display:block }
1807     .marginpar {
1808         max-width: 100%;
1809         float: none;
1810         display:block ;
1811         margin: 1ex 1em 1ex 1em ;
1812     }
1813 }
1814
1815 @media print {
1816     body {
1817         font-family: "Linux Libertine O",
1818             "DejaVu Serif", "Bitstream Vera Serif",
1819             "Liberation Serif", "Nimbus Roman No 9 L",
1820             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1821     }
1822     nav.sidetoc { display:none; }
1823     nav.topnavigation { display: none; }
1824     nav.botnavigation { display: none; }
1825 }
1826
1827 @media handheld {
1828     nav.sidetoc { display:none; }
1829     nav.topnavigation { display:block }
1830     nav.botnavigation { display:block }
1831 }
1832
1833 @media projection {
1834     nav.sidetoc { display:none; }
1835     nav.topnavigation { display:block }
1836     nav.botnavigation { display:block }
```

```

1837 }
1838 \end{filecontents*}
1839 % \end{Verbatim}% for syntax highlighting
1840 \end{warpprint}

```

## 28.5 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.

```

1841 \begin{warpprint}
1842 \begin{filecontents*}{lwarp_sagebrush.css}
1843 @import url("lwarp.css") ;
1844
1845
1846 A:link {color:#105030 ; text-decoration: none ; }
1847 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
1848 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
1849 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
1850
1851
1852
1853 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1854 {
1855     font-family: "URW Classico", Optima, "Linux Biolinum O",
1856               "Linux Libertine O", "Liberation Serif",
1857               "Nimbus Roman No 9 L", "FreeSerif",
1858               "Hoefler Text", Times, "Times New Roman", serif;
1859     font-variant: small-caps ;
1860     font-weight: normal ;
1861     color: #304070 ;
1862     text-shadow: 2px 2px 3px #808080;
1863 }
1864
1865 h1 { /* title of the entire website, used on each page */
1866     font-variant: small-caps ;
1867     color: #304070 ;
1868     text-shadow: 2px 2px 3px #808080;
1869     background-color: #F7F7F0 ;
1870     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
1871 }
1872
1873 h1 {
1874     border-bottom: 1px solid #304070;
1875     border-top: 2px solid #304070;

```

```
1876 }
1877
1878 h2 {
1879   border-bottom: 1px solid #304070;
1880   border-top: 2px solid #304070;
1881   background-color: #F7F7F0 ;
1882   background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
1883 }
1884
1885
1886
1887 div.abstract {
1888   background: #f5f5eb ;
1889   background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1890
1891   border: 1px solid silver;
1892   border-radius: 1em ;
1893 }
1894
1895 div.abstract dl {line-height:1.5;}
1896 div.abstract dt {color:#304070;}
1897
1898 div.abstracttitle{
1899   font-family: "URW Classico", Optima, "Linux Biolinum O",
1900     "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1901     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1902   font-weight:bold;
1903   font-variant: small-caps ;
1904   font-size:1.5em;
1905   border-bottom: 1px solid silver ;
1906   color: #304070 ;
1907   text-align: center ;
1908   text-shadow: 1px 1px 2px #808080;
1909 }
1910
1911 span.abstracrunintitle{
1912   font-family: "URW Classico", Optima, "Linux Biolinum O",
1913     "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1914     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1915   font-weight:bold;
1916 }
1917
1918
1919 div.epigraph {
1920   background: #f5f5eb ;
1921   background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1922
1923   border: 1px solid silver ;
1924   border-radius: 1ex ;
1925   box-shadow: 3px 3px 3px #808080 ;
```

```
1926 }
1927
1928
1929 .example {
1930     background-color: #f5f5eb ;
1931     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1932
1933 }
1934
1935 div.exampletitle{
1936     font-family: "URW Classico", Optima, "Linux Biolinum O",
1937         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1938         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1939     font-weight:bold;
1940     font-variant: small-caps ;
1941     border-bottom: 1px solid silver ;
1942     color: #304070 ;
1943     text-align: center ;
1944     text-shadow: 1px 1px 2px #808080;
1945 }
1946
1947
1948 .sidebar {
1949     background-color: #f5f5eb ;
1950     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1951
1952 }
1953
1954 div.sidebartitle{
1955     font-family: "URW Classico", Optima, "Linux Biolinum O",
1956         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1957         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1958     font-weight:bold;
1959     font-variant: small-caps ;
1960     border-bottom: 1px solid silver ;
1961     color: #304070 ;
1962     text-align: center ;
1963     text-shadow: 1px 1px 2px #808080;
1964 }
1965
1966
1967 .fancyvrblabel {
1968     font-family: "URW Classico", Optima, "Linux Biolinum O",
1969         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1970         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1971     font-weight:bold;
1972     font-variant: small-caps ;
1973     font-size: 1.5em ;
1974     color: #304070 ;
1975     text-align: center ;
```

```
1976     text-shadow: 1px 1px 2px #808080;
1977 }
1978
1979
1980
1981 .minipage {
1982     background-color: #eeeeee7 ;
1983     border: 1px solid silver ;
1984     border-radius: 1ex ;
1985 }
1986
1987 .framed .minipage , .framedleftbar .minipage {
1988     border: none ;
1989     background: none ;
1990     padding: 0ex ;
1991     margin: 0ex ;
1992 }
1993
1994 figure.figure .minipage, figcaption .minipage { border: none; }
1995
1996 div.marginblock div.minipage { border: none; }
1997
1998 figure , div.marginblock {
1999     background-color: #eeeeee7 ;
2000     border: 1px solid silver ;
2001     border-radius: 1ex ;
2002     box-shadow: 3px 3px 3px #808080 ;
2003 }
2004
2005 figure figure {
2006     border: 1px solid silver ;
2007     margin: 0em ;
2008     box-shadow: none ;
2009 }
2010
2011 /*
2012 figcaption {
2013     border-top: 1px solid silver ;
2014     border-bottom: 1px solid silver ;
2015     background-color: #e8e8e8 ;
2016 }
2017 */
2018
2019
2020 div.table {
2021     box-shadow: 3px 3px 3px #808080 ;
2022 }
2023
2024 /*
2025 .tnotes {
```

```
2026     background: #e8e8e8;
2027     border: 1px solid silver;
2028 }
2029 */
2030
2031
2032 nav.topnavigation{
2033     background-color: #b0b8b0 ;
2034     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
2035 }
2036
2037 nav.botnavigation{
2038     background-color: #b0b8b0 ;
2039     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
2040 }
2041
2042
2043
2044 header{
2045     background-color: #F7F7F0 ;
2046     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
2047 }
2048
2049 footer{
2050     background-color: #F7F7F0 ;
2051     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
2052 }
2053
2054
2055
2056 nav.sidetoc {
2057     background-color: #F7F7F0 ;
2058     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
2059     box-shadow: 3px 3px 3px #808080 ;
2060     border-radius: 0px 0px 0px 20px ;
2061 }
2062
2063 div.sidetoc title {color: #304070 ; }
2064
2065 nav.sidetoc a: hover {
2066     color: #006000 ;
2067     text-decoration: none ;
2068     text-shadow: 0px 0px 2px #a0a0a0;
2069 }
2070
2071
2072 @media screen and (max-width: 45em) {
2073     nav.sidetoc { border-radius: 0 ; }
2074 }
2075
```

```

2076
2077 \end{filecontents*}
2078 % \end{Verbatim}% for syntax highlighting
2079 \end{warpprint}

```

## 28.6 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.

```

2080 \begin{warpprint}
2081 \begin{filecontents*}{lwarp_formal.css}
2082 @import url("lwarp.css") ;
2083
2084
2085
2086 A:link {color:#802020 ; text-decoration:none; }
2087 A:visited {color:#802020 ; text-shadow:none ;}
2088 A:hover {color:#400000 ; text-shadow:none ;}
2089 A:active {color:#C00000 ; text-shadow:none ;}
2090
2091
2092 body {
2093     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2094                 "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2095                 "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2096                 "Times New Roman", serif;
2097     background: #fffcf5;
2098 }
2099
2100 span.textrm {
2101     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2102                 "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2103                 "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2104                 "Times New Roman", serif;
2105 }
2106
2107 span.textsf {
2108     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2109                 Geneva, Verdana, sans-serif ;
2110 }
2111
2112
2113
2114 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph

```

```
2115 {
2116     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2117         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2118         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2119         "Times New Roman", serif;
2120     color: #800000 ;
2121     text-shadow: none ;
2122 }
2123
2124 h1, h2 {
2125     background-color: #fffcf5 ;
2126     background-image: none ;
2127     border-bottom: 1px solid #808080;
2128     border-top: 2px solid #808080;
2129 }
2130
2131 div.abstracttitle {
2132     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2133         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2134         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2135         "Times New Roman", serif;
2136     color: black ;
2137     text-shadow: none ;
2138 }
2139
2140 span.abstracrunintitle {
2141     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2142         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2143         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2144         "Times New Roman", serif;
2145     color: black ;
2146     text-shadow: none ;
2147 }
2148
2149 div.abstract { font-size: 100% }
2150
2151 .sidebar {
2152     background: #fffcf5;
2153     background-image: none ;
2154     margin: 2em 5% 2em 5%;
2155     padding: 0.5em 1em;
2156     border: none ;
2157     border-top : 1px solid silver;
2158     border-bottom : 1px solid silver;
2159     font-size: 90% ;
2160 }
2161
2162 div.sidebartitle{
2163     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2164         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
```



```
2165         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2166         "Times New Roman", serif;
2167     color: #800000 ;
2168     text-shadow: none ;
2169     border: none ;
2170 }
2171
2172 .example {
2173     background: #fffcf5;
2174     background-image: none ;
2175     margin: 2em 5% 2em 5%;
2176     padding: 0.5em 1em;
2177     border: none ;
2178     border-top : 1px solid silver;
2179     border-bottom : 1px solid silver;
2180 }
2181
2182 div.exampletitle{
2183     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2184     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2185     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2186     "Times New Roman", serif;
2187     color: #800000 ;
2188     text-shadow: none ;
2189     border: none ;
2190 }
2191
2192 div.fancyvrblabel{
2193     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2194     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2195     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2196     "Times New Roman", serif;
2197     color: #800000 ;
2198     text-shadow: none ;
2199     border: none ;
2200 }
2201
2202
2203
2204 .verse {
2205     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2206     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2207     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2208     "Times New Roman", serif;
2209 }
2210
2211
2212 figure {
2213     margin: 3ex 5% 3ex 5% ;
2214     padding: 1ex 1em 1ex 1em ;
```

```
2215     background-color: #fffcf5 ;
2216     overflow-x: auto ;
2217     border: none ;
2218 /*     border-top: 1px solid silver; */
2219 /*     border-bottom: 1px solid silver; */
2220 }
2221
2222
2223 figcaption , .lstlisting {
2224     border: none ;
2225 /*     border-top: 1px solid silver ; */
2226 /*     border-bottom: 1px solid silver ; */
2227     background-color: #fffcf5 ;
2228 }
2229
2230 .tnotes {
2231     background: #fffcf5 ;
2232 }
2233
2234 .theorem {
2235     background: none ;
2236 }
2237
2238 .minipage {
2239     background-color: #fffcf5 ;
2240     border: none ;
2241 }
2242
2243 div.floatrow figure { border: none ; }
2244
2245 figure figure { border: none ; }
2246
2247
2248 nav.toc, nav.lof, nav.lot, nav.lol {
2249     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2250         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2251         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2252         "Times New Roman", serif;
2253 }
2254
2255 nav.sidetoc {
2256     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2257         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2258         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2259         "Times New Roman", serif;
2260     background-image: linear-gradient(to bottom, #fffcf5, #C0C0C0);
2261     border-radius: 0px 0px 0px 20px ;
2262 }
2263
2264 div.sidetoc{title{
```

```

2265     color: #800000 ;
2266 }
2267
2268 header{
2269     background-color: #e0e0e0 ;
2270     background-image: linear-gradient(to top, #fff5cf, #b0b0b0);
2271     text-align:center ;
2272 }
2273
2274 footer{
2275     background-color: #e0e0e0 ;
2276     background-image: linear-gradient(to bottom, #fff5cf, #b0b0b0);
2277     padding: 2ex 1em 2ex 1em ;
2278     clear:right ;
2279     text-align:left ;
2280 }
2281
2282 nav.botnavigation {
2283     background: #dedcd5 ;
2284     border-top: 1px solid black ;
2285 }
2286 \end{filecontents*}
2287 % \end{Verbatim}% for syntax highlighting
2288 \end{warpprint}

```

## 28.7 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.

```

2289 \begin{warpprint}
2290 \begin{filecontents*}{sample_project.css}
2291 /* ( --- Start of project.css --- ) */
2292 /* A sample project-specific CSS file for lwarp --- ) */
2293
2294 /* Load default lwarp settings: */
2295 @import url("lwarp.css") ;
2296 /* or lwarp_formal.css, lwarp_sagebrush.css */
2297
2298 /* Project-specific CSS setting follow here. */
2299 /* . . . */
2300
2301 /* ( --- End of project.css --- ) */
2302 \end{filecontents*}
2303 % \end{Verbatim}% for syntax highlighting
2304 \end{warpprint}

```

## 28.8 lwarp.xdy

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

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

```

2305 \begin{warpprint}
2306 \begin{filecontents*}{lwarp.xdy}
2307 (require "tex/inputenc/latin.xdy")
2308 (merge-rule "\\PS *" "Postscript")
2309 (require "texindy.xdy")
2310 (require "page-ranges.xdy")
2311 (require "book-order.xdy")
2312 (markup-locref :open "\hyperindexref{" :close "{")
2313 \end{filecontents*}
2314 % \end{Verbatim}% for syntax highlighting
2315 \end{warpprint}

```

## 28.9 lwarp\_mathjax.txt

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

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

```

2316 \begin{warpprint}
2317 \begin{filecontents*}{lwarp_mathjax.txt}
2318 <!-- https://groups.google.com/forum/#!topic/
2319                               mathjax-users/jUtewUcE2bY -->
2320 <script type="text/x-mathjax-config">
2321 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
2322     var seteqsectionDefault = {name: "", num: 0};
2323     var seteqsections = {}, seteqsection = seteqsectionDefault;
2324     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
2325     var AMS = MathJax.Extension["TeX/AMSmath"];
2326     TEX.Definitions.Add({
2327     macros: {
2328         seteqsection: "mySection",
2329         seteqnumber: "mySetEqNumber"
2330     }
2331     });
2332
2333     PARSE.Augment({
2334     mySection: function (name) {
2335         seteqsection.num = AMS.number;
2336         var n = this.GetArgument(name);

```

```

2337         if (n === "") {
2338             seteqsection = seteqsectionDefault;
2339         } else {
2340             if (!seteqsections["_"+n])
2341                 seteqsections["_"+n] = {name:n, num:0};
2342             seteqsection = seteqsections["_"+n];
2343         }
2344         AMS.number = seteqsection.num;
2345     },
2346     mySetEqNumber: function (name) {
2347         var n = this.GetArgument(name);
2348         if (!n || !n.match(/^ *[0-9]+ *$/))
2349             n = ""; else n = parseInt(n)-1;
2350         <!-- $ syntax highlighting -->
2351         if (n === "" || n < 1)
2352             TEX.Error
2353                 ("Argument to "+name+" should be a positive integer");
2354         AMS.number = n;
2355     }
2356 });
2357 MathJax.Hub.Config({
2358   TeX: {
2359     equationNumbers: {
2360       formatTag: function (n)
2361         {return "("+(seteqsection.name+"."+n).replace(/\./,"")+"}"},
2362       formatID: function (n) {
2363         n = (seteqsection.name+'.'+n).replace
2364             (/[<'>&]/g,"").replace(/\./,"");
2365         return 'mjax-eqn-' + n;
2366       }
2367     }
2368   }
2369 });
2370 });
2371 </script>
2372
2373 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
2374 <script type="text/x-mathjax-config">
2375   MathJax.Ajax.config.path["Contrib"] =
2376     "https://cdn.mathjax.org/mathjax/contrib";
2377 </script>
2378
2379 <!-- https://github.com/mathjax/MathJax-third-party-extensions/
2380                                tree/master/siunitx -->
2381 <script type="text/x-mathjax-config">
2382   MathJax.Hub.Config({
2383     extensions: ["tex2jax.js", "[Contrib]/siunitx/siunitx.js"],
2384     jax: ["input/TeX", "output/HTML-CSS"],
2385     tex2jax: {inlineMath: [["$","$"],["\\(","\\)"]]},
2386     TeX: {extensions: ["AMSmath.js", "AMSsymbols.js", "sinuitx.js"]}

```

```

2387 });
2388 </script>
2389
2390 <script type="text/x-mathjax-config">
2391 MathJax.Hub.Config({
2392     TeX: {
2393         equationNumbers: {
2394             autoNumber: "AMS"
2395         }
2396     }
2397 });
2398 </script>
2399
2400 <!-- Alternative CDN provider: -->
2401 <script type="text/javascript" async
2402 src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?config=TeX-AMS_HTML-full">
2403 </script>
2404
2405 <!-- No longer supported after April 30, 2017: -->
2406 <!--
2407 <script
2408     src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
2409 </script>
2410 -->
2411
2412 \end{filecontents*}
2413 % \end{Verbatim}% for syntax highlighting
2414 \end{warpprint}

```

## 28.10 lwarpmk option

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

```

2415 \begin{LWR@createlwarpmk}

```

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

```

2416 \begin{filecontents*}{lwarpmk.lua}
2417 #!/usr/bin/env texlua
2418
2419 -- Copyright 2016-2017 Brian Dunn
2420
2421 -- Print the usage of the lwarpmk command:
2422
2423 printversion = "v0.36"
2424
2425 function printhelp ()

```

```

2426 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
2427 end
2428
2429 function printusage ()
2430 print ( [[
2431
2432 lwarpmk print [project]: Compile a print version.
2433 lwarpmk printindex [project]: Process the index for the print version.
2434 lwarpmk printglossary [project]: Process the glossary for the print version.
2435 lwarpmk html [project]: Compile an HTML version.
2436 lwarpmk htmlindex [project]: Process the index for the html version.
2437 lwarpmk htmlglossary [project]: Process the glossary for the html version.
2438 lwarpmk again [project]: Touch the source code to trigger recompiles.
2439 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
2440 lwarpmk pdftohtml [project]:
2441     For use with latexmk or a Makefile:
2442     Convert project_html.pdf to project_html.html and
2443     individual HTML files.
2444 lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, .gl*
2445 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
2446 lwarpmk -h: Print this help message.
2447 lwarpmk --help: Print this help message.
2448
2449 ]] )
2450 printconf ()
2451 end
2452
2453 -- Print the format of the configuration file lwarpmk.conf:
2454
2455 function printconf ()
2456 print ( [[
2457 An example lwarpmk.conf or <project>.lwarpmkconf project file:
2458 --
2459 opsystem = "Unix"      (or "Windows")
2460 latexname = "pdflatex" (or "lualatex", or "xelatex")
2461 sourcename = "projectname" (the source-code filename w/o .tex)
2462 homehtmlfilename = "index" (or perhaps the project name)
2463 htmlfilename = "" (or "projectname" - filename prefix)
2464 latexmk = "false" (or "true" to use latexmk to build PDFs)
2465 language = "english" (use a language supported by xindy)
2466 xdyfile = "lwarp.xdy" (or a custom file based on lwarp.xdy)
2467 --
2468 Filenames must contain only letters, numbers, underscore, or dash.
2469 Values must be in "quotes".
2470
2471 ]] ) ;
2472 end
2473
2474
2475 -- Split one large sourcefile into a number of files,

```

```
2476 -- starting with destfile.
2477 -- The file is split at each occurrence of <!--|Start file|newfilename|*
2478
2479 function splitfile (destfile,sourcefile)
2480 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
2481 local sfile = io.open(sourcefile)
2482 io.output(destfile)
2483 for line in sfile:lines() do
2484 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
2485 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
2486 io.output(newfilename) ;
2487 else -- not a splitpoint
2488 io.write (line .. "\n") ;
2489 end
2490 end -- do
2491 io.close(sfile)
2492 end -- function
2493
2494 -- Incorrect value, so print an error and exit.
2495
2496 function cvalueerror ( line, linenum , cvalue )
2497     print ( linenum .. " : " .. line ) ;
2498     print ("lwarpmk: incorrect variable value \"" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
2499     printconf () ;
2500     os.exit(1) ;
2501 end
2502
2503 -- Load settings from the project's "lwarpmk.conf" file:
2504
2505 function loadconf ()
2506 -- Default configuration filename:
2507 local conffile = "lwarpmk.conf"
2508 -- Optional configuration filename:
2509 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
2510 -- Default language:
2511 language = "english"
2512 -- Default xdyfile:
2513 xdyfile = "lwarp.xdy"
2514 -- Verify the file exists:
2515 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
2516 print("lwarpmk: " .. conffile .. " does not exist.")
2517 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
2518 printhelp () ;
2519 os.exit(1) -- exit the entire lwarpmk script
2520 else -- file exists
2521 -- Read the file:
2522 print ("lwarpmk: Reading " .. conffile .. ".")
2523 local cfile = io.open(conffile)
2524 -- Scan each line:
2525 local linenum = 0
```



```

2526 for line in cfile:lines() do -- scan lines
2527     linenum = linenum + 1
2528     i,j,cvarname,cvalue = string.find (line,"([%w-_]*)%s*=%s*\"([%w%-_.]*)\"") ;
2529     -- Error if incorrect enclosing characters:
2530     if ( i == nil ) then
2531         print ( linenum .. " : " .. line ) ;
2532         print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
2533         printconf () ;
2534         os.exit(1) ;
2535     end
2536     if ( cvarname == "opsystem" ) then
2537         -- Verify choice of opsystem:
2538         if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
2539             opsystem = cvalue
2540         else
2541             cvalueerror ( line, linenum , cvalue )
2542         end
2543     elseif ( cvarname == "latexname" ) then
2544         -- Verify choice of LaTeX compiler:
2545         if (
2546             (cvalue == "pdflatex") or
2547             (cvalue == "xelatex") or
2548             (cvalue == "lualatex")
2549         ) then
2550             latexname = cvalue
2551         else
2552             cvalueerror ( line, linenum , cvalue )
2553         end
2554     elseif ( cvarname == "sourcename" ) then sourcename = cvalue
2555     elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
2556     elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
2557     elseif ( cvarname == "latexmk" ) then latexmk = cvalue
2558     elseif ( cvarname == "language" ) then language = cvalue
2559     elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue
2560     else
2561         print ( linenum .. " : " .. line ) ;
2562         print ( "lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. conffile .. ".\n" ) ;
2563         printconf () ;
2564         os.exit(1) ;
2565     end
2566 end -- do scan lines
2567 io.close(cfile)
2568 end -- file exists
2569 -- Select some operating-system commands:
2570 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
2571     rmname = "rm"
2572     mvname = "mv"
2573     touchnamepre = "touch"
2574     touchnamepost = ""
2575     dirslash = "/"

```

```
2576 opquote= "\'"
2577 elseif opsystem=="Windows" then -- For Windows
2578 rmname = "DEL"
2579 mvname = "MOVE"
2580 touchnamepre = "COPY /b"
2581 touchnamepost = "+,,"
2582 dirslash = "\\"
2583 opquote= "\""
2584 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
2585 end --- for Windows
2586
2587 -- set xindycmd according to pdflatex vs xelatex/lualatex:
2588 if ( latexname == "pdflatex" ) then
2589 xindycmd = "texindy -C utf8"
2590 glossarycmd = "xindy -C utf8"
2591 else
2592 xindycmd = "xindy -M texindy -C utf8"
2593 glossarycmd = "xindy -C utf8"
2594 end
2595
2596 end -- loadconf
2597
2598
2599 function refreshdate ()
2600 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
2601 end
2602
2603
2604 -- Scan the LaTeX log file for the phrase "Rerun to get",
2605 -- indicating that the file should be compiled again.
2606 -- Return true if found.
2607
2608 function reruntoget (filesorce)
2609 local fsource = io.open(filesorce)
2610 for line in fsource:lines() do
2611 if ( string.find(line,"Rerun to get") ~= nil ) then
2612 io.close(fsource)
2613 return true
2614 end
2615 end
2616 io.close(fsource)
2617 return false
2618 end
2619
2620
2621 -- Compile one time, return true if should compile again.
2622 -- fsuffix is "" for print, "_html" for HTML output.
2623
2624 function onetime (fsuffix)
2625 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
```

```
2626 err = os.execute(
2627 --      "echo " ..
2628      latexname .. " " .. sourcename..fsuffix )
2629 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2630 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
2631 end
2632
2633
2634 -- Compile up to five times.
2635 -- fsuffix is "" for print, "_html" for HTML output
2636
2637 function manytimes (fsuffix)
2638 if onetime(fsuffix) == true then
2639 if onetime(fsuffix) == true then
2640 if onetime(fsuffix) == true then
2641 if onetime(fsuffix) == true then
2642 if onetime(fsuffix) == true then
2643 end end end end end
2644 end
2645
2646 -- Exit if the given file does not exist.
2647
2648 function verifyfileexists (filename)
2649 if (lfs.attributes ( filename , "modification" ) == nil ) then
2650 print ( "lwarpmk: " .. filename .. " not found." ) ;
2651 os.exit (1) ;
2652 end
2653 end
2654
2655
2656 -- Convert <project>_html.pdf into HTML files:
2657
2658 function pdftohtml ()
2659 -- Convert to text:
2660 print ("lwarpmk: Converting " .. sourcename
2661      .. "_html.pdf to " .. sourcename .. "_html.html")
2662 os.execute("pdftotext -enc UTF-8 -npgbrk -layout "
2663      .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
2664 -- Split the result into individual HTML files:
2665 splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
2666 end
2667
2668
2669 -- Remove auxiliary files:
2670
2671 function removeaux ()
2672 os.execute ( rmname .. " " ..
2673      sourcename .. ".aux " .. sourcename .. "_html.aux " ..
2674      sourcename .. ".toc " .. sourcename .. "_html.toc " ..
2675      sourcename .. ".lof " .. sourcename .. "_html.lof " ..
```

```

2676     sourcename ..".lot " .. sourcename .. "_html.lot " ..
2677     sourcename ..".idx " .. sourcename .. "_html.idx " ..
2678     sourcename ..".ind " .. sourcename .. "_html.ind " ..
2679     sourcename ..".log " .. sourcename .. "_html.log " ..
2680     sourcename ..".gl* " .. sourcename .. "_html.gl* "
2681     )
2682 end
2683
2684
2685
2686 -- Create lateximages based on lateximages.txt:
2687 function createlateximages ()
2688 print ("lwarpmk: Creating lateximages.")
2689 local limagesfile = io.open("lateximages.txt")
2690 -- Create the lateximages directory, ignore error if already exists
2691 err = os.execute("mkdir lateximages")
2692 -- Scan lateximages.txt
2693 for line in limagesfile:lines() do
2694 -- lwimpage is the page number in the PDF which has the image
2695 -- lwimgnum is the sequential lateximage number to assign for the image
2696 i,j,lwimpage,lwimgnum = string.find (line,"|(.)|(.)|")
2697 -- For each entry:
2698 if ( i~=nil ) then
2699 -- Separate out the image into its own single-page pdf:
2700 err = os.execute(
2701 "pdfseparate -f " .. lwimpage .. " -l " ..
2702 lwimpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
2703 -- Crop the image:
2704 err = os.execute(
2705 "pdftocrop -- hires lateximagetemp-" .. lwimpage .. ".pdf lateximage-" .. lwimgnum .. ".pdf")
2706 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2707 -- Convert the image to svg:
2708 err = os.execute(
2709 "pdftocairo -svg lateximage-" .. lwimgnum .. ".pdf lateximage-" .. lwimgnum .. ".svg")
2710 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2711 -- Move the result into lateximages/:
2712 err = os.execute(
2713 mvname .. " lateximage-" .. lwimgnum .. ".svg lateximages" .. dirslash )
2714 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2715 -- Remove the temporary files:
2716 err = os.execute(
2717 rmname .. " lateximage-" .. lwimgnum .. ".pdf lateximagetemp-" .. lwimpage .. ".pdf")
2718 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2719 end
2720 end -- do
2721 io.close(limagesfile)
2722 end -- function
2723
2724
2725 -- Use latexmk to compile source and index:

```

```

2726 -- fsuffix is "" for print, or "_html" for HTML
2727 function compilelatexmk ( fsuffix )
2728     -- The recorder option is required to detect changes in <project>.tex
2729     -- while we are loading <project>_html.tex.
2730     err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
2731         .. "-e "
2732         .. opquote
2733         .. "$makeindex = q/" -- $
2734         .. xindycmd
2735         .. " -M " .. xdyfile
2736         .. " -L " .. language .. " /"
2737         .. opquote
2738         .. " -pdflatex=\"\" .. latexname .. " %O %S\" "
2739         .. sourcename..fsuffix ..".tex" ) ;
2740     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2741 end
2742
2743
2744
2745 -- lwarpmk --version :
2746
2747 if (arg[1] == "--version") then
2748     print ( "lwarpmk: " .. printversion )
2749
2750 else -- not -- version
2751
2752 -- print intro:
2753
2754 print ("lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package.")
2755
2756 -- lwarpmk print:
2757
2758 if arg[1] == "print" then
2759     loadconf ( )
2760     if ( latexmk == "true" ) then
2761         compilelatexmk ( "" )
2762         print ("lwarpmk: Done.")
2763     else -- not latexmk
2764         verifyfileexists (sourcename .. ".tex") ;
2765         -- See if up to date:
2766         if (
2767             ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
2768             (
2769                 lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2770                 lfs.attributes ( sourcename .. ".pdf" , "modification" )
2771             )
2772         ) then
2773             -- Recompile if not yet up to date:
2774             manytimes("")
2775             print ("lwarpmk: Done.") ;

```

```
2776     else
2777         print ("lwarpmk: " .. sourcename .. ".pdf is up to date.");
2778     end
2779 end -- not latexmk
2780
2781 -- lwarp printindex:
2782 -- Compile the index then touch the source
2783 -- to trigger a recompile of the document:
2784
2785 elseif arg[1] == "printindex" then
2786 loadconf ()
2787 print ("lwarpmk: Processing the index.")
2788 os.execute(
2789     xindycmd
2790     .. " -M " .. xdyfile
2791     .. " -L " .. language
2792     .. " " .. sourcename .. ".idx")
2793 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2794 refreshdate ()
2795 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2796 print ("lwarpmk: Done.")
2797
2798 -- lwarp printglossary:
2799 -- Compile the glossary then touch the source
2800 -- to trigger a recompile of the document:
2801
2802 elseif arg[1] == "printglossary" then
2803 loadconf ()
2804 print ("lwarpmk: Processing the glossary.")
2805
2806 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
2807     " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
2808     .. sourcename .. ".glo")
2809 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2810 refreshdate ()
2811 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2812 print ("lwarpmk: Done.")
2813
2814 -- lwarpmk html:
2815
2816 elseif arg[1] == "html" then
2817 loadconf ()
2818 if ( latexmk == "true" ) then
2819     compilelatexmk ("_html")
2820     pdftohtml ()
2821     print ("lwarpmk: Done.")
2822 else -- not latexmk
2823     verifyfileexists ( sourcename .. ".tex" ) ;
2824     -- See if exists and is up to date:
2825     if (
```

```
2826      ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
2827      (
2828          lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2829          lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
2830      )
2831  ) then
2832      -- Recompile if not yet up to date:
2833      manytimes("_html")
2834      pdftohtml ()
2835      print ("lwarpmk: Done.")
2836  else
2837      print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
2838  end
2839 end -- not latexmk
2840
2841 elseif arg[1] == "pdftohtml" then
2842     loadconf ()
2843     pdftohtml ()
2844
2845 -- lwarpmk htmlindex:
2846 -- Compile the index then touch the source
2847 -- to trigger a recompile of the document:
2848
2849 elseif arg[1] == "htmlindex" then
2850 loadconf ()
2851 print ("lwarpmk: Processing the index.")
2852 os.execute(
2853     xindy cmd
2854     .. " -M " .. xdyfile
2855     .. " -L " .. language
2856     .. " " .. sourcename .. "_html.idx"
2857 )
2858 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2859 refreshdate ()
2860 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2861 print ("lwarpmk: Done.")
2862
2863 -- lwarpmk htmlglossary:
2864 -- Compile the glossary then touch the source
2865 -- to trigger a recompile of the document:
2866
2867 elseif arg[1] == "htmlglossary" then
2868 loadconf ()
2869 print ("lwarpmk: Processing the glossary.")
2870
2871 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
2872     "_html -t " .. sourcename .. "_html.glg -o " .. sourcename ..
2873     "_html.gls " .. sourcename .. "_html.glo")
2874
2875 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
```

```
2876 refreshdate ()
2877 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2878 print ("lwarpmk: Done.")
2879
2880 -- lwarpmk limages:
2881 -- Scan the lateximages.txt file to create lateximages,
2882 -- then touch the source to trigger a recompile.
2883
2884 elseif arg[1] == "limages" then
2885 loadconf ()
2886 print ("lwarpmk: Processing images.")
2887 createlateximages ()
2888 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2889 refreshdate ()
2890 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2891 print ("lwarpmk: Done.")
2892
2893 -- lwarpmk again:
2894 -- Touch the source to trigger a recompile.
2895
2896 elseif arg[1] == "again" then
2897 loadconf ()
2898 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2899 refreshdate ()
2900 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2901 print ("lwarpmk: Done.")
2902
2903 -- lwarpmk clean:
2904 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, .gl*
2905
2906 elseif arg[1] == "clean" then
2907 loadconf ()
2908 removeaux ()
2909 print ("lwarpmk: Done.")
2910
2911 -- lwarpmk cleanall
2912 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, .gl*
2913 -- and also project.pdf, *.html
2914
2915 elseif arg[1] == "cleanall" then
2916 loadconf ()
2917 removeaux ()
2918 os.execute ( rmname .. " " ..
2919     sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
2920     "*.html"
2921 )
2922 print ("lwarpmk: Done.")
2923
2924 -- lwarpmk with no argument :
2925
```



```

2926 elseif (arg[1] == nil) then
2927 printhelp ()
2928
2929 -- lwarpmk -h or lwarpmk --help :
2930
2931 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
2932 printusage ()
2933
2934 else
2935 print ("lwarpmk: Unknown command \""..arg[1].."\".\n")
2936 printhelp ()
2937 end
2938
2939 end -- not --version
2940 \end{filecontents*}
2941 % \end{Verbatim}% for syntax highlighting
2942 \end{LWR@createlwarpmk}

```

## 29 Stacks

for HTML output: 2943 \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.

### 29.1 Assigning depths

initial depths for empty stack entries:

```
2944 \newcommand*{\LWR@depthnone}{-5}
```

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

```
2945 \newcommand*{\LWR@depthfinished}{-4}
```

```

2946 \newcommand*{\LWR@depthpart}{-1}
2947 \newcommand*{\LWR@depthchapter}{0}
2948 \newcommand*{\LWR@depthsection}{1}
2949 \newcommand*{\LWR@depthsubsection}{2}
2950 \newcommand*{\LWR@depthsubsubsection}{3}
2951 \newcommand*{\LWR@depthparagraph}{4}
2952 \newcommand*{\LWR@depthsubparagraph}{5}

```

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

```

2953 \newcommand*{\LWR@depthlist}{6}

```

used by `\item`:

```

2954 \newcommand*{\LWR@depthlistitem}{7}

```

## 29.2 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.

```

2955 \newcommand*{\LWR@closeone}{}% top of the stack
2956 \newcommand*{\LWR@closetwo}{}
2957 \newcommand*{\LWR@closethree}{}
2958 \newcommand*{\LWR@closefour}{}
2959 \newcommand*{\LWR@closefive}{}
2960 \newcommand*{\LWR@closesix}{}
2961 \newcommand*{\LWR@closeseven}{}
2962 \newcommand*{\LWR@closeeight}{}
2963 \newcommand*{\LWR@closenine}{}
2964 \newcommand*{\LWR@closeten}{}
2965 \newcommand*{\LWR@closeeleven}{}
2966 \newcommand*{\LWR@closetwelve}{}

```

## 29.3 Closing depths

A stack to record the depth of each level:



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

---

*Ex:*

---

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

---

```
2967 \newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
2968 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
2969 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
2970 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
2971 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
2972 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
2973 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
2974 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
2975 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
2976 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}
2977 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
2978 \newcommand*{\LWR@closedephtwelve}{\LWR@depthnone}
```

## 29.4 Pushing and popping the stack

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

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

```
2979 \NewDocumentCommand{\pushclose}{m m}
2980 {
2981 \let\LWR@closetwelve\LWR@closeeleven
2982 \let\LWR@closeeleven\LWR@closeten
2983 \let\LWR@closeten\LWR@closenine
2984 \let\LWR@closenine\LWR@closeeight
2985 \let\LWR@closeeight\LWR@closeseven
2986 \let\LWR@closeseven\LWR@closesix
2987 \let\LWR@closesix\LWR@closefive
2988 \let\LWR@closefive\LWR@closefour
2989 \let\LWR@closefour\LWR@closethree
2990 \let\LWR@closethree\LWR@closetwo
2991 \let\LWR@closetwo\LWR@closeone
2992 \let\LWR@closeone#1
2993 \let\LWR@closedephtwelve\LWR@closedeptheleven
2994 \let\LWR@closedeptheleven\LWR@closedepthten
2995 \let\LWR@closedepthten\LWR@closedepthnine
```

```

2996 \let\LWR@closedepthnine\LWR@closedeptheight
2997 \let\LWR@closedeptheight\LWR@closedepthseven
2998 \let\LWR@closedepthseven\LWR@closedepthsix
2999 \let\LWR@closedepthsix\LWR@closedepthfive
3000 \let\LWR@closedepthfive\LWR@closedepthfour
3001 \let\LWR@closedepthfour\LWR@closedepththree
3002 \let\LWR@closedepththree\LWR@closedepthtwo
3003 \let\LWR@closedepthtwo\LWR@closedepthone
3004 \let\LWR@closedepthone#2
3005 }

```

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

```

3006 \newcommand*{\popclose}
3007 {
3008 \let\LWR@closeone\LWR@closetwo
3009 \let\LWR@closetwo\LWR@closethree
3010 \let\LWR@closethree\LWR@closefour
3011 \let\LWR@closefour\LWR@closefive
3012 \let\LWR@closefive\LWR@closesix
3013 \let\LWR@closesix\LWR@closeseven
3014 \let\LWR@closeseven\LWR@closeeight
3015 \let\LWR@closeeight\LWR@closenine
3016 \let\LWR@closenine\LWR@closeten
3017 \let\LWR@closeten\LWR@closeeleven
3018 \let\LWR@closeeleven\LWR@closetwelve
3019 \let\LWR@closedepthone\LWR@closedepthtwo
3020 \let\LWR@closedepthtwo\LWR@closedepththree
3021 \let\LWR@closedepththree\LWR@closedepthfour
3022 \let\LWR@closedepthfour\LWR@closedepthfive
3023 \let\LWR@closedepthfive\LWR@closedepthsix
3024 \let\LWR@closedepthsix\LWR@closedepthseven
3025 \let\LWR@closedepthseven\LWR@closedeptheight
3026 \let\LWR@closedeptheight\LWR@closedepthnine
3027 \let\LWR@closedepthnine\LWR@closedepthten
3028 \let\LWR@closedepthten\LWR@closedeptheleven
3029 \let\LWR@closedeptheleven\LWR@closedepthtwelve
3030 }

3031 \end{warpHTML}

```

## 30 Data arrays

These macros are similar to the `arrayjobx` package, except that `\LWR@setexpparray`'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: 3032 \begin{warpHTML}

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

```
3033 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
3034 \ifthenelse{\isempty{#3}}{%
3035 {\csdef{#1#2}{}}
3036 {\expandafter\edef\csname #1#2\endcsname{\expandonce#3}}%
3037 }
```

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

```
3038 \newcommand*{\LWR@getexparray}[2]{\csuse{#1#2}}
```

```
3039 \end{warpHTML}
```

## 31 HTML entities

for HTML output: 3040 \begin{warpHTML}

HTML entites and HTML Unicode entities:

```
3041 \let\LWR@origampersand\&
```

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

```
3042 \newcommand*{\HTMLentity}[1]{%
3043 % \LWR@traceinfo{HTMLentity \detokenize{#1}}%
3044 \begingroup%
3045 \LWR@FBcancel%
3046 \LWR@origampersand#1;%
3047 \endgroup
3048 % \LWR@traceinfo{HTMLentity done}%
3049 }
```

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

```
3050 \newcommand*{\HTMLunicode}[1]{\HTMLentity{\x#1}}
```

\&

```
3051 \renewcommand*{\&}{\HTMLentity{amp}}
```

\textless

\textgreater

```
3052 \let\LWR@origtextless\textless
3053 \renewcommand*{\textless}{\HTMLentity{lt}}
3054
3055 \let\LWR@origtextgreater\textgreater
3056 \renewcommand*{\textgreater}{\HTMLentity{gt}}

3057 \end{warpHTML}
```

## 32 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:** 3058 \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.

```
3059 \providecommand*{\BaseJobname}{\jobname}
```

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

```
3060 \providecommand*{\HTMLFilename}{}
```

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

```
3061 \providecommand*{\HomeHTMLFilename}{\BaseJobname}
```

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

Sets the file number for the next file to be generated. 0 is the home page. Use just before the next sectioning command, and set it to one less than the desired

number of the next section. May be used to generate numbered groups of nodes such as 100+ for one chapter, 200+ for another chapter, etc.

```
3062 \newcommand*{\SetHTMLFileNumber}[1]{%
3063 \setcounter{LWR@htmlfilenumber}{#1}%
3064 }
```

Bool FileSectionNames Selects how to create HTML file names.

Defaults to use section names in the filenames.

```
3065 \newbool{FileSectionNames}
3066 \booltrue{FileSectionNames}

3067 \end{warpall}
```

**for HTML output:** 3068 \begin{warpHTML}

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

```
3069 \newcounter{LWR@htmlfilenumber}
3070 \setcounter{LWR@htmlfilenumber}{0}
```

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

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

```
3071 \newcommand*{\LWR@htmlsectionfilename}[1]{%
3072 \LWR@traceinfo{LWR@htmlsectionfilename A}%
```

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

```
3073 \LWR@traceinfo{about to assign temp}%
3074 \edef\LWR@tempone{#1}%
3075 \LWR@traceinfo{about to compare with ??}%
3076 \ifthenelse{\equal{\LWR@tempone}{??}}{%
3077 {\LWR@traceinfo{found ??}}}%
3078 {\LWR@traceinfo{not found ??}}}%
3079 \LWR@traceinfo{about to compare with zero or empty}%
3080 \ifthenelse{%
3081   \equal{\LWR@tempone}{0}}{%
3082   \OR \equal{\LWR@tempone}{}}{%
3083   \OR \equal{\LWR@tempone}{??}}%
3084 }%
3085 {%
3086   \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
```

```

3087     \HomeHTMLFilename.html%
3088 }%

```

For a L<sup>A</sup>T<sub>E</sub>X section named “Index” or “index” without a prefix, create a filename with a leading underscore to avoid colliding with the HTML filename `index.html`:

```

3089 {%
3090     \LWR@traceinfo{\LWR@htmlsectionfilename C \LWR@tempone}%
3091     \ifthenelse{%
3092         \equal{\HTMLFilename}{ } \AND
3093         \equal{\LWR@tempone}{Index} \OR
3094         \equal{\LWR@tempone}{index}%
3095     }%
3096     {%
3097         \LWR@traceinfo{prefixing the index name with an underscore.}%
3098         \_#1.html%
3099     }%

```

Otherwise, create a filename with the chosen prefix:

```

3100     {\HTMLFilename#1.html}%
3101 }%
3102 \LWR@traceinfo{\LWR@htmlsectionfilename Z}%
3103 }

```

```
\LWR@htmlrefsectionfilename {\label}
```

Prints the filename for the given label

```

3104 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
3105     \LWR@traceinfo{\LWR@htmlrefsectionfilename A: !#1!}%
3106     \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
3107     \LWR@traceinfo{\LWR@htmlrefsectionfilename B}%
3108 }

3109 \end{warpHTML}

```

### 33 Homepage link

**for HTML output:** 3110 \begin{warpHTML}

`\LinkHome` `\LinkHome` may be used wherever you wish to place a link back to the homepage. The filename must be detokenized for underscores.

```
3111 \newcommand*{\LinkHome}{%
```



```

3112 \LWR@subhyperrefclass{%
3113 \HomeHTMLFilename.html}%
3114 {Home}{linkhome}%
3115 }

```

**\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.

```

3116 \newcommand*{\LWR@topnavigation}{
3117 \LWR@htmlelementclassline{nav}{topnavigation}{\LinkHome}
3118 }

```

**\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.

```

3119 \newcommand*{\LWR@botnavigation}{
3120 \LWR@htmlelementclassline{nav}{botnavigation}{\LinkHome}
3121 }

```

```

3122 \end{warpHTML}

```

## 34 \PrintStack diagnostic tool



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

**for HTML output:** 3123 \begin{warpHTML}

**\PrintStack** Prints the closedepth stack.

```

3124 \newcommand*{\PrintStack}{
3125 \LWR@startpars
3126 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
3127 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
3128 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
3129 \LWR@closedephten{} \LWR@closedeptheleven{} \LWR@closedephtwelve{}
3130 }

```

```

3131 \end{warpHTML}

```

## 35 Closing stack levels

for HTML output: 3132 \begin{warpHTML}

Close one nested level:

```
3133 \newcommand*{\LWR@closeoneprevious}{%
3134
3135 \LWR@closeone{
3136
3137 \popclose{
3138 }
```

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

```
3139 \newcommand*{\LWR@closeprevious}[1]{
```

Close any pending paragraph:

```
3140 \LWR@stoppars
```

Close anything nested deeper than the desired depth:

```
3141 \whiledo{\not\(\LWR@closedepthone<\#1\)}{\LWR@closeoneprevious}
3142 }
```

```
3143 \end{warpHTML}
```

## 36 PDF pages and styles

for HTML output: 3144 \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.

```
3145 \newcommand{\LWR@forcenewpage}{%
3146 \LWR@stoppars\LWR@orignewpage\LWR@startpars%
3147 }
```

\pagestyle and \thispagestyle are nullified for HTML output.

```

\pagestyle   {\style}

3148 \renewcommand*{\pagestyle}[1]{}

\thispagestyle {\style}

3149 \renewcommand*{\thispagestyle}[1]{}

\pagenumbering {\commands}

3150 \renewcommand*{\pagenumbering}[1]{}

3151 \end{warpHTML}

```

## 37 HTML tags, spans, divs, elements

for HTML output: 3152 \begin{warpHTML}

### 37.1 Mapping L<sup>A</sup>T<sub>E</sub>X Sections to HTML Sections

```

3153 \newcommand*{\LWR@tagpart}{h2}
3154 \newcommand*{\LWR@tagpartend}{/h2}
3155 \newcommand*{\LWR@tagchapter}{h3}
3156 \newcommand*{\LWR@tagchapterend}{/h3}
3157 \newcommand*{\LWR@tagsection}{h4}
3158 \newcommand*{\LWR@tagsectionend}{/h4}
3159 \newcommand*{\LWR@tagsubsection}{h5}
3160 \newcommand*{\LWR@tagsubsectionend}{/h5}
3161 \newcommand*{\LWR@tagsubsubsection}{h6}
3162 \newcommand*{\LWR@tagsubsubsectionend}{/h6}
3163 \newcommand*{\LWR@tagparagraph}{span class="paragraph"{} }
3164 \newcommand*{\LWR@tagparagraphend}{/span}
3165 \newcommand*{\LWR@tagsubparagraph}{span class="subparagraph"{} }
3166 \newcommand*{\LWR@tagsubparagraphend}{/span}
3167
3168 \newcommand*{\LWR@tagregularparagraph}{p}

```

### 37.2 Babel-French

Adjust babel-french for HTML spaces. So far, this only works for pdf<sub>l</sub>atex and xelatex.

(Based on original code by DANIEL FLIPO.)

```

3169 \providecommand*{\LWR@FBcancel}{-}
3170
3171 \AtBeginDocument{
3172 \ifundefined{frenchbsetup}
3173 {}
3174 {
3175   \frenchbsetup{FrenchFootnotes=false}
3176   % OR: redefine \insertfootnotemarkFB?
3177   \LetLtxMacro\LWR@FBcancel\NoAutoSpacing
3178   \renewcommand*{\FBcolonspace}{%
3179     \begingroup%
3180     \LWR@FBcancel%
3181     \LWR@origampersand{}~\,
3182     \endgroup%
3183   }
3184   \renewcommand*{\FBthinspace}{%
3185     \begingroup%
3186     \LWR@FBcancel%
3187     \LWR@origampersand{\#x202f}% \,
3188     \endgroup%
3189   }
3190   \renewcommand*{\FBguillspace}{%
3191     \begingroup%
3192     \LWR@FBcancel%
3193     \LWR@origampersand{}~\, for \og xyz \fg{}
3194     \endgroup%
3195   }
3196   \DeclareDocumentCommand{\FBmedkern}{-}{%
3197     \begingroup%
3198     \LWR@FBcancel%
3199     \LWR@origampersand{\#x202f}% \,
3200     \endgroup%
3201   }
3202   \DeclareDocumentCommand{\FBthickkern}{-}{%
3203     \begingroup%
3204     \LWR@FBcancel%
3205     \LWR@origampersand{}~\,
3206     \endgroup%
3207   }
3208   \ifFBunicode
3209   \else
3210     \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
3211     \DeclareTextCommandDefault{\FBtextellipsis}{\textellipsis\xspace}
3212   \fi
3213 }
3214 }
```

### 37.3 HTML tags

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

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

```

3215 \newcommand*{\LWR@htmltagc}[1]{%
3216 {%
3217 % \LWR@traceinfo{\LWR@htmltagc \detokenize{#1}}%
3218 \begingroup%
3219 \LWR@FBcancel%
3220 \ifmode\else\protect\LWR@origttfamily\fi%
3221 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
3222 \endgroup%
3223 }%
3224 }
```

Env `LWR@nestspan` Disable minipage, `\parbox`, and HTML `<div>`s inside a `<span>`.

⚠ `\begin{LWR@nestspan}` must follow the opening `<span>` tag to allow a paragraph to start if the span is at the beginning of a new paragraph.

⚠ `\end{LWR@nestspan}` must follow the `</span>` or a `<p>` may appear inside the span.

```

3225 \newcommand*{\LWR@nestspanitem}{%
3226 \ifnewlist\else{\LWR@htmltagc{br /}}\fi%
3227 \LWR@origitem%
3228 }
3229
3230 \newenvironment*{LWR@nestspan}
3231 {%
3232 \LWR@traceinfo{LWR@nestspan}%
3233 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
3234 {%
3235   \LWR@traceinfo{LWR@nestspan: inside a lateximage}%
3236 }%
3237 {% not in a lateximage
3238   \LWR@traceinfo{LWR@nestspan: NOT inside a lateximage}%
3239   \addtocounter{LWR@spandepth}{1}%
3240   \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
3241   \RenewDocumentEnvironment{BlockClass}{o m}{-}{-}%
3242   \renewcommand{\BlockClassSingle}[2]{##2}%
3243   \renewcommand{\LWR@forcenewpage}{-}%
3244   \renewcommand{\LWR@itemizestart}{%
3245     \let\item\LWR@nestspanitem%
3246   }%
3247   \renewcommand{\LWR@itemizeend}{-}%
3248   \renewcommand{\LWR@enumeratestart}{%
```

```

3249      \let\item\LWR@nestspanitem%
3250    }%
3251    \renewcommand{\LWR@enumerateend}{}%
3252    \renewcommand{\LWR@descriptionstart}{%
3253      \let\item\LWR@nestspanitem%
3254    }%
3255    \renewcommand{\LWR@descriptionend}{}%
3256 }% not in a lateximage
3257 }%
3258 {%
3259 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}{>}{0}}%
3260 {}%
3261 {\addtocounter{\LWR@spandepth}{-1}}%
3262 \LWR@traceinfo{\LWR@nestspan: done}%
3263 }
3264
3265 \AfterEndEnvironment{\LWR@nestspan}{\global\let\par\LWR@closeparagraph}

```

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



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

```

3266 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
3267 \LWR@ensuredoingapar%
3268 \LWR@htmltagc{#1}%
3269 \begin{\LWR@nestspan}%
3270 #2%
3271 \LWR@htmltagc{/#1}%
3272 \end{\LWR@nestspan}%
3273 }

```

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

```

3274 \NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{%
3275 \LWR@ensuredoingapar%
3276 \LWR@subhtmlclass{span}[#1]{#2}%
3277 \begin{\LWR@nestspan}%
3278 #3%
3279 \LWR@htmltagc{/span}%
3280 \end{\LWR@nestspan}%
3281 }

```

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

Print an HTML tag: `<tag>`

```

3282 \newcommand*{\LWR@htmltagb}[1]{%
3283 \LWR@htmltagc{#1}%
3284 \endgroup%
3285 }
3286
3287 \newcommand*{\LWR@htmltag}{%
3288 \begingroup\catcode'\_ =12
3289 \LWR@htmltagb%
3290 }

```

## 37.4 Block tags and comments

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

```

\LWR@htmlopencomment
\LWR@htmlclosecomment
3291 \newcommand*{\LWR@htmlopencomment}{%
3292 {%
3293 % \LWR@traceinfo{\LWR@htmlopencomment}%
3294 \begingroup%
3295 \LWR@FBcancel%
3296 \ifmmode\else\protect\LWR@origttfamily\fi%
3297 \LWR@origtextless{}!{-}{-}%
3298 \endgroup%
3299 }%
3300 }
3301
3302 \newcommand*{\LWR@htmlclosecomment}{%
3303 {%
3304 % \LWR@traceinfo{\LWR@htmlclosecomment}%
3305 \begingroup%
3306 \LWR@FBcancel%
3307 \ifmmode\else\protect\LWR@origttfamily\fi%
3308 {-}{-}\LWR@origtextgreater{}%
3309 \endgroup%
3310 }%
3311 }

```

`\LWR@htmlcomment`    `{\comment}`

```

3312 \newcommand{\LWR@htmlcomment}[1]{%
3313 \LWR@htmlopencomment}%
3314 {%
3315 \LWR@origttfamily% break ligatures
3316 #1%

```

```

3317 }%
3318 \LWR@htmlclosecomment{}}

```

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

```

3319 \newcommand{\LWR@htmlblockcommentb}[1]
3320 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
3321
3322 \newcommand{\LWR@htmlblockcomment}
3323 {%
3324 \begingroup\catcode'\_ =12%
3325 \LWR@htmlblockcommentb%
3326 }

```

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

```

3327 \newcommand*\LWR@htmlblocktag[1]{%
3328 \LWR@stoppars%
3329 \LWR@htmltag{#1}%
3330 \LWR@startpars%
3331 }

```

### 37.5 Div class and element class

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

Factored and reused in several places.

The trailing spaces allow more places for a line break.

```

3332 \NewDocumentCommand{\LWR@subhtmlclass}{m o m}{%
3333 \IfValueTF{#2}%
3334 {% option
3335   \ifthenelse{\equal{#2}{}}{%
3336     {\LWR@htmltag{#1 class="#3" }}% empty option
3337     {\LWR@htmltag{#1 class="#3" style="#2" }}% non-empty option
3338 }% option
3339 {\LWR@htmltag{#1 class="#3" }}% no option
3340 }

```

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

```

3341 \NewDocumentCommand{\LWR@htmlclass}{m o m}{%
3342 \LWR@stoppars%
3343 \LWR@subhtmlclass{#1}[#2]{#3}%

```



```
3344 \LWR@startpars%
3345 }
```

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

```
3346 \newcommand*{\LWR@htmlelementclassend}[2]{%
3347 \LWR@stoppars%
3348 \LWR@htmltag{/#1}%
3349 \ifbool{HTMLDebugComments}{%
3350     \LWR@htmlcomment{End of #1 ‘‘#2’’}%
3351 }{}%
3352 \LWR@startpars%
3353 }
```

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

```
3354 \NewDocumentCommand{\LWR@htmldivclass}{o m}{%
3355 \LWR@htmlelementclass{div}[#1]{#2}%
3356 }
```

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

```
3357 \newcommand*{\LWR@htmldivclassend}[1]{%
3358 \LWR@htmlelementclassend{div}[#1]%
3359 }
```

## 37.6 Single-line elements

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

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

```
3360 \NewDocumentCommand{\LWR@htmlelementclassline}{m o m +m}{%
3361 \LWR@stoppars
3362 \LWR@subhtmlelementclass{#1}[#2]{#3}%
3363 #4%
3364 \LWR@htmltag{/#1}
3365 \LWR@startpars
3366 }
```

## 37.7 HTML5 semantic elements

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

```
3367 \newcommand*\LWR@htmlelement}[1]{%
3368 \LWR@htmlblocktag{#1}
3369 }
```

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

```
3370 \newcommand*\LWR@htmlelementend}[1]{%
3371 \LWR@stoppars
3372 \LWR@htmltag{/#1}
3373 \LWR@startpars
3374 }
3375
3376 \end{warpHTML}
```

## 37.8 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.

For other direct-formatting commands, see section [70](#).

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

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

**for HTML output:**

```
3377 \begin{warpHTML}
3378 \NewDocumentEnvironment{BlockClass}{o m}{%
3379 {
3380 \LWR@htmldivclass[#1]{#2}
3381 }
3382 {
3383 \LWR@htmldivclassend{#2}
3384 }
3385 \end{warpHTML}
```

**for PRINT output:**

```
3386 \begin{warpprint}
3387 \NewDocumentEnvironment{BlockClass}{o m}{\}{\}%
3388 \end{warpprint}
```

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

for HTML output: 3389 \begin{warpHTML}  
 3390 \newcommand{\BlockClassSingle}[2]{%  
 3391 \LWR@html@element@classline{div}{#1}{#2}%  
 3392 }  
 3393 \end{warpHTML}

for PRINT output: 3394 \begin{warpprint}  
 3395 \newcommand{\BlockClassSingle}[2]{#2}  
 3396 \end{warpprint}

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

for HTML output: 3397 \begin{warpHTML}  
 3398 \NewDocumentCommand{\InlineClass}{o m +m}{%  
 3399 \LWR@html@spanclass[#1]{#2}{#3}%  
 3400 }  
 3401 \end{warpHTML}

for PRINT output: 3402 \begin{warpprint}  
 3403 \NewDocumentCommand{\InlineClass}{o m +m}{#3}%  
 3404 \end{warpprint}

## 37.9 Closing HTML tags

for HTML output: 3405 \begin{warpHTML}

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

```
3406 \newcommand*{\LWR@printclosepart}  
3407   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}  
3408 \newcommand*{\LWR@printclosechapter}  
3409   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}  
3410 \newcommand*{\LWR@printclosesection}  
3411   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}  
3412 \newcommand*{\LWR@printclosesubsection}  
3413   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}  
3414 \newcommand*{\LWR@printclosesubsubsection}  
3415   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}  
3416 \newcommand*{\LWR@printcloseparagraph}  
3417   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}  
3418 \newcommand*{\LWR@printclosesubparagraph}  
3419   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}
```

Lists require closing HTML tags:

```

3420 \newcommand*{\LWR@printcloselistitem}
3421     {\LWR@htmltag{/li}}
3422 \newcommand*{\LWR@printclosedescitem}
3423     {\LWR@htmltag{/dd}}
3424 \newcommand*{\LWR@printcloseitemize}
3425     {\LWR@htmltag{/ul}}
3426 \newcommand*{\LWR@printcloseenumerate}
3427     {\LWR@htmltag{/ol}}
3428 \newcommand*{\LWR@printclosedescription}
3429     {\LWR@htmltag{/dl}}

3430 \end{warpHTML}

```

## 38 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 39 has high-level commands which allow paragraph-tag generation to start/stop. Even when allowed (`\LWR@doingstartpars`), tags are not generated until a  $\text{\LaTeX}$  paragraph is being used (`\LWR@doingapar`). `LWR@lateximagedepth` is used to prevent nesting tags inside a `lateximage`. `LWR@spandepth` is used to prevent nesting paragraph tags inside a paragraph, which became important inside `\fbox` commands and other spans.

**for HTML output:** 3431 `\begin{warpHTML}`

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

```

3432 \newcounter{LWR@spandepth}
3433 \setcounter{LWR@spandepth}{0}

```

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

```

3434 \newbool{LWR@doingstartpars}
3435 \boolfalse{LWR@doingstartpars}

```

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

```

3436 \newbool{LWR@doingapar}
3437 \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:

```

3438 \newcommand*{\LWR@ensuredoingapar}{%
3439 \ifbool{\LWR@doingstartpars}%
3440 {\global\booltrue{\LWR@doingapar}}%
3441 {}%
3442 }

```

\LWR@openparagraph

```

3443 \newcommand*{\LWR@openparagraph}
3444 {%

```

See if paragraph handling is enabled:

```

3445 \ifbool{\LWR@doingstartpars}%
3446 {% handling pars

```

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

```

3447 \ifthenelse{%
3448 \cnttest{\value{\LWR@lateximagedepth}}{>}{0} \OR%
3449 \cnttest{\value{\LWR@spandepth}}{>}{0}%
3450 }% nested par tags?

```

If so: Do nothing if already started a `lateximage` page. Cannot nest a `lateximage`. Also do nothing if already inside a `<span>`. Do not nest paragraph tags inside a `<span>`.

```

3451 {}% no nested par tags

```

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

```

3452 {% yes nest par tags
3453 \LWR@htmltagc{\LWR@tagregularparagraph}%

```

Manually indent item list labels to avoid left margin intrusion:

See if are nested inside an item list:

```

3454 \ifnumcomp{\@listdepth}{>}{0}%
3455 {%

```

If so, leave some horizontal room in the L<sup>A</sup>T<sub>E</sub>X PDF output for list labels:

```

3456 \LWR@origspace{1in}%
3457 }%
3458 {}%

```

Now have started a paragraph.

```
3459      \global\booltrue{LWR@doingapar}%
```

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

```
3460      \let\par\LWR@closeparagraph%
3461    }% end of yes nest par tags
3462 }% end of handling pars
3463 {}% not handling pars
3464 }
```

\LWR@closeparagraph

```
3465 \newcommand*{\LWR@closeparagraph}
3466 {%
```

See if paragraph handling is enabled:

```
3467 \ifbool{LWR@doingapar}%
```

If currently in paragraph mode:

```
3468 {% handling pars
```

See if already started a lateximage or a <span>:

```
3469   \ifthenelse{%
3470     \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3471     \cnttest{\value{LWR@spandepth}}{>}{0}%
3472   }%
```

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

```
3473   {% no nested par tags
3474     \ifthenelse{%
3475       \cnttest{\value{LWR@spandepth}}{>}{0}%
3476       \AND%
3477       \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3478     }%
3479     {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
3480     {}%
3481   }% no nested par tags
```

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

```
3482      {% yes nest par tags
```

Print a closing tag and some extra vertical space:

```
3483      \unskip%
3484      \LWR@htmltagc{/LWR@tagregularparagraph}%
3485      \LWR@orignewline%
```

No longer doing a paragraph:

```
3486      \global\boolfalse{LWR@doingapar}%
3487 % Disable the special \env{minipage} \& \cs{hspace} interaction
3488 % until a new minipage is found:
3489 %   \begin{macrocode}
3490      \global\boolfalse{LWR@minipagethispar}%
3491      }% end of yes nest par tags
3492 }% end of handling pars
```

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

```
3493 {% not handling pars
3494     \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
3495     {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
3496     {}%
3497 }% not handling pars
```

Finish with regular paragraph processing

```
3498 \LWR@origpar%
3499 }
```

```
3500 \end{warpHTML}
```

## 39 Paragraph start/stop handling

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

Section 38 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:** 3501 \begin{warpHTML}

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

```
3502 \newcommand*{\LWR@startpars}%
3503 {%
```

Ignore if inside a span:

```
3504 \ifthenelse{\cnttest{\value{\LWR@spandepth}}{>}{0}}%
3505 {}%
3506 {%
```

See if currently handling HTML paragraphs:

```
3507 \ifbool{\LWR@doingstartpars}%
```

If already in paragraph mode, do nothing.

```
3508 {}%
```

If not currently in paragraph mode:

```
3509 {%
```

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

```
3510 \PushPreHook{par}{\LWR@openparagraph}%
```

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

```
3511 \let\par\LWR@closeparagraph
3512
3513 }% an intentionally blank line
```

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

```
3514 \global\setbool{\LWR@doingstartpars}{true}%
```

No `<par>` tag yet to undo:

```
3515 \global\boolfalse{\LWR@doingapar}%
3516 }% nestspan
3517 }
```

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



```

3518 \newcommand*{\LWR@stoppars}%
3519 {%

```

Ignore if inside a span:

```

3520 \ifthenelse{\cnttest{\value{\LWR@spandepth}}{>}{0}}%
3521 {}%
3522 {%

```

See if currently handling HTML paragraphs:

```

3523 \ifbool{\LWR@doingapar}%

```

if currently in an HTML paragraph:

```

3524 {%

```

Print a closing tag:

```

3525 \unskip%
3526 \LWR@htmltagc{/LWR@tagregularparagraph}%
3527 \LWR@orignewline%

```

No longer have an open HTML paragraph:

```

3528 \global\boolfalse{\LWR@doingapar}%

```

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

```

3529 \global\boolfalse{\LWR@minipagethispar}
3530
3531 }% an intentionally blank line

```

If was not in an HTML paragraph:

```

3532 {}%

```

See if currently allowing HTML paragraphs:

```

3533 \ifbool{\LWR@doingstartpars}%

```

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

```

3534 {\ClearPreHook{par}}%

```

Else: do nothing

```

3535 {}%

```

no longer in paragraph mode

```
3536 \global\setbool{LWR@doingstartpars}{false}%
```

no <p> tag to undo:

```
3537 \global\boolfalse{LWR@doingapar}%
```

```
3538 }% nestspan
```

```
3539 }
```

```
3540 \end{warpHTML}
```

## 40 Page headers and footers

**for HTML & PRINT:** 3541 \begin{warpall}

In the following, catcode is manually changes back and forth without groups, since new macros are being defined which must not be contained within the groups.

```
3542 \newcommand{\LWR@firstpagetop}{} % for the home page alone
```

```
3543 \newcommand{\LWR@pagetop}{} % for all other pages
```

```
3544 \newcommand{\LWR@pagebottom}{}%
```

```
3545
```

```
3546 \newcommand{\LWR@setfirstpagetopb}[1]{%
```

```
3547 \renewcommand{\LWR@firstpagetop}{#1}
```

```
3548 \catcode'\_ =8
```

```
3549 }
```

\HTMLFirstPageTop {*{text and logos}*}

```
3550 \newcommand{\HTMLFirstPageTop}{%
```

```
3551 \catcode'\_ =12
```

```
3552 \LWR@setfirstpagetopb
```

```
3553 }
```

```
3554 \newcommand{\LWR@setpagetopb}[1]{%
```

```
3555 \renewcommand{\LWR@pagetop}{#1}
```

```
3556 \catcode'\_ =8
```

```
3557 }
```

\HTMLPageTop {*{text and logos}*}

```
3558 \newcommand{\HTMLPageTop}{%
```

```
3559 \catcode'\_ =12
```

```

3560 \LWR@setpagetopb
3561 }

3562 \newcommand{\LWR@setpagebottomb}[1]{%
3563 \renewcommand{\LWR@pagebottom}{#1}
3564 \catcode'\_ =8
3565 }

```

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

```

3566 \newcommand{\HTMLPageBottom}{%
3567 \catcode'\_ =12
3568 \LWR@setpagebottomb
3569 }

3570 \end{warpall}

```

## 41 CSS

**for HTML output:** 3571 `\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.

```

3572 \newcommand*{\LWR@currentcss}{lwarp.css}

```

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

```

3573 \newcommand*{\LWR@newcssb}[1]{%
3574 \renewcommand*{\LWR@currentcss}{#1}
3575 \catcode'\_ =8
3576 }
3577
3578 \newcommand*{\CSSFilename}{
3579 \catcode'\_ =12
3580 \LWR@newcssb
3581 }
3582 \end{warpHTML}

```

**for PRINT output:** 3583 `\begin{warpprint}`  
3584 `\newcommand*{\CSSFilename}[1]{}`  
3585 `\end{warpprint}`

## 42 HTML meta description and author

for HTML & PRINT: 3586 \begin{warpall}

\HTMLAuthor {*\authorname*}      The author to place into an HTML meta tag.

```
3587 \providecommand{\theauthor}{}
3588 \newcommand{\theHTMLAuthor}{\theauthor}
3589
3590 \newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}

3591 \end{warpall}
```

for HTML & PRINT: 3592 \begin{warpall}

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



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

\HTMLDescription {*\New HTML meta description.*}      Assigns the HTML file's description meta tag.

```
3593 \newcommand{\LWR@currentHTMLDescription}{}
3594
3595 \newcommand{\HTMLDescription}[1]{%
3596 \renewcommand{\LWR@currentHTMLDescription}{#1}
3597 }
3598
3599 \end{warpall}
```

## 43 Footnotes

lwarp uses native L<sup>A</sup>T<sub>E</sub>X footnote code, although with its own \box to avoid the L<sup>A</sup>T<sub>E</sub>X output routine. The usual functions work as-is.

Several kinds of footnotes are used: in a regular page, in a minipage, or as thanks in the titlepage. Each of these is handle differently.

### 43.1 Regular page footnotes

In HTML documents, footnotes are placed at the bottom of the web page using the L<sup>A</sup>T<sub>E</sub>X 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 43.4 for the implementation.

### 43.2 Minipage footnotes

See section 43.5 for how minipage footnotes are gathered. See section 69.3 for how minipage footnotes are placed into the document.

### 43.3 Titlepage thanks

See section 50.6 for titlepage footnotes.

### 43.4 Regular page footnote implementation

for HTML output: 3600 `\begin{warpHTML}`

Patch L<sup>A</sup>T<sub>E</sub>X footnotes to use a new `\box` for lwarp footnotes.

3601 `\newbox\LWR@footnotes`

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

`\@makefntext`  $\{\langle text \rangle\}$

3602 `\long\def\@makefntext#1{\textsuperscript{\@thefnmark} #1}`

`\@makefnmark`

3603 `\def\@makefnmark{\hbox{\textsuperscript{\@thefnmark}}}`

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

In a `lateximage` during HTML output, the `lateximage` is placed inside a print-mode `minipage`, but the footnotes are broken out by:

```

\def\@mpfn{footnote}
\def\thempfn{\thefootnote}
\let\@footnotetext\LWR@footnotetext

```

`\LWR@footnotetext` `{\text}`

```

3604 \long\def\LWR@footnotetext#1{%
3605 \global\setbox\LWR@footnotes=\vbox{%

```

Add to any current footnotes:

```

3606 \unvbox\LWR@footnotes%

```

Remember the footnote number for `\ref`:

```

3607 \protected@edef\@currentlabel{%
3608 \csname p@footnote\endcsname\@thefnmark%
3609 }% \@currentlabel

```

Open a group:

```

3610 \color@begingroup%

```

Use HTML superscripts even inside a `lateximage`:

```

3611 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%

```

Use paragraph tags if in a tabular data cell or a `lateximage`:

```

3612 \ifthenelse{%
3613 \boolean{LWR@doingstartpars} \AND%
3614 \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3615 }%
3616 {}%
3617 {\LWR@htmltagc{\LWR@tagregularparagraph}}%

```

Append the footnote to the list:

```

3618 \@makefntext{#1}%

```

Closing paragraph tag:

```

3619 \ifthenelse{%
3620 \boolean{LWR@doingstartpars} \AND%
3621 \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3622 }%
3623 {\par}%
3624 {}%

```

```

3625      \LWR@htmltagc{/LWR@tagregularparagraph}%
3626      \LWR@orignewline%
3627  }%

```

Close the group:

```

3628      \color@endgroup%
3629 }% vbox
3630 }%

```

```
\@footnotetext {<{text}>}
```

```
3631 \let\@footnotetext\LWR@footnotetext
```

### 43.5 Minipage footnote implementation

```
\@mpfootnotetext {<{text}>}
```

```

3632 \long\def\@mpfootnotetext#1{%
3633 \global\setbox\@mpfootins\vbox{%
3634   \unvbox\@mpfootins
3635   \reset@font\footnotesize
3636   \hsize\columnwidth
3637   \@parboxrestore
3638   \protected@edef\@currentlabel
3639   {\csname p@mpfootnote\endcsname\@thefnmark}%
3640   \color@begingroup
3641   \@makefntext{%
3642     \ignorespaces#1%
3643   }%

```

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

```

3644   \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
3645   {}%
3646   {%
3647     \LWR@htmltagc{/LWR@tagregularparagraph}%
3648     \LWR@orignewline%
3649   }%
3650   \color@endgroup%
3651 }% vbox
3652 }

```

`\LWR@printpendingfootnotes` Enclose the footnotes in a class, print, then clear.

```

3653 \newcommand*{\LWR@printpendingfootnotes}{%
3654 \ifvoid\LWR@footnotes\else
3655     \LWR@forcenewpage
3656     \begin{BlockClass}{footnotes}
3657     \LWR@origmedskip
3658     \unvbox\LWR@footnotes
3659     \setbox\LWR@footnotes=\vbox{}
3660     \end{BlockClass}
3661 \fi
3662 }

```

`\LWR@epubprintpendingfootnotes` Used to print footnotes before sections only if formatting for an EPUB or word processor:

```

3663 \newcommand*{\LWR@epubprintpendingfootnotes}{%
3664 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}{%
3665 {\LWR@printpendingfootnotes}%
3666 {}%
3667 }

3668 \end{warpHTML}

```

## 44 Marginpars

for HTML output: 3669 \begin{warpHTML}

`\marginpar` [*left*] [*right*]

```

3670 \renewcommand{\marginpar}[2] []{%
3671 \LWR@htmlspanclass{marginpar}{#2}%
3672 }

3673 \end{warpHTML}

```

## 45 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:** 3674 `\begin{warpall}`

Ctrl FileDepth `{\section depth}` determines how deeply to break into new HTML files, similar to `tocdepth`. The default of -5 produces one large HTML file.

```
3675 \newcounter{FileDepth}
3676 \setcounter{FileDepth}{-5}
```

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

```
3677 \newbool{CombineHigherDepths}
3678 \booltrue{CombineHigherDepths}

3679 \end{warpall}
```

**for HTML output:** 3680 `\begin{warpHTML}`

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

```
3681 \newcommand*{\LWR@thisfilename}{}
```

`\LWR@thisnewfilename` The filename being sanitized.

```
3682 \newcommand*{\LWR@thisnewfilename}{}
```

`\LWR@filenamenoblanks` `{\filename}`

Convert blanks into dashes, removes short words, store result in `\LWR@thisfilename`.



Be sure that this does not result in filename collisions! Use the optional TOC caption entry parameter for formatting. Remember to `\protect` L<sup>A</sup>T<sub>E</sub>X commands which appear in section names and TOC captions.

```
3683 \newcommand*{\LWR@filenamenoblanks}[1]{%
3684 \begingroup
```

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

```
3685 \LWR@nullfonts
3686 \renewcommand*{\LWR@htmltagc}[1]{%
```

Replaces common symbols and short words with hyphens:

```
3687 \edef\LWR@thisnewfilename{#1}
3688 \fullexpandarg
```

Convert spaces into hyphens:

```
3689 \StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctutation into hyphens:

```
3690 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
3691 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
3692 \StrSubstitute{\LWR@thisnewfilename}%
3693 {\LWR@origampersand}{-}[\LWR@thisnewfilename]
3694 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
3695 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
3696 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
3697 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
3698 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
3699 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
3700 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
3701 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
3702 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
3703 \StrSubstitute{\LWR@thisnewfilename}%
3704 {\textless}{-}[\LWR@thisnewfilename]
3705 \StrSubstitute{\LWR@thisnewfilename}%
3706 {\textgreater}{-}[\LWR@thisnewfilename]
3707 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
3708 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
3709 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
3710 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
3711 \StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
3712 \StrSubstitute{\LWR@thisnewfilename}%
3713 {\textbackslash}{-}[\LWR@thisnewfilename]
3714 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
3715 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
3716 %      "~{" for babel
3717 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
3718 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
3719 \StrSubstitute{\LWR@thisnewfilename}{' }{-}[\LWR@thisnewfilename]
```

Convert short words:

```
3720 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
3721 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
3722 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
3723 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
3724 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
3725 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
3726 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
3727 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
3728 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
3729 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
3730 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
```

```

3731 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
3732 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
3733 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
3734 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
3735 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
3736 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
3737 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

3738 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
3739 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
3740 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
3741 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
3742 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
3743 %      emdash
3744 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
3745 %      endash
3746 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
3747 \endgroup
3748 }

```

`\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.

```

3749 \newcommand*{\LWR@newhtmlfile}[1]{

```

At the bottom of the ending file:

```

3750 \LWR@htmllementclassend{section}{textbody}
3751
3752 \LWR@printpendingfootnotes
3753

```

No footer between files if EPUB:

```

3754 \ifbool{FormatEPUB}
3755 {}
3756 {
3757     \LWR@htmllement{footer}
3758
3759     \LWR@pagebottom
3760
3761     \LWR@htmllementend{footer}

```

3762 }

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

```
3763 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3764 {}
3765 {\ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}}
```

End of this HTML file:

```
3766 \LWR@stoppars
3767 \LWR@htmltag{/body}\LWR@orignewline
3768 \LWR@htmltag{/html}\LWR@orignewline
3769 \LWR@orignewpage
3770
3771 \addtocounter{LWR@htmlfilenumber}{1}%
```

If using a filename, create a version without blanks. The filename without blanks will be placed into `\LWR@thisfilename`. If not using a filename, the file number will be used instead.

```
3772 \ifbool{FileSectionNames}%
3773 {\LWR@filenameno blanks{#1}}
3774 {\renewcommand*{\LWR@thisfilename}{\theLWR@htmlfilenumber}}
```

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.

```
3775 \LWR@htmlblockcomment{%
3776 |Start file|
3777 \LWR@htmlsectionfilename{\LWR@thisfilename}|
3778 }
```

At the top of the starting file:

```
3779 \LWR@stoppars
3780
3781 \LWR@filestart{ -- #1}% there is an EMDash in front of the #1
3782
```

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

```
3783 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3784 {}
3785 {\LWR@topnavigation}
```

3786

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

```
3787 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3788 {}
3789 {
3790     \LWR@html element{header}
3791
3792     \LWR@pagetop
3793
3794     \LWR@html elementend{header}
3795 }
3796
```

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

```
3797 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3798 {}
3799 {\ifcvoid{thetitle}{}\LWR@printthetitle}
3800
```

No sideTOC if formatting for an EPUB or word processor:

```
3801 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3802 {}
3803 {\LWR@sidetoc}
3804
```

Start of the <textbody>:

```
3805 \LWR@html elementclass{section}{textbody}
3806
```

Keep paragraph tags disabled for now:

```
3807 \LWR@stoppars
3808
```

Track the page numbers:

```
3809 \setcounter{LWR@latestautopage}{\value{page}}
3810 }
```

```
3811 \end{warpHTML}
```

## 46 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing L<sup>A</sup>T<sub>E</sub>X 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<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X and LuaL<sup>A</sup>T<sub>E</sub>X directly support accented section and file names.

**for HTML output:** 3812 `\begin{warpHTML}`

### 46.1 Book class commands

`\mainmatter` Declare the main matter section of the document. Does not reset the page number, which must be consecutive arabic numbers for the HTML conversion.



```
3813 \newbool{LWR@mainmatter}
3814 \DeclareDocumentCommand{\mainmatter}{}{%
3815   \booltrue{LWR@mainmatter}%
3816 }
```

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

```
3817 \DeclareDocumentCommand{\frontmatter}{}{%
3818   \boolfalse{LWR@mainmatter}%
3819 }
```

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

```
3820 \DeclareDocumentCommand{\backmatter}{}{%
3821   \boolfalse{LWR@mainmatter}
3822 }
```

### 46.2 Sectioning support macros

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

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

```
3823 \newcommand*{\LWR@sectionnumber}[1]{%
3824 \InlineClass{sectionnumber}{#1}%
3825 }
```


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

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

Create an autosection tag.

```
3826 \newcommand*{\LWR@createautosec}[1]{%
3827 \LWR@htmltag{#1 id="autosec-\thepage"{}%
3828 }
```


**\LWR@pushoneclose**  $\{\langle depth\rangle\}$   $\{\langle printclose\rangle\}$  Stacks the new sectioning level's closing tag, to be used when this section is closed some time later.

 **\LWR@stoppars** must be executed first.

```
3829 \NewDocumentCommand{\LWR@pushoneclose}{m m}{\pushclose{#2}{#1}}
```

**\LWR@startnewdepth**  $\{\langle depth\rangle\}$   $\{\langle printclose\rangle\}$

Closes currently stacked tags of a lesser level, then opens the new nesting level by saving this new sectioning level's closing tag for later use.

 **\LWR@stoppars** must be executed first.

```
3830 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
3831 \LWR@closeprevious{#1}%
```

Push a new section depth:

```
3832 \LWR@pushoneclose{#1}{#2}%
3833 }
```

**Ctrl** **\LWR@prevFileDepth** Remembers the previous **\LWR@FileDepth**.

Initialized to a deep level so that any section will trigger a new HTML page after the home page.

```
3834 \newcounter{\LWR@prevFileDepth}
3835 \setcounter{\LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

`\LWR@section * [⟨TOC name⟩] {⟨name⟩} {⟨sectiontype⟩}`

The common actions for the high-level sectioning commands.

```
3836 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
3837 \LWR@traceinfo{\LWR@section}%
3838 \LWR@stoppars%
```

Cancel special minipage horizontal space interaction:

```
3839 \global\boolfalse{\LWR@minipagethispar}%
```

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

```
3840 \LWR@traceinfo{\LWR@section: testing whether to start a new HTML file}%
3841 \IfBooleanTF{#1}{% starred
```

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

```
3842     \LWR@orignewpage%
3843
3844 }{% not starred
3845     \ifthenelse{%
3846         \cnttest{\csuse{\LWR@depth#4}}{<=}{\value{FileDepth}}%
3847         \AND%
3848         \(%
3849         \NOT\boolean{CombineHigherDepths}\OR%
3850         \cnttest{\csuse{\LWR@depth#4}}{<=}{\value{\LWR@prevFileDepth}}%
3851         \)%
3852     }%
```

If so: start a new HTML file:

```
3853     {% new file
3854         \LWR@traceinfo{\LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
3855         \IfNoValueTF{#2}%
```

If no optional entry

```
3856         {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
3857         {\LWR@newhtmlfile{#2}}%
3858     }% new file
```



Else: No new HTML file:

```
3859    {% not new file
```

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

```
3860        \LWR@orignewpage%
3861
3862    }% not new file
3863 }% not starred
```

Remember this section's name for \nameref:

```
3864 \LWR@traceinfo{\LWR@section: about to \LWR@setlatestname}%
3865 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
```

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

```
3866
3867 \ifbool{HTMLDebugComments}{%
3868     \LWR@htmlcomment{Opening #4 ‘#3’}}}%
3869 }{}%
3870
```

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

```
3871 \ifthenelse{%
3872     \cnttest{\csuse{\LWR@depth#4}}{>=}{\LWR@depthparagraph}%
3873 }%
3874 {\LWR@startpars}%
3875 }%
```

Create the opening tag with an autosec:

```
3876 \LWR@createautosec{\csuse{\LWR@tag#4}}%
```

If not starred, step counter and add to TOC:

```
3877 \IfBooleanTF{#1}%
3878 {}% starred
3879 {% not starred
```

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

```
3880     \ifthenelse{%
3881         \cnttest{\csuse{\LWR@depth#4}}{<=}{\value{secnumdepth}}%
3882     }%
3883     {% if secnumdepth
```

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

```

3884      \LWR@traceinfo{LWR@section: about to test main matter}%
3885      \ifbool{LWR@mainmatter}%
3886      {%
3887          \LWR@traceinfo{LWR@section: yes mainmatter}%
3888          \refstepcounter{#4}%

```

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

```

3889      \LWR@traceinfo{LWR@section: about to addcontentsline}%
3890      \addcontentsline{toc}{#4}%
3891      {\protect\LWR@sectionnumber{\csuse{the#4}}}%
3892      \IfValueTF{#2}{ #2}{ #3}}%
3893      \LWR@traceinfo{LWR@section: finished addcontentsline}%
3894      }% end of if main matter

```

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

```

3895      {% not main matter
3896          \LWR@traceinfo{LWR@section: no main matter}%
3897          \addcontentsline{toc}{#4}{\IfValueTF{#2}{ #2}{ #3}}%
3898      }% end of not main matter
3899  }% end of secnumdepth

```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```

3900      {%
3901          \addcontentsline{toc}{#4}{\IfValueTF{#2}{ #2}{ #3}}%
3902      }%

```

For part, print the section type:

```

3903      \ifbool{LWR@mainmatter}%
3904      {%
3905          \ifthenelse{%
3906              \(\cnttest{\csuse{LWR@depth#4}}{<=}%
3907              {\value{secnumdepth}}\)\ AND%
3908              \(\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}}\)%
3909          }%
3910          {\csuse{#4name}~{}}%
3911          {}%

```

Print the section number:

```

3912      \LWR@traceinfo{LWR@section: about to print section number}%
3913      \ifthenelse{%
3914          \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%

```

```

3915      }%
3916      {\protect\LWR@sectionnumber{\csuse{the#4}}~}%
3917      }%
3918      \LWR@traceinfo{LWR@section: finished print section number}%
3919      }{}%
3920 }% end of not starred

```

Print the section name:

```

3921 #3%

```

Close the heading tag, such as /H2:

```

3922 \LWR@htmltag{\csuse{LWR@tag#4end}}%

```

Generate a L<sup>A</sup>T<sub>E</sub>X label:

```

3923 \label{autopage-\thepage}%

```

Start paragraph handing unless is an inline paragraph or subparagraph:

```

3924 \ifthenelse{%
3925     \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}%
3926 }%
3927 {\LWR@startpars}%
3928 {}%

```

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

A starred section does not trigger a new HTML page at the beginning of this macro, so it should not affect it here at the end either. This became an issue when a `\listoftables` was tested in the middle of the document. The `\chapter*` for the list was not allowing a new HTML page for the section following it while `CombineHigherDepths` was true.

```

3929 \IfBooleanTF{#1}%
3930 {}%
3931 {% not starred
3932     \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
3933 }% not starred
3934 \LWR@traceinfo{LWR@section: done}%
3935 }

```

### 46.3 \section and friends

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

```

3936 \@ifundefined{chapter}
3937 {}
3938 {%
3939 \DeclareDocumentCommand{\part}{s o m}{%
3940 \LWR@epubprintpendingfootnotes%
3941 \LWR@stoppars%
3942
3943 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
3944
3945 \LWR@section{#1}{#2}{#3}{part}%
3946 }
3947 }

```

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

```

3948 \@ifundefined{chapter}
3949 {}
3950 {%
3951 \DeclareDocumentCommand{\chapter}{s o m}{%
3952 \LWR@traceinfo{chapter #3}%
3953 \LWR@epubprintpendingfootnotes%
3954 \LWR@stoppars%
3955
3956 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
3957
3958 \LWR@traceinfo{chapter: about to LWR@section}%
3959 \LWR@section{#1}{#2}{#3}{chapter}%
3960 \LWR@traceinfo{chapter: done}%
3961 }
3962 }

```

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

```

3963 \DeclareDocumentCommand{\section}{s o m}{%
3964 \LWR@epubprintpendingfootnotes%
3965 \LWR@stoppars%
3966
3967 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
3968
3969 \LWR@section{#1}{#2}{#3}{section}%
3970 }

```

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

```

3971 \DeclareDocumentCommand{\subsection}{s o m}{%
3972 \LWR@epubprintpendingfootnotes%
3973 \LWR@stoppars%
3974

```

```

3975 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
3976
3977 \LWR@section{#1}{#2}{#3}{subsection}%
3978 }

```

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

```

3979 \DeclareDocumentCommand{\subsubsection}{s o m}{%
3980 \LWR@epubprintpendingfootnotes%
3981 \LWR@stoppars%
3982
3983 \LWR@startnewdepth{\LWR@depthsubsubsection}%
3984 {\LWR@printclosesubsubsection}%
3985
3986 \LWR@section{#1}{#2}{#3}{subsubsection}%
3987 }

```

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

```

3988 \DeclareDocumentCommand{\paragraph}{s o m}{%
3989 \LWR@epubprintpendingfootnotes%
3990 \LWR@stoppars%
3991
3992 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
3993
3994 \LWR@section{#1}{#2}{#3}{paragraph}%
3995 }

```

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

```

3996 \DeclareDocumentCommand{\subparagraph}{s o m}{%
3997 \LWR@epubprintpendingfootnotes%
3998 \LWR@stoppars%
3999
4000 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
4001
4002 \LWR@section{#1}{#2}{#3}{subparagraph}%
4003 }

4004 \end{warpHTML}

```

## 47 Starting a new file

for HTML & PRINT: 4005 \begin{warpall}

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

```
4006 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
4007
4008 \newcommand*{\HTMLLanguage}[1]{%
4009 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
4010 }

4011 \end{warpall}
```

for HTML output: 4012 `\begin{warpHTML}`

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

Creates the opening HTML tags.

```
4013 \newcommand*{\LWR@filestart}[1]{
4014 \LWR@traceinfo{\LWR@filestart}
```

Locally temporarily disable direct-formatting commands:

```
4015 \begingroup
4016 \LWR@traceinfo{\LWR@filestart: A}
4017 \LWR@nullfonts
4018 \LWR@traceinfo{\LWR@filestart: B}
```

Create the page's HTML header:

```
4019 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
4020 \LWR@traceinfo{\LWR@filestart: C}
```

The language is user-adjustable:

```
4021 \LWR@htmltag{html lang="\LWR@currentHTMLLanguage"}\LWR@orignewline
```

Start of the meta data:

```
4022 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
4023 \LWR@htmltag{meta charset="UTF-8" /\LWR@orignewline
```

Author:

```
4024 \ifcempty{theHTMLAuthor}{}{
4025 \LWR@htmltag{meta name="author" content="\theHTMLAuthor" /\LWR@orignewline
4026 }
```

lwarp is the generator:

```
4027 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /}%
4028   \LWR@orignewline
```

If there is a description, add it now:

```
4029 \ifdefempty{\LWR@currentHTMLDescription}{}%
4030 \LWR@htmltag{%
4031 meta name="description" content="\LWR@currentHTMLDescription" /}%
4032   \LWR@orignewline
4033 }%
```

Mobile-friendly viewport:

```
4034 \LWR@htmltag{meta name="viewport" %
4035 content="width=device-width, initial-scale=1.0" /}%
4036   \LWR@orignewline
```

IE patch:

```
4037 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
4038 \LWR@htmltag{%
4039 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{}%
4040 \LWR@htmltag{/script}\LWR@orignewline
4041 \LWR@htmltag{![endif]{-}{-}}\LWR@orignewline
```

The page's title:

```
4042 \ifcsvoid{thetitle}{}%
4043 \LWR@htmltag{title}\thetitle#1\LWR@htmltag{/title}\LWR@orignewline%
4044 }%
```

The page's stylesheet:

```
4045 \LWR@htmltag{%
4046 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
4047 \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.

```
4048 \ifbool{mathjax}%
4049 {%
4050   \boolfalse{LWR@verbtags}
4051   \verbatiminput{lwarp_mathjax.txt}%
4052   \booltrue{LWR@verbtags}
4053   \LWR@stoppars
```

```
4054 }% end of mathjax
4055 {}%
```

End of the header:

```
4056 \LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
4057 \LWR@htmltag{body}\LWR@orignewline
4058 \endgroup
4059 \LWR@traceinfo{LWR@filestart: done}
4060 }

4061 \end{warpHTML}
```

## 48 Starting HTML output

**for HTML output:** 4062 \begin{warpHTML}

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

```
4063 \catcode'\$=\active
4064 \newcommand*{\LWR@LwarpStart}
4065 {%
4066 \LWR@traceinfo{LWR@lwarpStart}
```

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

```
4067 \ifbool{FormatWordProcessor}{%
4068   \setcounter{FileDepth}{-5}%
4069   \boolfalse{HTMLDebugComments}%
4070 }{}
```

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

```
4071 \edef\LWR@strresult{\HomeHTMLFilename}
4072 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
4073 \edef\LWR@strresult{\HTMLFilename}
4074 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn and empty page style:

```
4075 \LWR@origonecolumn%
4076 \LWR@origpagestyle{empty}
```



Reduce chance of line overflow in verbatim environments:

```
4077 \LWR@origscriptsize%
```

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

```
4078 \LWR@origraggedright%
4079 \LetLtxMacro{\}\{\LWR@endofline}%
```

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

```
4080 \linespread{1.3}%
```

For `pdftotext` to reliably identify paragraph splits:

```
4081 \setlength{\parindent}{0pt}
4082 \setlength{\parskip}{2ex}
```

For the `lateximages` record file:

```
4083 \immediate\openout\LWR@file=lateximages.txt
```

Removes space after the caption in the HTML:

```
4084 \setlength{\belowcaptionskip}{-3ex}
```

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

```
4085 \renewcommand{\ps@plain}{}%
```

```
\centering Not used in the HTML environment:
\raggedleft
\raggedright 4086 \renewcommand*{\centering}{}
4087 \renewcommand*{\raggedleft}{}
4088 \renewcommand*{\raggedright}{}%
```

Plug in some new actions. This is done just before the document start so that they won't be over-written by some other package.

Tabular:

```
4089 \LetLtxMacro{\LWR@origtabular}\tabular}
4090 \LetLtxMacro{\LWR@origendtabular}\endtabular}
4091 \LetLtxMacro{\tabular}\LWR@tabular}
4092 \LetLtxMacro{\endtabular}\endLWR@tabular}
```

Float captions:

```
4093 \let\LWR@origcaption\caption
```

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

```
4094 \let\LWR@origltx@label\ltx@label
4095 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
4096 \let\TeX\LWR@TeX
4097 \let\LaTeX\LWR@LaTeX
4098 \let\LuaTeX\LWR@LuaTeX
4099 \let\LuaLaTeX\LWR@LuaLaTeX
4100 \let\XeTeX\LWR@XeTeX
4101 \let\XeLaTeX\LWR@XeLaTeX
4102 \let\ConTeXt\LWR@ConTeXt
```

Graphics:

```
4103 \LetLtxMacro{\rotatebox}{\LWR@rotatebox}
4104 \LetLtxMacro{\scalebox}{\LWR@scalebox}
4105 \let\reflectbox\LWR@reflectbox
```

Not yet started any paragraph handling:

```
4106 \global\boolfalse{LWR@doingapar}
4107 \global\boolfalse{LWR@doingstartpars}
```

Start a new HTML file and a header:

```
4108 \LWR@traceinfo{LWR@lwarpStart: Starting new file.}
4109 \LWR@filestart{}
4110 \LWR@traceinfo{LWR@lwarpStart: Generating first header.}
4111 \LWR@htmltag{header}\LWR@orignewline
4112 \LWR@startpars
4113 \LWR@firstpagetop
4114 \LWR@stoppars
4115 \LWR@htmltag{/header}\LWR@orignewline
4116 \LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
4117 \LWR@htmltag{section class="textbody"{{}}
```

Document and page settings:

```
4118 \mainmatter
4119 \LWR@origpagenumbering{arabic}
```

Set default titlepage thanks footnote marks. See section 50.6.

```
4120 \if@titlepage
4121   \thanksmarkseries{arabic}
```

```

4122 \else
4123   \thanksmarkseries{fnsymbol}
4124 \fi

```

Patch the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native L<sup>A</sup>T<sub>E</sub>X environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```

4125 \LWR@patchlists

```

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

```

4126 \catcode'\$=\active

```

Allow HTML paragraphs to begin:

```

4127 \LWR@startpars
4128 \LWR@traceinfo{\LWR@lwarpStart: done}
4129 }
4130 \catcode'\$=3% math shift until lwarp starts
4131 \end{warpHTML}

```

## 49 Ending HTML output

**for HTML output:** 4132 `\begin{warpHTML}`

`\LWR@requesttoc`  $\{\langle boolean \rangle\} \{\langle suffix \rangle\}$  Requests that a toc, lof, or lot be generated.

```

4133 \newcommand*{\LWR@requesttoc}[2]{%
4134 \ifbool{#1}
4135 {
4136   \expandafter\newwrite\csuse{tf@#2}
4137   \immediate\openout \csuse{tf@#2} \jobname.#2\relax
4138 }{}
4139 }

```

`\LWR@LwarpEnd` Final stop of all HTML output:

```

4140 \newcommand*{\LWR@LwarpEnd}
4141 {
4142 \LWR@stoppars
4143 \LWR@closeprevious{\LWR@depthfinished}

```

At the bottom of the ending file:

Close the textbody:

```
4144 \LWR@html\element\classend{section}{textbody}
```

Print any pending footnotes:

```
4145 \LWR@printpendingfootnotes
```

Create the footer:

```
4146 \LWR@html\element{footer}
4147
4148 \LWR@pagebottom
4149
4150 \LWR@html\elementend{footer}
```

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

Presumably has a table-of-contents.

```
4151 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
4152 {}
4153 {
4154     \ifnumcomp{\value{LWR@html\filenumber}}{>}{0}{\LWR@botnavigation}{}
4155 }
```

```
4156 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
4157 \LWR@htmltag{/body}\LWR@orignewline
4158 \LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
4159 \LWR@orignewpage
```

For lateximage commands:

```
4160 \immediate\closeout\LWR@file
4161 }
```

```
4162 \end{warpHTML}
```

## 50 Titles and the titling package

Supports and extends the titling package.

Additional functions include `\published` and `\subtitle`, and the `\author` command has an additional `\affiliation` command to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using `\theauthor` in the main text.

The `titling` package maintains the definitions of `\thetitle`, `\theauthor`, etc., after the title has been typeset. These commands are to be used to refer to the document's title and author, etc., in the main text. These definitions have the `\thanks` and `\affiliation` removed, and for author the `\and` is replaced to generate a simple inline list of authors separated by commas.

`\printtitle`, `\printauthor`, etc., are provided for use inside the `titlepage` or `titlingpage` environments, and these retain the `\thanks` and `\affiliation`.

Several additional hooks are provided in addition to titling:

`\maketitlehookaa` `\maketitlehookaa`: Between “published” and the title.

`\maketitlehookaaa` `\maketitlehookaaa`: Between the title and the subtitle.


`\prepublished` `\prepublished`: Before the “published” field.

`\postpublished` `\postpublished`: After the “published” field.

`\presubtitle` `\presubtitle`: Before the subtitle.

`\postsubtitle` `\postsubtitle`: After the subtitle.

`\printthanks` `\printthanks` has been added to force the printing of thanks inside a `titlingpage` environment when `\maketitle` is not used.

 **No footnotes!** Inside a `\titlepage` or `\titlingpage` environment, use `\thanks` for footnotes, do not use `\footnote`.

At the end of the `titlingpage` environment, footnote marks are forced to reset to zero.

Inside a `titlingpage` environment with the `article` document class, thanks marks will be `fnsymbol` instead of `arabic`. `arabic` is usually used when inside `titlepage` environments where the title page is on its own page, but is not automatically used inside a `titlingpage` environment.

To force the thanks marks to be `arabic`:

---

```
\begin{titlingpage}
\thanksmarkseries{arabic}
...
```

---

## 50.1 Setting the title, etc.

The following provide setting commands for both HTML and print outputs.

<pre>\@published   \@title   \@subtitle   \@author   \@date</pre>	<pre>\@title, \@subtitle, \@author, etc. store the values as originally assigned, including any \thanks, \and, or \affiliation. These are low-level macros intended to be used by other macros only inside a titlepage or titlingpage, and are used by \maketitle. The author is printed inside a single-column table, which becomes multiple single-column tables if multiples authors are included.</pre>
<pre>\printpublished   \printtitle   \printsubtitle   \printauthor   \printdate</pre>	<pre>\printtitle, \printsubtitle, etc. are user-level macros intended to be used in titlepage and titlingpage environments in cases where \maketitle is not desired. These commands preserve the \thanks, etc., and should not be used in the main text. The author is printed inside a single-column table, which becomes multiple single-column tables if multiples authors are included.</pre>
<pre>\thepublished   \thetitle   \thesubtitle   \theauthor   \thedate</pre>	<pre>\thetitle, \thesubtitle, \theauthor, etc. are user-level sanitized versions which have 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:</pre>
<pre>\HTMLPageBottom</pre>	<pre>{\text}</pre>

---

```
\HTMLPageBottom{
\begin{center}\textcopyright~2016 \theauthor\end{center}
}
```

---

<pre>\author   \and</pre>	<pre>{\author}} While using \maketitle, the author is treated as a single-column table and the \and feature finishes the current table then starts a new one for the next author. Each author thus is placed into its own table, and an affiliation may be placed on its own line such as</pre>
---------------------------	---

```
\author{Name \\ Affiliation \and Second Name \\ Second Affiliation}
```

After `\maketitle` has completed, `\theauthor` retains the definition of the author, but `\and` is changed to become a comma and a space, intending to print the authors

names separated by spaces. This fails when affiliations are included on their own table rows.

A solution, provide here, is to define a macro `\affiliation` which during `\maketitle` starts a new table row and adds the affiliation, but after `\maketitle` is finished `\affiliation` is re-defined to throw away its argument, thus printing only the author names when `\author` is later used inline.

```
\affiliation {<text>}
```

Adds the affiliation to the author for use in `\maketitle`. Nullified when later used for inline use of `\theauthor`.

```
for HTML output: 4163 \begin{warpHTML}
4164 \newrobustcmd{\affiliation}[1]{\ \ \ \InlineClass{affiliation}{#1}}
4165 \end{warpHTML}
```

```
for PRINT output: 4166 \begin{warpprint}
4167 \newrobustcmd{\affiliation}[1]{\ \ \ \textsc{\small#1}}
4168 \end{warpprint}
```

The following are based on the original titling code:

```
for HTML & PRINT: 4169 \begin{warpall}
```

```
\author {<text>}
```

Redefined to nullify `\affiliation`, etc. before printing the authors inline.

`\@author` retains the entire author with its `\thanks`, while `\theauthor` will have `\thanks` removed and `\and` simplified.

```
4170 \renewcommand{\author}[1]{%
4171 \gdef\@author{#1}
4172 \begingroup
4173 \renewcommand{\thanks}[1]{ }
4174 \renewcommand{\and}{\unskip, }
4175 \renewcommand{\thanksmark}[1]{ }
4176 \renewcommand{\thanksgap}[1]{ }
4177 \renewcommand{\affiliation}[1]{ }
4178 \protected@xdef\theauthor{#1}
4179 \endgroup}
```

```
\published {<text>}
```

```
4180 \newcommand{\published}[1]{%
4181 \gdef\@published{#1}
```

```

4182 \begingroup
4183   \renewcommand{\thanks}[1]{%
4184     \renewcommand{\thanksmark}[1]{%
4185       \renewcommand{\thanksgap}[1]{%
4186         \protected@xdef\thepublished{#1}
4187       }
4188     }
4189   \newcommand{\@published}{%
4190     \newcommand{\thepublished}{%

\subtitle  {\textit{}}

4191 \newcommand{\subtitle}[1]{%
4192 \gdef\@subtitle{#1}
4193 \begingroup
4194   \renewcommand{\thanks}[1]{%
4195     \renewcommand{\thanksmark}[1]{%
4196       \renewcommand{\thanksgap}[1]{%
4197         \protected@xdef\thesubtitle{#1}
4198       }
4199     }
4200   \newcommand{\@subtitle}{%
4201     \newcommand{\thesubtitle}{%

4202 \end{warpall}

```

## 50.2 Changes to HTML titlepage and titlingpage

for HTML output: 4203 \begin{warpHTML}

Env **titlepage** Sets up a titlepage div with a L<sup>A</sup>T<sub>E</sub>X PDF minipage inside.

```

4204 \renewenvironment*{titlepage}
4205 {
4206   \LWR@forcenewpage
4207   \BlockClass{titlepage}\LWR@subminipage
4208 }
4209 {\LWR@endsubminipage\endBlockClass}

```

Env **titlingpage**

```

4210 \renewenvironment*{titlingpage}
4211 {%

```

Start an HTML titlepage div:



```
4212 \begin{titlepage}
```

Prepare for a custom version of `\maketitle` inside the `titlingpage`:

```
4213 \LWR@maketitlesetup
4214 \let\maketitle\LWR@titlingmaketitle
4215 }
4216 {
```

At the end of the environment, end the HTML titlepage div:

```
4217 \end{titlepage}
```

Reset the footnote counter:

```
4218 \@bscontmark
4219 }
```

```
4220 \end{warpHTML}
```

**for HTML & PRINT:** 4221 `\begin{warpall}`

`\printthanks` Forces the `\thanks` to be printed.

This is necessary in a `titlingpage` environment when `\maketitle` was not used.

```
4222 \newcommand*{\printthanks}{\@thanks}
```

Env `titlingpage` At the end of the `titlingpage` for both print and HTML, reset footnote markers to zero.

```
4223 \AtEndEnvironment{titlingpage}{\@bscontmark}
```

```
4224 \end{warpall}
```

### 50.3 Printing the title, etc. in HTML

The following are for printing the title, etc. in a `titlepage` or a `titlingpage` in HTML:

**for HTML output:** 4225 `\begin{warpHTML}`

Patch the pre/post title/author/date to add HTML tags, then initialize:

```
4226 \newcommand{\prepublished}[1]{%
```

```
4227 \def\@bsprepublished{\BlockClass{published}#1}%
4228 }
4229
4230 \newcommand{\postpublished}[1]{%
4231 \def\@bspostpublished{#1\endBlockClass}%
4232 }
4233
4234 \renewcommand{\pretitle}[1]{%
4235 \def\@bspretitle{#1\LWR@stoppars\LWR@htmltag{h1}}}%
4236 }
4237
4238 \renewcommand{\posttitle}[1]{%
4239 \def\@bsposttitle{\LWR@htmltag{/h1}\LWR@startpars#1}%
4240 }
4241
4242 \newcommand{\presubtitle}[1]{%
4243 \def\@bspresubtitle{\BlockClass{subtitle}#1}%
4244 }
4245
4246 \newcommand{\postsubtitle}[1]{%
4247 \def\@bspostsubtitle{#1\endBlockClass}%
4248 }
4249
4250 \renewcommand{\preauthor}[1]{%
4251 \def\@bspreauthor{\BlockClass{author}#1}%
4252 }
4253
4254 \renewcommand{\postauthor}[1]{%
4255 \def\@bspostauthor{#1\endBlockClass}%
4256 }
4257
4258 \renewcommand{\predate}[1]{%
4259 \def\@bspredate{#1\BlockClass{titledate}}}%
4260 }
4261
4262 \renewcommand{\postdate}[1]{%
4263 \def\@bspostdate{\endBlockClass#1}%
4264 }
4265
4266 \prepublished{\begin{center}}
4267 \postpublished{\par\end{center}}
4268
4269 \pretitle{\begin{center}}
4270 \posttitle{\par\end{center}}
4271
4272 \presubtitle{\begin{center}}
4273 \postsubtitle{\par\end{center}}
4274
4275 \preauthor{\begin{center}}%
4276 \begin{tabular}[t]{c}%
```

```

4277 }
4278 \postauthor{\end{tabular}\par\end{center}}
4279
4280 \predate{\begin{center}}
4281 \postdate{\par\end{center}}

```

### \printpublished

```

4282 \newcommand*{\printpublished}{
4283 \ifthenelse{\equal{\thepublished}{}}{
4284 {}
4285 {
4286     \begin{BlockClass}{published}
4287     \@published
4288     \end{BlockClass}
4289 }
4290 }

```

### \printtitle

```

4291 \newcommand*{\printtitle}
4292 {
4293 \LWR@stoppars
4294 \LWR@htmltag{h1}%
4295 \@title%
4296 \LWR@htmltag{/h1}
4297 \LWR@startpars
4298 }

```

**\LWR@printthetitle** A private version which prints the title without footnotes, used to title each HTML page.

```

4299 \newcommand*{\LWR@printthetitle}
4300 {
4301 \LWR@stoppars
4302 \LWR@htmltag{h1}%
4303 \thetitle%
4304 \LWR@htmltag{/h1}
4305 \LWR@startpars
4306 }

```

### \printsubtitle

```

4307 \newcommand*{\printsubtitle}{
4308 \ifthenelse{\equal{\thesubtitle}{}}{
4309 {}
4310 {

```

```

4311 \begin{BlockClass}{subtitle}
4312 \@subtitle
4313 \end{BlockClass}
4314 }
4315 }

```

`\printauthor`

```

4316 \newcommand*{\printauthor}{
4317 \begin{BlockClass}{author}
4318 \begin{tabular}{c}\@author\end{tabular}
4319 \end{BlockClass}
4320 }

```

`\printdate`

```

4321 \newcommand*{\printdate}{%
4322 \begin{BlockClass}{titledate}
4323 \@date
4324 \end{BlockClass}
4325 }

```

```

4326 \end{warpHTML}

```

## 50.4 Printing the title, etc. in print form

The following are for printing the title, etc. in a `titlepage` or a `titlingpage` in print form:

**for PRINT output:** 4327 `\begin{warpprint}`

`\printpublished`

```

4328 \newcommand*{\printpublished}{\Large\scshape\@published}

```

`\printtitle`

```

4329 \newcommand*{\printtitle}{\Huge\@title}

```

`\printsubtitle`

```

4330 \newcommand*{\printsubtitle}{\Large\itshape\@subtitle\bigskip}

```

`\printauthor`

```
4331 \newcommand*{\printauthor}
4332   {{\large\begin{tabular}[t]{c}\@author\end{tabular}}}
```

`\printdate`

```
4333 \newcommand*{\printdate}{{\small\textit{\@date}}}
```

## 50.5 `\maketitle` for print output

`\maketitle` From the titling package, patched to add the publisher and subtitle.

```
4334 \providecommand{\maketitle}{}
4335 \if@titlepage
4336   \renewcommand{\maketitle}{\begin{titlepage}%
4337     \let\footnotesize\small
4338     \let\footnoterule\relax
4339     \LetLtxMacro{\footnote}{\thanks}
4340     \bsmarkseries
4341     \def\@makefnmark{\rlap{\@textsuperscript{%
4342       \normalfont\bsthanksheadpre \tamark \bsthanksheadpost}}}%
4343     \long\def\@makefntext##1{\makethanksmark ##1}
4344     \null\vfil
4345     \vskip 60\p@
4346     \vspace*{\droptitle}
4347     \maketitlehooka
4348     \ifcsempy{@published}
4349       {}
4350       {\@bsprepublished \@published \bspostpublished}\maketitlehookaa}
4351     {\@bspretitle \@title \bsposttitle}
4352     \ifcsempy{@subtitle}
4353       {}
4354       {\maketitlehookaaa{\@bspresubtitle \@subtitle \bspostsubtitle}}
4355     \maketitlehookb
4356     {\@bspreauthor \@author \bspostauthor}
4357     \maketitlehookc
4358     {\@bspredate \@date \bspostdate}
4359     \maketitlehookd
4360     \par
4361     \thanks
4362     \vfil\null
4363     \end{titlepage}%
4364     \bscontmark % \setcounter{footnote}{0}%
4365     %% \bsmtitleempty
4366   } % end titlepage defs
```

```

4367 \else
4368   \renewcommand{\maketitle}{\par
4369     \begin{group}
4370       \@bsmarkseries
4371       \def\@makefnmark{\rlap{\@textsuperscript{%
4372         \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
4373       \long\def\@makefntext##1{\makethanksmark ##1}
4374       \if@twocolumn
4375         \ifnum \col@number=\@ne
4376           \@maketitle
4377         \else
4378           \twocolumn[\@maketitle]%
4379         \fi
4380       \else
4381         \newpage
4382         \global\@topnum\z@
4383         \@maketitle
4384       \fi
4385       \thispagestyle{plain}\@thanks
4386     \endgroup
4387     \@bscontmark % \setcounter{footnote}{0}%
4388     %% \@bsmtitleempty
4389   } % end non-titlepage
4390
4391   \def\@maketitle{%
4392     \newpage
4393     \null
4394     \vskip 2em%
4395     \vspace*{\droptitle}
4396     \maketitlehooka
4397     \ifcsempy{\@published}
4398       {}
4399       {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
4400     {\@bspretitled \@title \@bsposttitled}
4401     \ifcsempy{\@subtitle}
4402       {}
4403       {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
4404     \maketitlehookb
4405     {\@bspreauthor \@author \@bspostauthor}
4406     \maketitlehookc
4407     {\@bspredate \@date \@bspostdate}
4408     \maketitlehookd
4409     \par
4410     \vskip 1.5em}
4411 \fi
4412
4413 \providecommand{\maketitlehookaa}{}
4414
4415 \providecommand{\maketitlehookaaa}{}
4416

```

```

4417 \newcommand{\prepublished}[1]{%
4418 \def\@bsprepublished{#1}%
4419 }
4420
4421 \newcommand{\postpublished}[1]{%
4422 \def\@bspostpublished{#1}%
4423 }
4424
4425 \newcommand{\presubtitle}[1]{%
4426 \def\@bspresubtitle{#1}%
4427 }

```

`\presubtitle` Hook after printing the subtitle.

```

4428 \newcommand{\postsubtitle}[1]{%
4429 \def\@bspostsubtitle{#1}%
4430 }

```

Initial settings:

```

4431 \if@titlepage
4432 \prepublished{
4433 \vspace*{-\baselineskip}\vspace*{-\medskipamount}\vspace*{-2em}
4434 \begin{center}}
4435 \postpublished{\par\end{center}\vskip 2em}
4436
4437 \presubtitle{\unskip\begin{center}\unskip}
4438 \postsubtitle{\par\end{center}\vskip 2em}
4439 \else
4440 \prepublished{\begin{center}}
4441 \postpublished{\par\end{center}\vskip 0.5em}
4442
4443 \presubtitle{\begin{center}\unskip}
4444 \postsubtitle{\par\end{center}\vskip 0.5em}
4445 \fi
4446 \end{warpprint}

```

## 50.6 \maketitle for HTML output

An HTML div of class `titlepage` is created, inside of which a  $\text{\LaTeX}$  PDF minipage is generated (without HTML tags), allowing the `\thanks` footnotes to be generated immediately at the end of the title page during HTML output. This is desirable when a large table of contents immediately follows the title.

`\thanks` are a form of footnotes used in the title page. See section 43 for other kinds of footnotes.

See `\thanksmarkseries{series}`, below, to set the style of the footnote marks.

for HTML output: 4447 `\begin{warpHTML}`

`\LWR@maketitlesetup` Patches `\thanks` macros to use L<sup>A</sup>T<sub>E</sub>X minipage footnotes.

4448 `\newcommand*{\LWR@maketitlesetup}{%`

Select which kind of footnote marks to use:

4449 `\@bsmarkseries`

4450 `\@mpbsmarkseries`

Redefine the footnote mark:

4451 `\def\@makefnmark{\textsuperscript{\thefootnote}}`

`\thefootnote`  $\Rightarrow$  `\nameuse{arabic}{footnote}`, or  
`\thefootnote`  $\Rightarrow$  `\nameuse{fnsymbol}{footnote}`

Redefine the footnote text:

4452 `\long\def\@makefntext##1{%`

Make the footnote mark and some extra horizontal space for the tags:

4453 `\makethanksmark \LWR@orighspace{1in}`

`\makethanksmark`  $\Rightarrow$  `\thanksfootmark`  $\Rightarrow$  `\tamark`  $\Rightarrow$   
`\@thefnmark`  $\Rightarrow$  `\itshape` a (or similar)

Print the text:

4454 `##1%`

4455 `}%`

4456 `}`

`\@fnsymbol` `{\langle counter \rangle}`

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used `\|` which was not being found by `pdftotext`.

4457 `\def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or`  
4458 `\HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLUnicode{2016}}\or`  
4459 `**\or \HTMLentity{dagger}\HTMLentity{dagger} \or`  
4460 `\HTMLentity{Dagger}\HTMLentity{Dagger} \else\@ctrerr\fi}`



`\maketitle` Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

```
4461 \renewcommand*{\maketitle}{%
```

An HTML titlepage div is used for all classes.

```
4462 \begin{titlepage}
```

Set up special patches:

```
4463 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4464 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
4465 \@thanks
```

Close the HTML titlepage div:

```
4466 \end{titlepage}
```

Reset the footnote counter:

```
4467 \@bscontmark
```

```
4468 }
```

`\@maketitle` Typesets the title, etc. for HTML:

```
4469 \DeclareDocumentCommand{\@maketitle}{}{%
4470   \maketitlehooka
4471   \ifcempty{@published}
4472     {}
4473     {{\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
4474     {\@bsprettitle \@title \@bsposttitle}
4475     \ifcempty{@subtitle}
4476       {}
4477       {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
4478     \maketitlehookb
4479     {\@bspreauthor \@author \@bspostauthor}
4480     \maketitlehookc
4481     {\@bspredate \@date \@bspostdate}
4482     \maketitlehookd
4483 }
```

```
4484 \providecommand{\maketitlehookaa}{}
4485 \providecommand{\maketitlehookaaa}{}

```

`\LWR@titlingmaketitle` `\maketitle` for use inside an HTML `titlingpage` environment.

```
4486 \newcommand*{\LWR@titlingmaketitle}{%

```

Typeset the title, etc:

```
4487 \@maketitle

```

Immediately generate any `\thanks` footnotes:

```
4488 \@thanks
4489 }

```

`\thanksmarkseries` `{\series}`

Sets the type of footnote marks used by `\thanks`, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc. Modified to use the L<sup>A</sup>T<sub>E</sub>X PDF minipage which is included with the title page.

```
4490 \renewcommand{\thanksmarkseries}[1]{%
4491 \def\@mpbsmarkseries{%
4492 \renewcommand*{\thempfootnote}{\@nameuse{#1}{mpfootnote}}}%
4493 \def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
4494 }

4495 \end{warpHTML}

```

## 51 Abstract

The following code replaces the L<sup>A</sup>T<sub>E</sub>X default, and will itself be replaced later if the `abstract` package is loaded.

**for HTML output:** 4496 `\begin{warpHTML}`

`\abstractname` User-redefinable title for the abstract.

Also over-written by the `babel` package.

```
4497 \providecommand*{\abstractname}{Abstract}

```

Env abstract

```

4498 \DeclareDocumentEnvironment{abstract}{}
4499 {
4500 \LWR@forcenewpage
4501 \BlockClass{abstract}
4502 \BlockClassSingle{abstracttitle}{\abstractname}
4503 }
4504 {
4505 \endBlockClass
4506 }

4507 \end{warpHTML}

```

## 52 Quote and verse

### 52.1 Citations and attributions

`\attribution` for use inside quote, quotation, verse:

ex: `\attribution{author name} --- \citetitle{book name}`

**for HTML output:**

```

4508 \begin{warpHTML}
4509 \newcommand{\attribution}[1]{%
4510 \InlineClass{attribution}{--\,#1}}% emdash
4511 \end{warpHTML}

```

**for PRINT output:**

```

4512 \begin{warpprint}
4513 \newcommand{\attribution}[1]{\textsc{---\,#1}}
4514 \end{warpprint}

```

`\citetitle` for use inside quote, quotation, verse:

**for HTML output:**

```

4515 \begin{warpHTML}
4516 \newcommand{\citetitle}[1]{%
4517 \InlineClass{citetitle}{--\,#1}}% emdash
4518 \end{warpHTML}

```

**for PRINT output:**

```

4519 \begin{warpprint}
4520 \newcommand{\citetitle}[1]{\textsl{---\,#1}}
4521 \end{warpprint}

```

## 52.2 Quotes, quotations

for HTML output: 4522 `\begin{warpHTML}`

Env `quote`

```

4523 \renewenvironment*{quote}
4524 {
4525 \LWR@forcenewpage
4526 \LWR@htmlblocktag{blockquote}
4527 }
4528 {\LWR@htmlblocktag{/blockquote}}
4529
4530 \renewenvironment*{quotation}
4531 {
4532 \LWR@forcenewpage
4533 \LWR@htmlblocktag{blockquotation}
4534 }
4535 {\LWR@htmlblocktag{/blockquotation}}

4536 \end{warpHTML}

```

## 52.3 Verse

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

---

```

\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}

```

---

<p>Len <code>\leftskip</code></p> <p>Len <code>\leftmargini</code></p> <p>Len <code>\TMLvleftskip</code></p> <p>Len <code>\TMLleftmargini</code></p>	<p>These lengths are used by <code>verse</code> and <code>memoir</code> to control the left margin, and they may already be set by the user for print output. New lengths <code>\HTMLvleftskip</code> and <code>\HTMLleftmargini</code> are provided to control the margins in HTML output. These new lengths may be set by the user before any <code>verse</code> environment, and persist until they are manually changed again. One reason to change <code>\HTMLleftmargini</code> is if there is a wide <code>\flagverse</code> in use, such as the word “Chorus”, in which case the value of <code>\HTMLleftmargini</code> should be set to a wide enough length to contain “Chorus”. The default is wide enough for a stanza number.</p>
--	--

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

**for HTML & PRINT:** 4537 `\begin{warpall}`

The following lengths may be set in either print or HTML output, but are only used in HTML. This allows the user to set `\vleftskip` and `\leftmargini` for print output, and optionally select different values for HTML.

Len `\TMLvleftskip` Sets `\vleftskip` inside a `verse` environment in HTML.

```
4538 \newlength{\TMLvleftskip}
4539 \setlength{\TMLvleftskip}{1em}
```

Len `\TMLleftmargini` Sets `\leftmargini` inside a `verse` environment in HTML.

```
4540 \newlength{\TMLleftmargini}
4541 \setlength{\TMLleftmargini}{4.5em}

4542 \end{warpall}
```

## 53 Verbatim

**for HTML & PRINT:** 4543 `\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.

```
4544 \newlength{\VerbatimHTMLWidth}
4545 \setlength{\VerbatimHTMLWidth}{4in}
4546 \end{warpall}
```

**for HTML output:** 4547 `\begin{warpHTML}`

Bool `LWR@verbtags` Used to temporarily turn off verbatim tags while doing `\verbatiminput` in the HTML head.

```
4548 \newbool{LWR@verbtags}
4549 \booltrue{LWR@verbtags}
```

`\LWR@atbeginverbatim` [*style*] {*class*}

Encloses a verbatim environment with the given CSS class.

```
4550 \newcommand*{\LWR@atbeginverbatim}[2] []
4551 {%
```

Avoid excessive space between lines:

```
4552 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
4553 \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.

```
4554 \ifbool{LWR@verbtags}{%
4555     \LWR@htmltag{pre class="#2"
4556     \ifthenelse{\equal{#1}{}}{}{}{style="#1"}}%
4557 }% pre
4558 }{}%
```

Use a mono-spaced font to preserve horizontal positioning. If horizontal alignment is important for the user, use a mono-spaced font in the CSS for the `verse` class.

Also turn off `babel-french` extra space before punctuation:

```
4559 \begingroup%
4560 \LWR@origttfamily%
4561 \LWR@FBcancel%
```

Do not produce HTML tags for `\hspace` inside a verse `par`. Restore plain L<sup>A</sup>T<sub>E</sub>X `\hspace` functionality:

```
4562 \LetLtxMacro{\hspace}{\LWR@orighspace}%
4563 }
```

`\LWR@afterendverbatim` Finishes enclosing a verbatim environment.

```
4564 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the `pre`:

```
4565 \endgroup%
4566 \unskip%
```

At the end of the environment, close the `pre`:

```

4567 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}
4568
4569 }{}%

```

Resume regular paragraph handling:

```

4570 \LWR@startpars%
4571 }

```

`\verbatiminput`  $\{\langle filename \rangle\}$

Patch `\verbatiminput` to add HTML tags:

```

4572 \let\LWRV@origverbatim@input\verbatim@input
4573
4574 \renewcommand{\verbatim@input}[2]{%
4575 \ifbool{LWR@verbtags}{\LWR@forcenewpage}{}%
4576 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%%
4577 \LWRV@origverbatim@input{#1}{#2}%
4578 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
4579 }

```

Env `verbatim`

```

4580 \AfterEndPreamble{
4581 \LWR@traceinfo{Patching verbatim.}
4582 \AtBeginEnvironment{verbatim}{%
4583 \LWR@forcenewpage
4584 \LWR@atbeginverbatim{verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
4585 }
4586 \AfterEndEnvironment{verbatim}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
4587 }

4588 \end{warpHTML}

```

## 54 Theorems

`\newtheorem`  $\{\langle text \rangle\}$  [ $\langle counter \rangle$ ] -or- [ $\langle oldname \rangle$ ]  $\{\langle text \rangle\}$

A few minor changes are made to supply HTML tags.

- The entire theorem is placed into a div of class `theoremcontents`.
- The label for each theorem is placed inside a span of class `theoremlabel`.

- The contents are placed inside a div of class `theoremcontents`.

for HTML output: 4589 `\begin{warpHTML}`

`\@begintheorem`  $\{\langle name \rangle\}$   $\{\langle number \rangle\}$

```
4590 \renewcommand{\@begintheorem}[2]{%
4591 \LWR@forcenewpage
4592 \BlockClass{theoremcontents}
4593 \InlineClass{theoremlabel}{#1\ #2\ }
4594 }
```

`\@opargbegintheorem`  $\{\langle name \rangle\}$   $\{\langle number \rangle\}$   $\{\langle oparg \rangle\}$

```
4595 \renewcommand{\@opargbegintheorem}[3]{%
4596 \LWR@forcenewpage
4597 \BlockClass{theoremcontents}
4598 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
4599 }
```

`\@endtheorem`

```
4600 \renewcommand*\@endtheorem{%
4601 \endBlockClass% theoremcontents
4602 }
```

```
4603 \end{warpHTML}
```

## 55 Lists

The environments `itemize`, `enumerate`, and `description` are patched when `lwarp` is started. These patches support the standard  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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}

```

---

## 55.1 Itemize

for HTML output: 4604 \begin{warpHTML}

\LWR@itemizeitem [*label*]

Handles \item inside an itemize or enumerate.

See \LWR@openparagraph where extra \hspace is used to leave room for the label while inside a list during paragraph construction.

```

4605 \newcommand*{\LWR@itemizeitem}{%
4606 \LWR@stoppars%
4607 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloseitem{}}%
4608 \LWR@htmltag{li}%
4609 \LWR@startpars%
4610 \LWR@origitem%
4611 }

```

Env itemize [*options*]

```

4612 \newcommand*{\LWR@itemizestart}{%
4613 \LWR@stoppars%
4614 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4615 \LWR@htmltag{ul style="list-style-type:none">{}}%
4616 \LWR@startpars%
4617 \let\item\LWR@itemizeitem%
4618 }
4619
4620 \newcommand*{\LWR@itemizeend}{%
4621 \LWR@stoppars%
4622 \LWR@closeprevious{\LWR@depthlistitem}%
4623 \LWR@closeoneprevious{}%
4624 \LWR@startpars%
4625 }

```

## 55.2 Enumerate

An HTML unordered list is used with customized L<sup>A</sup>T<sub>E</sub>X-generated labels.

Env `enumerate` [*options*]

```

4626 \newcommand*{\LWR@enumeratestart}{%
4627 \LWR@stoppars%
4628 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4629 \LWR@htmltag{ul style="list-style-type:none">{}}%
4630 \LWR@startpars%
4631 \let\item\LWR@itemizeitem%
4632 }
4633
4634 \newcommand*{\LWR@enumerateend}{%
4635 \LWR@stoppars%
4636 \LWR@closeprevious{\LWR@depthlistitem}%
4637 \LWR@closeoneprevious{}}%
4638 \LWR@startpars%
4639 }
```

## 55.3 Description

`\LWR@descitem` [*label*] Handles an `\item` inside a description.

```

4640 \newcommand*{\LWR@descitem}[1][]%
4641 {%
4642 \LWR@stoppars%
4643 \LWR@setlatestname{#1}%
4644 \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.

```

4645 \LetLtxMacro{\hspace}{\LWR@nohspace}%
```

Process the original `\item` code:

```

4646 \LWR@origitem[]%
```

Restore `\hspace` for use in the item text:

```

4647 \LetLtxMacro{\hspace}{\LWR@hspace}%
```

Be sure the label doesn't print to the left of the rest of the file:

```

4648 \LWR@originspace{1in}
4649 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
4650 \LWR@orignewline%
4651 \LWR@htmltag{dd}%
4652 \LWR@startpars%
4653 }

```

Env **description** [*options*]

```

4654 \newcommand*{\LWR@descriptionstart}{%
4655 \LWR@stoppars%
4656 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}%
4657 \LWR@htmltag{dl}%
4658 \LWR@startpars%
4659 \let\item\LWR@descitem%
4660 }
4661
4662 \newcommand*{\LWR@descriptionend}{%
4663 \LWR@stoppars%
4664 \LWR@closeprevious{\LWR@depthlistitem}%
4665 \LWR@closeoneprevious{}}%
4666 \LWR@startpars%
4667 }

```

**\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<sup>A</sup>T<sub>E</sub>X environments, as well as those provided by **enumitem**, **enumerate**, and **paralist**.

```

4668 \newcommand*{\LWR@patchlists}{%
4669 \LetLtxMacro{\LWR@origitem}{\item}
4670 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
4671 \AtEndEnvironment{itemize}{\LWR@itemizeend}
4672 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
4673 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
4674 \AtBeginEnvironment{description}{\LWR@descriptionstart}
4675 \AtEndEnvironment{description}{\LWR@descriptionend}
4676 }

4677 \end{warpHTML}

```

## 56 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

Limitations:

⚠ misplaced alignment  
alignment tab character &

column types

⚠ `\multirow &`  
`\multicolumn`

⚠ `\multirow`

`\multirow with rules`

rule at last row

⚠ paragraphs

`\cmidrule width, trim`

`\cmidrule and`  
`\multicolumn`

- When defining environments or macros which include `tabular` plus instances of the `&` character, it may be necessary to make `&` active before the environment or macro is defined, then restore `&` to its default catcode after, using the following commands:

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

- Vertical rules are not yet supported.
- `*` in a column specification is not used (so far). Repeat the column type the correct number of times.
- Only one each of `@`, `!`, `>`, and `<` may be used at each column, and they are used in that order.
- `\newcolumnntype` is ignored; unknown column types are set to `l`.
- `tabularx` ignores the width, but `X` columns do produce paragraph columns or multicolumns.
- `Multirow` and `multicolumn` cannot be used at the same time. (No rectangular holes wider than one column or taller than one row.)
- For `multirow`, insert `\mrowcell` into any empty multi-row cells. This will be a null function for the print output, and is a placeholder for parsing the table for HTML output.
- If a `multirow` reaches to the bottom of a table, and `\bottomrule` does not go over to that edge, try adding a line of empty cells below the `\bottomrule`. This may be a browser bug.
- If a `\midrule` is desired after the last row, an additional row of blank cells must be used.
- Multiple paragraphs in one cell of a `p`, `b`, `m` column must have `\newline` between paragraphs.
- `\cmidrule` does not support width or trim options due to CSS limitations.
- `\cmidrule` borders are generated by the individual cells on the following row, and so do not necessarily work correctly when the following row has `\multicolumn` cells below `\cmidrule` borders.

**longtable headings**

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

**⚠ \warpprintonly**

- For `\toprule` and `\bottomrule`, when combined with a `warpprint` or `warppHTML` environment, if a “misplaced `\noalign`” error occurs, change  
`This & That \endhead`  
to  
`\warpprintonly{This & That \endhead}`  
and likewise with the other `\end` headings. Keep the `\endfirsthead` row unchanged, as it is still relevant to HTML output.

**⚠ S columns**

- For S columns (from the `siunitx` package), while producing print output, anything non-numeric must be placed inside `{ }` braces, including commands such as `\multirow`. While producing HTML output, though, anything placed inside braces is not seen by `lwarp`’s tabular handling algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:

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

## 56.1 Token lookahead

Used by `\LWR@futurenonSPACElet` to look at the next token.

**for HTML output:** 4678 `\begin{warppHTML}`

`\LWR@mynexttoken`

4679 `\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`

```
4680 \def\LWR@futurenonSPACElet#1{\def\LWR@cs{#1}%
4681 \afterassignment\LWR@fnslone\let\nexttoken= }
4682 \def\LWR@fnslone{\expandafter\futurelet\LWR@cs\LWR@fnsltwo}
4683 \def\LWR@fnsltwo{%
4684 \expandafter\ifx\LWR@cs\@sptoken\let\next=\LWR@fnslthree%
4685 \else\let\next=\nexttoken\fi\next}
4686 \def\LWR@fnslthree{\afterassignment\LWR@fnslone\let\next= }
```

`\LWR@getmynexttoken` Looks ahead and copies the next token into `\LWR@mynexttoken`.

```
4687 \newcommand*{\LWR@getmynexttoken}{%
4688 \LWR@traceinfo{\LWR@getmynexttoken}%
4689 % nothing must follow this next line
4690 \LWR@futurenonspacel\LR@mynexttoken\LWR@tabledatacolumnntag
4691 }
```

## 56.2 Booleans

Bool `LWR@startedrow` True if should print a row tag before this column.

```
4692 \newbool{LWR@startedrow}
4693 \boolfalse{LWR@startedrow}
```

Bool `LWR@doinghline` True if the next row will have an hline above it.

```
4694 \newbool{LWR@doinghline}
4695 \boolfalse{LWR@doinghline}
```

Bool `LWR@doingtbrule` True if the next row will have a top/bottom rule above it.

```
4696 \newbool{LWR@doingtbrule}
4697 \boolfalse{LWR@doingtbrule}
```

Bool `LWR@tableparcell` True if are handling a paragraph inside a table cell, so must close the paragraph tag before moving on.

```
4698 \newbool{LWR@tableparcell}
```

Bool `LWR@skippingmrowcell` True if are doing an empty multi-row cell, and thus there is no data tag to close.

```
4699 \newbool{LWR@skippingmrowcell}
```

Bool `LWR@skipatbang` True if just finished a `\multicolumn` so should not print the trailing `@` or `!` columns.

```
4700 \newbool{LWR@skipatbang}
```

Bool `LWR@intabularmetadata` True if are in a tabular but not in a data cell. Used to prevent extra HTML breaks if not inside table data.

```
4701 \newbool{LWR@intabularmetadata}
4702 \boolfalse{LWR@intabularmetadata}
```

### 56.3 Handling &, @, and !

For technical discussion regarding problems redefining \&, See:

<http://tex.stackexchange.com/questions/11638/where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860>

`\LWR@closetabledatacell` If `LWR@skippingmrowcell` then there is no data tag to close. Otherwise, close any paragraphs, then close the data tag.

```
4703 \newcommand*{\LWR@closetabledatacell}{%
4704 \global\booltrue{LWR@intabularmetadata}%
4705 \ifbool{LWR@exitingtabular}{}%
4706 {% not exiting tabular
4707     \ifbool{LWR@skippingmrowcell}{}%
4708     {% not skippingmrowcell
```

Insert any < then any @ and ! column contents:

```
4709         \unskip%
4710         \LWR@getexpparray{LWR@colafterspec}{\theLWR@tablecolspos}%
4711 % % \LWR@getexpparray{LWR@colatspec}{\theLWR@tablecolspos}%
4712 % \LWR@printatbang{at}{\theLWR@tablecolspos}%
4713 % % \LWR@getexpparray{LWR@colbangspec}{\theLWR@tablecolspos}%
4714 % \LWR@printatbang{bang}{\theLWR@tablecolspos}%
```

Close paragraphs:

```
4715         \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
4716         \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell. Skip the @ and ! cells if are closing a multicolumn cell.

```
4717         \leavevmode\unskip\LWR@htmltag{/td}\LWR@orignewline%
4718         \ifbool{LWR@skipatbang}%
4719         {\boolfalse{LWR@skipatbang}}%
4720         {%
4721             \LWR@printatbang{at}{\theLWR@tablecolspos}%
4722             \LWR@printatbang{bang}{\theLWR@tablecolspos}%
4723         }% not skipping at or bang
4724     }% not skipping mrowcell
4725 }% not exiting tabular
4726 \global\boolfalse{LWR@skippingmrowcell}%
4727 }
```

`LWR@tabulardepth` tracks whether & is being used inside a `tabular`.

```
4728 \newcounter{LWR@tabulardepth}
```

```
4729 \setcounter{LWR@tabulardepth}{0}
4730
```

When not used inside a `tabular`, `&` performs its original function as recorded here ( with catcode 4 ).

```
4731 \let\LWR@origampmacro&
4732 \end{warpHTML}
```

### 56.3.1 Localizing & catcodes

**for HTML & PRINT:** 4733 \begin{warpall}

 misplaced alignment  
tab character &

Place `\StartDefiningTabulars` and `\EndDefiningTabulars` before and after defining macros or environments which include the tabular `&` character in their definitions.

The catcode of `&` must be changed before the definitions begin, and must be restored afterwards. Doing so avoids the error

```
misplaced alignment tab character &
```

`\StartDefiningTabulars` Place before defining something with `&` in it.

```
4734 \newcommand{\StartDefiningTabulars}{\%
4735 \warpHTMLonly{\catcode'\&=\active}%
4736 }
```

`\EndDefiningTabulars` Place after defining something with `&` in it.

```
4737 \newcommand{\EndDefiningTabulars}{\%
4738 \warpHTMLonly{\catcode'\&=4}%
4739 }
```

```
4740 \end{warpall}
```

### 56.3.2 Handling &

**for HTML output:** 4741 \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.

```
4742 \newcommand*\LWR@tabularampersand}{%
4743 \LWR@traceinfo{\LWR@tabularampersand}%
4744 \ifthenelse{\cnttest{\value{\LWR@tabulardepth}}{>}{0}}%
4745 {%
```

If not skipping a multirow cell, close the current data cell.

```
4746 \unskip%
4747 \LWR@closetabledatacell%
```

Move to the next column.

```
4748 \addtocounter{\LWR@tablecolspos}{1}%
```

Look at the next token to decide multi or single column data tag.

```
4749 \LWR@getmynexttoken%
4750 }%
```

If not inside a tabular, performs the original action:

```
4751 {\LWR@origampmacro}%
4752 }
```

`&` 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.

## 56.4 Handling `\\`

Inside `tabular`, `\\` is redefined to `\LWR@tabularendoffline`

Throws away options `\\[dim]` or `\\*`

```
\LWR@tabularendoffline
```

```
4753 \NewDocumentCommand{\LWR@tabularendoffline}{s o}
4754 {%
4755 \LWR@closetabledatacell%
```

Finish the previous row:

```
4756 \LWR@htmltag{/tr}\LWR@orignewline
4757 \global\booltrue{LWR@intabularmetadata}
```

Not yet started a table row:

```
4758 \global\boolfalse{LWR@startedrow}
```

Additional setup:

```
4759 \global\boolfalse{LWR@doinghline}%
4760 \global\boolfalse{LWR@doingtbrule}%
4761 \LWR@clearmidrules%
```

Start at first column:

```
4762 \setcounter{LWR@tablecolspos}{1}
```

Look at the next token to decide between single column data tag or a special case:

```
4763 \LWR@getmynexttoken%
4764 }
```

## 56.5 Variables

```
4765 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
4766 \newcommand*{\LWR@pposition}{}
4767 \newcommand*{\LWR@pleft}{}
4768 \newcommand*{\LWR@pright}{}

```

**\LWR@tablecolspec** Holds the parsed column specification, of total width `LWR@tabletotalcols`, not counting `@` and `!` columns.

Will contain a string such as `llrrccpc`, exactly one letter per  $\text{\LaTeX}$  table column, without `@`, `!`, `>`, `<`, or the vertical pipe.

This is indexed by the counter `LWR@tabletotalcols`.

```
4769 \newcommand*{\LWR@tablecolspec}{}

```

**\LWR@strresult** Holds the result of `Str` functions.

```
4770 \newcommand*{\LWR@strresult}{}
4771 \newcommand*{\LWR@strresulttwo}{}

```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

4772 `\newcommand*{\LWR@origcolspec}{}`

Ctr `LWR@tablecolwidth` Holds the width of the table columns specification.

This is the number of tokens, including one for each `@` etc. column, and also one each for the parameters of `p`, `@`, etc. columns, and three for each `D` column.

(This is not the total # of L<sup>A</sup>T<sub>E</sub>X columns in the table.)

4773 `\newcounter{LWR@tablecolwidth}`

Ctr `LWR@tablecolspos` Where are currently looking into the table column specification. Index starts at 1.

4774 `\newcounter{LWR@tablecolspos}`

Ctr `LWR@tabletotalcols` Holds the final number of table columns, not counting `@` and `!` columns.

4775 `\newcounter{LWR@tabletotalcols}`

Ctr `LWR@tabletotalcolsnext` Holds the next column while parsing. Is one more than `LWR@tabletotalcols`.

4776 `\newcounter{LWR@tabletotalcolsnext}`

`LWR@colatspec` A data array of specifications for `@` columns. The leftmost's index is `leftedge`, the others are counter values. See section 30.

`LWR@colbangspec` A data array of specifications for `!` columns. The leftmost's index is `leftedge`, the others are counter values. See section 30.

`LWR@colbeforespec` A data array of specifications for `>` columns.

`LWR@colafterspec` A data array of specifications for `<` columns.

## 56.6 Parsing @, >, <, ! columns

Holds the parsed argument for `@`, `>`, `<`, or `!` columns:

4777 `\newcommand*{\LWR@colparameter}{}`

`\LWR@parseatcolumn` Handles `@{text}` columns.

4778 `\newcommand*{\LWR@parseatcolumn}{%`

Move to the next token after the `'@'`:

```

4779 \LWR@traceinfo{at column}%
4780 \addtocounter{LWR@tablecolspos}{1}%

```

Read the next token into \LWR@strresult, expanding once:

```

4781 \LWR@traceinfo{about to read the next token:}%
4782 \expandarg%
4783 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]
4784 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@strresult:

```

4785 \LWR@traceinfo{have now read the next token}%
4786 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}{
4787 {% left edge of the table:
4788     \LWR@traceinfo{at the left edge}%
4789     \LWR@setexparray{LWR@colatspec}{\leftedge}{\LWR@colparameter}%
4790     \LWR@traceinfo{at the left edge: %
4791     \LWR@getexparray{LWR@colatspec}{\leftedge}}%
4792 }%
4793 {% not at the left edge:
4794     \LWR@traceinfo{not at the left edge}%
4795     \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
4796     \LWR@traceinfo{at \theLWR@tabletotalcols: %
4797     \LWR@getexparray{LWR@colatspec}{\theLWR@tabletotalcols}}%
4798 }%
4799 \let\LWR@colparameter\relax%
4800 \booltrue{LWR@validtablecol}%
4801 }

```

\LWR@parsebangcolumn Handles !{text} columns.

```

4802 \newcommand*{\LWR@parsebangcolumn}{%

```

Move to the next token after the '!':

```

4803 \LWR@traceinfo{bang column}%
4804 \addtocounter{LWR@tablecolspos}{1}%

```

Read the next token into \LWR@strresult, expanding once:

```

4805 \LWR@traceinfo{about to read the next token:}%
4806 \expandarg%
4807 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]
4808 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@strresult:

```

4809 \LWR@traceinfo{have now read the next token}%
4810 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}{
4811 {% left edge of the table:
4812     \LWR@traceinfo{at the left edge}%
4813     \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@colparameter}%
4814 }%
4815 {% not at the left edge:
4816     \LWR@traceinfo{not at the left edge}%
4817     \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
4818     \LWR@traceinfo{bang \theLWR@tabletotalcols: \LWR@colparameter!}%
4819 }%
4820 \let\LWR@colparameter\relax%
4821 \booltrue{LWR@validtablecol}%
4822 }

```

`\LWR@parsebeforecolumn` Handles `>{text}` columns.

```

4823 \newcommand*{\LWR@parsebeforecolumn}{%

```

Move to the next token after the `'>'`:

```

4824 \addtocounter{LWR@tablecolspos}{1}%

```

Read the next token, expanding once into `\LWR@strresult`:

```

4825 \expandarg%
4826 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]%
4827 \fullexpandarg%

```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```

4828 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\LWR@colparameter}%
4829 \let\LWR@colparameter\relax%
4830 \booltrue{LWR@validtablecol}%
4831 }

```

`\LWR@parseaftercolumn` Handles `<{text}` columns.

```

4832 \newcommand*{\LWR@parseaftercolumn}{%

```

Move to the next token after the `'<'`:

```

4833 \addtocounter{LWR@tablecolspos}{1}%

```

Read the next token, expanding once into `\LWR@strresult`:

```

4834 % \StrChar{#1}{\theLWR@tablecolspos}[\LWR@colparameter]

```

```

4835 \expandarg%
4836 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]%
4837 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@strresult:

```

4838 \LWR@setexparray{\LWR@colafterspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
4839 \let\LWR@colparameter\relax%
4840 \booltrue{\LWR@validtablecol}%
4841 }

```

`\LWR@parseskipcolumn` Handles columns to skip, such as the vertical bar.

```

4842 \newcommand*{\LWR@parseskipcolumn}{%
4843 \booltrue{\LWR@validtablecol}%
4844 }

```

## 56.7 Parsing ‘l’, ‘c’, or ‘r’ columns

`\LWR@parsenormalcolumn`  $\{\langle thiscolumn \rangle\}$

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```

4845 \newcommand*{\LWR@parsenormalcolumn}[1]{%
4846 \appto\LWR@tablecolspec{#1}%
4847 \addtocounter{\LWR@tabletotalcols}{1}%
4848 \addtocounter{\LWR@tabletotalcolsnext}{1}%
4849 \LWR@traceinfo{normal column \theLWR@tabletotalcols: #1}%
4850 \LWR@setexparray{\LWR@colatspec}{\theLWR@tabletotalcolsnext}{}%
4851 \LWR@setexparray{\LWR@colbangspec}{\theLWR@tabletotalcolsnext}{}%
4852 \LWR@setexparray{\LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{}%
4853 \LWR@setexparray{\LWR@colafterspec}{\theLWR@tabletotalcolsnext}{}%
4854 \booltrue{\LWR@validtablecol}%
4855 }

```

## 56.8 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn`  $\{\langle thiscolumn \rangle\}$  The width will be ignored.

```

4856 \newcommand*{\LWR@parsepcolumn}[1]{%

```

Converts to the given column type:

```
4857 \LWR@parsenormalcolumn{#1}%
```

Skips the following width token:

```
4858 \addtocounter{LWR@tablecolspos}{1}%
4859 }
```

## 56.9 Parsing ‘D’ columns

From the dcolumn package.

`\LWR@parseDcolumn`  $\{(\textit{thiscolumn})\}$  The three parameters will be ignored.

```
4860 \newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
4861 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
4862 \addtocounter{LWR@tablecolspos}{3}%
4863 }
```

## 56.10 Parsing the column specifications



HTML CSS cannot exactly match the  $\text{\LaTeX}$  concept of a baseline for a table row. Table 7 shows the  $\text{\LaTeX}$  results for various vertical-alignment choices, with the baseline of the first column drawn across all the columns for comparison. See the p column specification in table 8 for details.

Table 8 describes how each kind of column is converted to HTML.

Bool `LWR@validtablecol` True if found a valid table column type.

```
4864 \newbool{LWR@validtablecol}
```

`\LWR@parsetablecols`  $\{(\textit{colspecs})\}$

Scans the column specification left to right.

Builds `\LWR@tablecolspec` with the final specification, one column per entry. The number of final columns is stored in `LWR@tabletotalcols`.

Table 7: Tabular baseline

l	p	m	b	r
			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

Table 8: Tabular HTML column conversions

- l, r, c:**

Converted to table cells without paragraph tags.  
Uses CSS `vertical-align:middle` so that top or bottom-aligned cells may go above or below this cell.
- p:**

Converted to table cells with paragraph tags. Ref: Table 7,  $\text{\LaTeX}$  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  $\text{\LaTeX}$  baselines.
- m:**

With paragraph tags, CSS `vertical-align:middle`.
- b:**

With paragraph tags, CSS `vertical-align:top` so that the bottom of the text is closest to the middle of the text line.
- P, M, B:**

Horizontally-centered versions.
- S:**

Converted to 'r'. From the `siunitx` package.
- D:**

Converted to 'c'. From the `dcolumn` package.
- @, !, >, <:**

One each, in that order.
- Unknown:**

Converted to 'l'.
- \newcolumn:**

Currently treated as unknown.



```
4865 \newcommand*{\LWR@parsetablecols}[1]{%
4866 \LWR@traceinfo{\LWR@parsetablecols started}%
```

Remember the original supplied column spec:

```
4867 \renewcommand*{\LWR@origcolspec}{#1}%
```

Clear the parsed resulting column spec:

```
4868 \renewcommand*{\LWR@tablecolspec}{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
4869 \setcounter{\LWR@tabletotalcols}{0}%
4870 \setcounter{\LWR@tabletotalcolsnext}{1}%
4871 \LWR@setexparray{\LWR@colatspec}{leftedge}{}%
4872 \LWR@setexparray{\LWR@colatspec}{1}{}%
4873 \LWR@setexparray{\LWR@colatspec}{2}{}%
4874 \LWR@setexparray{\LWR@colatspec}{3}{}%
4875 \LWR@setexparray{\LWR@colbangspec}{leftedge}{}%
4876 \LWR@setexparray{\LWR@colbangspec}{1}{}%
4877 \LWR@setexparray{\LWR@colbangspec}{2}{}%
4878 \LWR@setexparray{\LWR@colbangspec}{3}{}%
4879 \LWR@setexparray{\LWR@colbeforespec}{1}{}%
4880 \LWR@setexparray{\LWR@colbeforespec}{2}{}%
4881 \LWR@setexparray{\LWR@colbeforespec}{3}{}%
4882 \LWR@setexparray{\LWR@colafterspec}{1}{}%
4883 \LWR@setexparray{\LWR@colafterspec}{2}{}%
4884 \LWR@setexparray{\LWR@colafterspec}{3}{}%
```

Starting at the first column specification:

```
4885 \setcounter{\LWR@tablecolspos}{1}%
```

Place the colspecs string length into \LWR@strresult, and remember the number of characters in the column specification:

```
4886 \LWR@traceinfo{about to StrLen}%
4887 \noexpandarg%
4888 \StrLen{#1}[\LWR@strresult]%
4889 \fullexpandarg%
4890 \LWR@traceinfo{finished StrLen}%
4891 \setcounter{\LWR@tablecolwidth}{\LWR@strresult}%
```

Scan through the column specifications:

```
4892 \whiledo{\not\value{\LWR@tablecolspos}>\value{\LWR@tablecolwidth}}{%
```

Place the next single-character column type into \LWR@strresult:

```
4893 \noexpandarg%
4894 \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]%
4895 \LWR@traceinfo{position \arabic{LWR@tablecolspos}: \LWR@strresult}%
4896 \fullexpandarg%
```

Not yet found a valid column type

```
4897 \boolfalse{LWR@validtablecol}%
```



Note that the parameter for a p{spec} column is a token list which will NOT match l,c,r,p.

```
4898 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{}%
4899 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{}%
4900 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{}%
4901 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{}%
4902 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{}%
4903 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{}%
4904 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{}%
4905 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{}%
4906 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
4907 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
4908 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
4909 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
4910 \IfStrEq{\LWR@strresult}{|}{\LWR@parseskipcolumn}{}%
4911 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
4912 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
4913 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%
```

From the dcolumn package:

```
4914 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{}%
```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```
4915 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%
```

---

Many people define centered versions “P”, “M”, and “B”:

`\newcolumnstype{P}[1]{>\centering\arraybackslash}p{#1}`

---

```
4916 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
4917 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
4918 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%
```

If this column was an invalid column type, convert it to an l column:

```

4919 \ifbool{LWR@validtablecol}{}{%
4920     \LWR@traceinfo{invalid column type: \LWR@strresult}%
4921     \LWR@parsenormalcolumn{1}%
4922 }%
4923 \addtocounter{LWR@tablecolspos}{1}%
4924 }%
4925 }%
```

## 56.11 Starting a new row

`\LWR@maybenewtablerow` If have not yet started a new table row, begin one now. Creates a new row tag, adding a class for `hline` or `tbrule` if necessary.

```

4926 \newcommand*{\LWR@maybenewtablerow}
4927 {%
4928 \ifbool{LWR@startedrow}%
4929 {}% started the row
4930 {}% not started the row
```

Remember that now have started the row:

```

4931     \global\booltrue{LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```

4932     \global\booltrue{LWR@intabularmetadata}%
4933     \ifbool{LWR@doinghline}%
4934     {\LWR@htmltag{tr class="hline"{} }\LWR@orignewline}%
4935     {% not doing hline
4936         \ifbool{LWR@doingtbrule}%
4937         {\LWR@htmltag{tr class="tbrule"{} }\LWR@orignewline}%
4938         {\LWR@htmltag{tr} \LWR@orignewline}%
4939     }% end of not doing hline
4940 }% end of not started the row
4941 }
```

## 56.12 Printing at or bang tags

`\LWR@printatbang`  $\{ \langle at -or- bang \rangle \} \{ \langle index \rangle \}$

```

4942 \newcommand*{\LWR@printatbang}[2]{%
4943 \edef\LWR@atbangspec{\LWR@getexpparray{LWR@col#1spec}{#2}}
```

```

4944 \LWR@traceinfo{atbang: !\LWR@atbangspec!}
4945 \ifdefempty{\LWR@atbangspec}%
4946 % \ifthenelse{\isempty{\LWR@atbangspec}}}%
4947 {}%
4948 {%
4949   \LWR@htmltag{td class="td#1%
4950   \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}}%
4951   "%
4952   \LWR@atbangspec%
4953   \LWR@htmltag{/td}\LWR@orignewline%
4954 }%
4955 }%

```

### 56.13 Data opening tag

`\LWR@tabledatasinglecolumnntag` Print a table data opening tag with style for alignment

```

4956 \newcommand*{\LWR@tabledatasinglecolumnntag}%
4957 {%
4958 \LWR@maybenewtablerow%

```

Don't start a new paragraph tag if have already started one:

```

4959 \ifbool{\LWR@intabularmetadata}%
4960 {%

```

If have found the end of tabular command, do not create the next data cell:

```

4961   \ifbool{\LWR@exitingtabular}{}%
4962   {% not exiting tabular

```

Print the @ and ! contents before first column:

```

4963   \ifthenelse{\cnttest{\value{\LWR@tablecolspos}}=1}%
4964   {%
4965     \LWR@printatbang{at}{leftedge}%
4966     \LWR@printatbang{bang}{leftedge}%
4967   }% left edge
4968   {% not left edge

```

Fetch the current column's alignment character into `\LWR@strresult`:

```

4969   \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}[\LWR@strresult]%

```

print the start of a new table data cell:

```

4970   \LWR@htmltag{td class="td%

```

append this column's spec:

```
4971      \LWR@strresult%
```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag:

```
4972      \ifthenelse{\equal{\LWR@gettexpparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{%
4973      {rule}}%
4974      {}%
4975      "{}}%
```

If this is a p, m, b, or X column, allow paragraphs:

```
4976      \ifthenelse{%
4977      \equal{\LWR@strresult}{p}\OR%
4978      \equal{\LWR@strresult}{m}\OR%
4979      \equal{\LWR@strresult}{b}\OR%
4980      \equal{\LWR@strresult}{P}\OR%
4981      \equal{\LWR@strresult}{M}\OR%
4982      \equal{\LWR@strresult}{B}\OR%
4983      \equal{\LWR@strresult}{X}}%
4984      }%
4985      {% allow pars
4986      \LWR@startpars%
4987      \global\booltrue{\LWR@tableparcell}}%
4988      }% allow pars
4989      {}% no pars
```

Print the > contents:

```
4990      \LWR@gettexpparray{\LWR@colbeforespec}{\theLWR@tablecolspos}%
4991      \global\boolfalse{\LWR@intabularmetadata}%
4992      }% not exiting tabular
4993 }{}% in tabular metadata
4994 }%
```

## 56.14 Midrules

**LWR@midrules** `LWR@midrules` is a data array (section 30) of columns containing Y if a midrule should be created for each column.

**Ctr LWR@midrulecounter** Indexes across the `LWR@midrules` data array.

```
4995 \newcounter{\LWR@midrulecounter}
```

**\LWR@clearmidrules** Start new midrules. Called at beginning of tabular and also at `\.`

Clears all LWR@midrules markers for this line.

```

4996 \newcommand*{\LWR@clearmidrules}
4997 {%
4998 \setcounter{LWR@midrulecounter}{1}%
4999 \whiledo{%
5000 \cntttest{\value{LWR@midrulecounter}}{<=}{\value{LWR@tablecolswidth}}%
5001 }%
5002 {%
5003 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{}%
5004 \addtocounter{LWR@midrulecounter}{1}%
5005 }%
5006 }

```

`\LWR@subcmidrule`  $\{\langle width \rangle\} \{\langle trim \rangle\} \{\langle leftcolumn \rangle\} \{\langle rightcolumn \rangle\}$

Marks LWR@midrules data array elements to be “Y” from left to right columns.

```

5007 \newcommand*{\LWR@subcmidrule}[4]{%
5008 \setcounter{LWR@midrulecounter}{#3}%
5009 \whiledo{\cntttest{\value{LWR@midrulecounter}}{<=}{#4}}%
5010 {%
5011 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{Y}%
5012 \addtocounter{LWR@midrulecounter}{1}%
5013 }% end of the whiledo
5014 }

```

`\LWR@docmidrule`  $[\langle width \rangle] \{\langle trim \rangle\} \{\langle leftcolumn-rightcolumn \rangle\}$

Marks LWR@midrules array elements to be “Y” from left to right columns.

```

5015 \NewDocumentCommand{\LWR@docmidrule}{o d() >{\SplitArgument{1}{-}}m}%
5016 {\LWR@subcmidrule{#1}{#2}{#3}}

```

## 56.15 Multicolumns

### 56.15.1 Parsing multicolumns

```

5017 \newcounter{LWR@tablemulticolswidth}

```

Indexes into the multicolumn specification:

```

5018 \newcounter{LWR@tablemulticolspos}

```

`\LWR@printmccoltype`  $\{\langle colspec \rangle\}$  Print any valid column type found. Does not print @, !, >, or < columns or their associated tokens.

This is printed as part of the table data tag's class.

```
5019 \newcommand*{\LWR@printmccoltype}[1]{%
5020 \LWR@traceinfo{lw@printmccoltype -#1-}%
```

Get one token of the column spec:

```
5021 \StrChar{#1}{\the\LWR@tablemulticolspos}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```
5022 \IfStrEq{\LWR@strresult}{l}{l}{}%
5023 \IfStrEq{\LWR@strresult}{c}{c}{}%
5024 \IfStrEq{\LWR@strresult}{r}{r}{}%
5025 \IfStrEq{\LWR@strresult}{p}{p}{}%
5026 \IfStrEq{\LWR@strresult}{m}{m}{}%
5027 \IfStrEq{\LWR@strresult}{b}{b}{}%
5028 \IfStrEq{\LWR@strresult}{P}{P}{}%
5029 \IfStrEq{\LWR@strresult}{M}{M}{}%
5030 \IfStrEq{\LWR@strresult}{B}{B}{}%
5031 \IfStrEq{\LWR@strresult}{S}{r}{}%
5032 \IfStrEq{\LWR@strresult}{X}{p}{}%
5033 \LWR@traceinfo{lw@printmccoltype done}%
5034 }
```

`\LWR@multicolpartext` Print the data with paragraph tags:

```
5035 \newcommand*{\LWR@multicolpartext}{%
5036 \LWR@startpars%
5037 \LWR@multicoltext%
5038 \LWR@stoppars%
5039 }
```

`\LWR@multicolother`  $\{\langle colspec \rangle\}$  For @, !, >, <, print the next token without paragraph tags:

```
5040 \newcommand*{\LWR@multicolother}[1]{%
5041 \addtocounter{LWR@tablemulticolspos}{1}%
5042 \StrChar{#1}{\the\LWR@tablemulticolspos}[\LWR@strresult]%
5043 \LWR@strresult%
```

A valid column data type was found:

```
5044 \booltrue{LWR@validtablecol}%
5045 }
```

`\LWR@multicolskip` Nothing to print for this column type.

```
5046 \newcommand*{\LWR@multicolskip}{%
```

A valid column data type was found:

```
5047 \booltrue{LWR@validtablecol}%
5048 }
```

`\LWR@printmccoldata {(<colspec>}` Print the data for any valid column type found.

```
5049 \newcommand*{\LWR@printmccoldata}[1]{%
5050 \LWR@traceinfo{lwarp@printmccoldata -#1}%
```

Not yet found a valid column type:

```
5051 \boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
5052 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
5053 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
5054 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
5055 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
5056 \IfStrEq{\LWR@strresult}{D}{}%
5057 \addtocounter{LWR@tablemulticolspos}{3}% skip parameters
5058 \LWR@multicoltext%
5059 }{}%
5060 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolparttext}{}%
5061 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolparttext}{}%
5062 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolparttext}{}%
5063 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolparttext}{}%
5064 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolparttext}{}%
5065 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolparttext}{}%
5066 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolparttext}{}%
5067 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolparttext}{}%
5068 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
5069 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@multicolother{#1}}{%
5070 \IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}{%
5071 \IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}{%
5072 \IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}{%
```

If an invalid column type:

```
5073 \ifbool{LWR@validtablecol}{\LWR@multicoltext}%
```

Tracing:



```
5074 \LWR@traceinfo{lwr@printmccoldata done}%
5075 }
```

```
\parsemulticolumnalignment {\langle 1: colspec \rangle} {\langle 2: printresults \rangle}
```

Scan the multicolumn specification and execute the printfunction for each entry.

Note that the spec for a `p{spec}` column, or `@`, `>`, `<`, is a token list which will NOT match `l`, `c`, `r`, or `p`.

```
5076 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
5077 \setcounter{LWR@tablemulticolspos}{1}%
5078 \StrLen{#1}[\LWR@strresult]%
5079 \setcounter{LWR@tablemulticolwidth}{\LWR@strresult}%
```

Scan across the tokens in the column spec:

```
5080 \whiledo{%
5081 \not\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolwidth}%
5082 }%
5083 {%
```

Execute the assigned print function for each token in the column spec:

```
5084 #2{#1}%
```

Move to the next token in the column spec:

```
5085 \addtocounter{LWR@tablemulticolspos}{1}%
5086 }%
5087 }
```

### 56.15.2 High-level multicolumn interface

```
5088 \newcommand{\LWR@multicoltext}{}%
```

```
\LWR@domulticolumn {\langle 1: numcols \rangle} {\langle 2: colspec \rangle} {\langle 3: text \rangle}
```

```
5089 \NewDocumentCommand{\LWR@domulticolumn}{m m +m}{%
5090 \LWR@traceinfo{lwr@domulticolumn -#1- -#2-}%
```

Remember the text to be inserted, and remember that a valid column type was found:

```
5091 \renewcommand{\LWR@multicoltext}{%
5092 #3%
5093 \booltrue{LWR@validtablecol}%
5094 }%
```

Row processing:

```
5095 \LWR@maybe newtable row%
```

Begin the opening table data tag:

```
5096 \LWR@htmltag{td colspan="#1"
5097 class="td%
```

Print the column type:

```
5098 \LWR@parse multicolumn alignment{#2}{\LWR@print mcol type}%
```

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”.

```
5099 \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
```

Close the class tag’s opening quote:

```
5100 "%
5101 }% end of the opening table data tag
5102 \global\boolfalse{\LWR@intabular metadata}%
5103 \LWR@parse multicolumn alignment{#2}{\LWR@print mcol data}%
5104 }
```

### 56.15.3 Longtable captions

`Bool LWR@starredlongtable` Per the caption package, step the counter if `longtable*`.

```
5105 \newbool{\LWR@starredlongtable}
5106 \boolfalse{\LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
5107 \providecommand*{\LTcaptype}{table}
```

```
\LWR@longtable data caption tag * [(<toc entry>)] {(<caption>)}
```

```
5108 \NewDocumentCommand{\LWR@longtable data caption tag}{s o +m}
5109 {%
```

Remember the latest name for `\nameref`:

```
5110 \IfValueTF{#2}{% optional given?
```

```

5111 \ifthenelse{\equal{#2}{}}{% optional empty?
5112 {\LWR@setlatestname{#3}}% empty
5113 {\LWR@setlatestname{#2}}% given and non-empty
5114 }% optional given
5115 {\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:

```

5116 \LWR@tabularhtmlcolumns{1}{\theLWR@tabletotalcols}

```

Create the multicolumn tag:

```

5117 \LWR@domulticolumn{\theLWR@tabhtmlcoltotal}{P}{% \LWR@domulticolumn
5118 \IfBooleanTF{#1}}% star?

```

Star version, show a caption but do not make a LOT entry:

```

5119 {% yes star
5120 \LWR@htmlblocktag{figcaption}%
5121 #3%
5122 \LWR@htmlblocktag{/figcaption}%
5123 }%
5124 {% No star:

```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```

5125 \ifbool{LWR@starredlongtable}%
5126 {%
5127 \ifthenelse{\equal{#2}{}}{% TOC entry
5128 }%
5129 {%
5130 \refstepcounter{LTcaption}%
5131 \protected@edef\@currentlabel{%
5132 \csuse{p@LTcaption}\csuse{the\LTcaption}}%
5133 }%
5134 }{}%

```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```

5135 \LWR@htmlblocktag{figcaption}%
5136 \csuse{fnum@LTcaption}\CaptionSeparator#3%
5137 \LWR@htmlblocktag{/figcaption}%

```

See if an optional caption was given:

```
5138 \ifthenelse{\equal{#2}{}}{% TOC entry empty
```

if the optional caption was given, but empty, do not form a TOC entry

```
5139 {}%
```

If the optional caption was given, but might only be []:

```
5140 {% TOC entry not empty
```

```
5141 \IfNoValueTF{#2}% No TOC entry?
```

The optional caption is []:

```
5142 {% No TOC entry
```

```
5143 \addcontentsline%
```

```
5144 {\csuse{ext@LTcapttype}}%
```

```
5145 {\LTcapttype}%
```

```
5146 {%
```

```
5147 \protect\numberline%
```

```
5148 {\csuse{p@LTcapttype}\csuse{theLTcapttype}}%
```

```
5149 {\ignorespaces #3\protect\relax}%
```

```
5150 }%
```

```
5151 }% end of No TOC entry
```

The optional caption has text enclosed:

```
5152 {% yes TOC entry
```

```
5153 \addcontentsline%
```

```
5154 {\csuse{ext@LTcapttype}}%
```

```
5155 {\LTcapttype}%
```

```
5156 {%
```

```
5157 \protect\numberline%
```

```
5158 {\csuse{p@LTcapttype}\csuse{theLTcapttype}}%
```

```
5159 {\ignorespaces #2\protect\relax}%
```

```
5160 }%
```

```
5161 }% end of yes TOC entry
```

```
5162 }% end of TOC entry not empty
```

```
5163 }% end of no star
```

Skip any trailing @ or ! columns for this cell:

```
5164 \booltrue{LWR@skipatbang}%
```

```
5165 }% end of \LWR@domulticolumn
```

```
5166
```

```
5167 \addtocounter{LWR@tablecolspos}{\theLWR@tabletotalcols}
```

```
5168 \addtocounter{LWR@tablecolspos}{-1}
```

```
5169
```

```
5170 }
```

#### 56.15.4 Counting HTML tabular columns

The  $\text{\LaTeX}$  specification for a table includes a number of columns separated by the  $\&$  character. These columns differ in content from line to line. Additional virtual columns may be specified by the special  $\@$  and  $!$  columns. These columns are identical from line to line, but may be skipped during a multicolumn cell.

For HTML output,  $\@$  and  $!$  columns are placed into their own tabular columns. Thus, a  $\text{\LaTeX}$  `\multicolumn` command may span several additional  $\@$  and  $!$  columns in HTML output. These additional columns must be added to the total number of columns spanned by an HTML multi-column data cell.

```
5171 \newcounter{LWR@tabhtmlcolindex}
5172 \newcounter{LWR@tabhtmlcolend}
5173 \newcounter{LWR@tabhtmlcoltotal}
```

```
\LWR@tabularhtmlcolumns {\langle starting \text{\LaTeX} column \rangle} {\langle number \text{\LaTeX} columns \rangle}
```

Compute the total number of HTML columns being spanned, considering the starting  $\text{\LaTeX}$  table column and the number of  $\text{\LaTeX}$  tabular columns being spanned. Any  $\@$  and  $!$  columns within this span are included in the total count. The resulting number of HTML columns is returned in the counter `LWR@tabhtmlcoltotal`.

```
5174 \newcommand*{\LWR@tabularhtmlcolumns}[2]{%
```

Count the starting index, compute ending index, and begin with the count being the  $\text{\LaTeX}$  span, to which additional  $\@$  and  $!$  columns may be added:

```
5175 \setcounter{LWR@tabhtmlcolindex}{#1}%
5176 \setcounter{LWR@tabhtmlcoltotal}{#2}%
5177 \setcounter{LWR@tabhtmlcolend}{#1}%
5178 \addtocounter{LWR@tabhtmlcolend}{#2}%
```

Walk across the  $\text{\LaTeX}$  columns looking for  $\@$  and  $!$  columns:

```
5179 \whiledo{\value{LWR@tabhtmlcolindex}<\value{LWR@tabhtmlcolend}}{%
```

Temporarily define a macro equal to the  $\@$  specification for this column:

```
5180 \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{\theLWR@tabhtmlcolindex}}%
```

If the  $\@$  specification is not empty, add to the count:

```
5181 \ifdefempty{\LWR@atbangspec}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the  $!$  columns:

```

5182 \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{\theLWR@tabhtmlcolindex}}%
5183 \ifdefempty{LWR@atbangspec}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%

```

Move to the next L<sup>A</sup>T<sub>E</sub>X column:

```

5184 \addtocounter{LWR@tabhtmlcolindex}{1}%
5185 }%

```

If at the left-most column, also skip the leftmost @ and ! cells:

```

5186 \ifthenelse{\value{LWR@tablecolspos}=1}{%
5187   \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{leftedge}}%
5188   \ifdefempty{LWR@atbangspec}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
5189   \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{leftedge}}%
5190   \ifdefempty{LWR@atbangspec}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
5191 }{}%
5192 }

```

### 56.15.5 \tabledatamulticolumntag

`\LWR@tabledatamulticolumntag` {*<numcols>*} {*<alignment>*} {*<text>*}

```

5193 \NewDocumentCommand{\LWR@tabledatamulticolumntag}{m m +m}%
5194 {}%

```

Figure out how many extra HTML columns to add for @ and ! columns:

```

5195 \LWR@tabularhtmlcolumns{\theLWR@tablecolspos}{#1}

```

Create the multicolumn tag:

```

5196 \LWR@domulticolumn{\theLWR@tabhtmlcoltotal}{#2}{#3}%

```

Move to the next L<sup>A</sup>T<sub>E</sub>X column:

```

5197 \addtocounter{LWR@tablecolspos}{#1}%
5198 \addtocounter{LWR@tablecolspos}{-1}%

```

Skip any trailing @ or ! columns for this cell:

```

5199 \booltrue{LWR@skipatbang}%
5200 }

```

## 56.16 Multirow

Pkg multirow

`\LWR@tabledatamultirowtag`  $\{\langle numrows \rangle\}$   $[\langle bigstruts \rangle]$   $\{\langle width \rangle\}$   $[\langle fixup \rangle]$   $\{\langle text \rangle\}$

```
5201 \NewDocumentCommand{\LWR@tabledatamultirowtag}{m o m o m}%
5202 {%
5203 \LWR@maybenewtablerow%
```

Print the start of a new table data cell:

```
5204 \LWR@htmltag{td rowspan="#1" class="td%
```

Append this column's spec:

```
5205 \StrChar{\LWR@tablecolspec}{\the\LWR@tablecolspos}%
```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag:

```
5206 \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}{\the\LWR@tablecolspos}}{Y}}{rule}{}%
5207 "{}}
```

While printing the text, redefine `\` to generate a new line

```
5208 \begingroup \LetLtxMacro{\}{\LWR@endoffline} #5 \endgroup
5209 \LWR@stoppars%
5210 \global\boolfalse{\LWR@intabularmetadata}%
5211 }%
```

## 56.17 Utility macros inside a table

```
5212 \newcommand*{\LWR@donothing}{}
5213 \newcommand*{\LWR@domidrule}{\booltrue{\LWR@doinghline}}
5214 \newcommand*{\LWR@dotbrule}{\booltrue{\LWR@doingtbrule}}
```

## 56.18 Checking for a new table cell

`\LWR@tabledatacolumnntag` Open a new HTML table cell unless the next token is for a macro which does not create data, such as `\hline`, `\toprule`, etc:

```
5215 \newbool{\LWR@exitingtabular}
5216
5217 \newcommand*{\LWR@tabledatacolumnntag}%
5218 {%
5219 \LWR@traceinfo{\LWR@tabledatacolumnntag}%
```

`\show\LWR@mynexttoken` to see what tokens to look for

If not any of the below, start a new table cell:

```
5220 \let\mynext\LWR@tabledatasinglecolumnntag%
```

If exiting the tabular:

```
5221 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\end}}{%
5222 {\booltrue\LWR@exitingtabular}}{ }%
```

`longtable` can have a caption in a cell

```
5223 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\caption}}{%
5224 {\let\mynext\LWR@donothing}}{ }%
```

Look for other things which would not start a table cell:

```
5225 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multicolumn}}{%
5226 {\let\mynext\LWR@donothing}}{ }%
5227 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multirow}}{%
5228 {\let\mynext\LWR@donothing}}{ }%
```

if come to an `\mrowcell`, this is a cell to be skipped over

```
5229 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\mrowcell}}{%
5230 {\let\mynext\LWR@donothing}}{ }%
5231 %
5232 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\hline}}{%
5233 {\let\mynext\LWR@donothing}}{ }%
5234 %
5235 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\toprule}}{%
5236 {\let\mynext\LWR@donothing}}{ }%
5237 %
5238 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\midrule}}{%
5239 {\let\mynext\LWR@donothing}}{ }%
5240 %
5241 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cmidrule}}{%
5242 {\let\mynext\LWR@donothing}}{ }%
5243 %
5244 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\specialrule}}{%
5245 {\let\mynext\LWR@donothing}}{ }%
5246 %
5247 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cline}}{%
5248 {\let\mynext\LWR@donothing}}{ }%
5249 %
5250 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\bottomrule}}{%
5251 {\let\mynext\LWR@donothing}}{ }%
5252 %
5253 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpprintonly}}{%
5254 {\let\mynext\LWR@donothing}}{ }%
```



```

5255 %
5256 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpHTMLonly}}
5257 {\let\mynext\LWR@donothing}{}%

```

no action for an `\end` token

Add similar to the above for any other non-data tokens which might appear in the table.

Start the new table cell if was not any of the above:


```

5258 \mynext%
5259 }

5260 \end{warpHTML}

```

## 56.19 `\mrowcell`

`\mrowcell` The user must insert `\mrowcell` into any multirow cells which must be skipped.  
 This command has no action during print output.

**for HTML & PRINT:**

```

5261 \begin{warpall}
5262 \newcommand*{\mrowcell}{}
5263 \end{warpall}

```

## 56.20 New `\tabular` definition

**for HTML output:**

```

5264 \begin{warpHTML}

```

These are default definitions in case `booktabs` is not loaded, and are not expected to used, but must exist as placeholders.

```

5265 \newcommand*{\LWR@origtoprule}[1]{\hline}
5266 \newcommand*{\LWR@origmidrule}[1]{\hline}
5267 \LetLtxMacro\LWR@origcmidrule\cline
5268 \newcommand*{\LWR@origbottomrule}[1]{\hline}
5269 \newcommand*{\LWR@origaddlinespace}[1][1]{}
5270 \newcommand*{\LWR@origmorecmidrules}{}
5271 \newcommand*{\LWR@origspecialrule}[3]{\hline}

```

Env `LWR@tabular` [*vertical position*] [*colspecs*]

The new tabular environment will be `\let` in `\LWR@LwarpStart`, since `siunitx` might redefine `tabular` in the user's document.

```

5272 \StartDefiningTabulars
5273
5274 \newenvironment*{LWR@tabular}[2] []
5275 {%
5276 \LWR@traceinfo{LWR@tabular started}%
5277 \addtocounter{LWR@tabulardepth}{1}%

```

Not yet started a table row:

```
5278 \global\boolfalse{LWR@startedrow}%
```

Not yet doing an hline:

```
5279 \global\boolfalse{LWR@doinghline}%
```

Not yet doing a top/bottom rule:

```
5280 \global\boolfalse{LWR@doingtbrule}%
```

For babel-french:

```
5281 \LWR@FBcancel%
```

Have not yet found the end of tabular command:

```
5282 \boolfalse{LWR@exitingtabular}%
```

Create the `table` tag:

```

5283 \global\booltrue{LWR@intabularmetadata}%
5284 \LWR@forcenewpage
5285 \LWR@htmlblocktag{table}%

```

Parse the table columns:

```
5286 \LWR@parsetablecols{#2}%
```

Table col spec is: `\LWR@tablecolspec` which is a string of `llccrr`, etc.

Do not place the table inside a paragraph:

```
5287 \LWR@stoppars%
```

Track column # for setting text-align:

```
5288 \setcounter{LWR@tablecolspos}{1}%
```

Start looking for midrules:

```
5289 \LWR@clearmidrules%
```

\\ becomes a macro to end the table row:

```
5290 \LetLtxMacro{\\}{\LWR@tabularendoffline}%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with `\LWR@getmynexttoken` to see if the next token might create a new data cell:

```
5291 \LWR@traceinfo{\LWR@tabular: redefining macros}%
5292 \renewcommand*{\hline}{\LWR@domidrule\LWR@getmynexttoken}%
5293
5294 \RenewDocumentCommand{\cline}{m}%
5295 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%
5296
5297 \DeclareDocumentCommand{\toprule}{o}{\LWR@dotbrule\LWR@getmynexttoken}%
5298
5299 \DeclareDocumentCommand{\midrule}{o}{\LWR@domidrule\LWR@getmynexttoken}%
5300
5301 \DeclareDocumentCommand{\cmidrule}{o d() m}%
5302 {\LWR@docmidrule[##1](##2){##3}\LWR@getmynexttoken}%
5303
5304 \DeclareDocumentCommand{\bottomrule}{o}{\LWR@dotbrule\LWR@getmynexttoken}%
5305 \DeclareDocumentCommand{\addlinespace}{o}{\LWR@getmynexttoken}%
5306 \DeclareDocumentCommand{\morecmidrules}{\LWR@getmynexttoken}%
5307 \DeclareDocumentCommand{\specialrule}{m m m}{\LWR@domidrule\LWR@getmynexttoken}
```

The following create data cells and will have no more data in this cell, so we do not want to look ahead for a possible data cell, so do not want to use `\LWR@getmynexttoken`.

```
5308 \LetLtxMacro{\multicolumn}{\LWR@tabledatamulticolumntag}%
5309 \LetLtxMacro{\multirow}{\LWR@tabledatamultirowtag}%
5310 \renewcommand*{\mrowcell}{\global\booltrue{\LWR@skippingmrowcell}}%
5311 \LetLtxMacro{\caption}{\LWR@longtabledatacaptiontag}%
```

Reset for new processing:

```
5312 \global\boolfalse{\LWR@tableparcell}%
5313 \global\boolfalse{\LWR@skippingmrowcell}%
5314 \global\boolfalse{\LWR@skipatbang}%
```

Set & for its special meaning inside the tabular:

```
5315 \StartDefiningTabulars%
5316 \protected\gdef&{\LWR@tabularampersand}%
```

Look ahead for a possible table data cell:

```
5317 \LWR@traceinfo{LWR@tabular: about to LWR@getmynexttoken}%
5318 \LWR@getmynexttoken%
5319 }%
```

Ending the environment:

```
5320 {%
5321 \LWR@traceinfo{LWR@tabular ending}%
5322 \LWR@closetabledatacell%
5323 \LWR@htmlblocktag{/tr}%
5324 \LWR@htmlblocktag{/table}%
5325 \global\boolfalse{LWR@intabularmetadata}%
```

Unnest one level of tabular:

```
5326 \addtocounter{LWR@tabulardepth}{-1}%
```

Restore `&` to its usual meaning:

```
5327 \protected\gdef&{\LWR@origampmacro}%
5328 \EndDefiningTabulars%
5329 \LWR@traceinfo{LWR@tabular finished ending}%
5330 }
5331
5332 \EndDefiningTabulars

5333 \end{warpHTML}
```

## 56.21 Array

Pkg `array`

`array` is also automatically loaded by `siunitx`.

## 57 Cross-references

Sectioning commands have been emulated from scratch, so the cross-referencing commands are custom-written for them. Emulating both avoids several layers of patches.

The `zref` package is used to remember section name, file, and `lateximage` depth and number for each label.

Table 9 shows the data structures related to cross-referencing.

for HTML output: 5334 \begin{warpHTML}

## 57.1 Setup

\@currentlabelname To remember the most recently defined section name, description, or caption, for \nameref.

```
5335 \newcommand*{\@currentlabelname}{}
```

\LWR@stripperperiod  $\{ \langle text \rangle \}$  [ $\langle . \rangle$ ]

Removes a trailing period.

```
5336 \def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%
```

\LWR@setlatestname  $\{ \langle object name \rangle \}$

Removes \label, strips any final period, and remembers the result.

```
5337 \newcommand*{\LWR@setlatestname}[1]{%
```

Remove \label and other commands from the name, the strip any final period.  
See zref-titleref and gettitlestring.

```
5338 \GetTitleStringExpand{#1}%
5339 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
5340 \edef\@currentlabelname{%
5341 \expandafter\LWR@stripperperiod\@currentlabelname%
5342 \ltx@empty.\ltx@empty\@nil%
5343 }%
5344 }
```

## 57.2 Zref setup

See:

[http://tex.stackexchange.com/questions/57194/  
extract-section-number-from-equation-reference](http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference)

Create a new property list called special:

```
5345 \zref@newlist{special}
```

Table 9: Cross-referencing data structures

---

<b>Original L<sup>A</sup>T<sub>E</sub>X:</b>	(print and HTML)
<b>\refstepcounter:</b> Steps the counter and sets \currentlabel.	
<b>\currentlabel:</b> \p@<ctr>\the<ctr> Updated by \refstepcounter.	
<b>\label:</b> Writes to the .aux file: \newlabel{<label>}{\currentlabel}{\thepage}}	
<b>\newlabel:</b> When the .aux file is read, sets \r@<label>.	
<b>\r@&lt;label&gt;:</b> Set to: {\currentlabel}{\thepage}}	
<b>\ref:</b> Returns the first part of \r@<label>.	
<b>\pageref:</b> Returns the second part of \r@<label>.	
<b>Added by lwarp:</b>	(HTML only)
<b>\label:</b> Adds HTML tags (section 57.3), plus \slabel data (section 57.2): <b>zLWR@name:</b> The section name for this label. <b>zLWR@htmlfilenumber:</b> The filenumber or name for this label. <b>zLWR@lateximagedepth:</b> The lateximagedepth for this label. <b>zLWR@lateximagenumber:</b> The lateximagenumber for this label.	
<b>\nameref:</b> Emulated from hyperref for lwarp. See section 57.4.	
<b>\ref and \nameref:</b> Adds HTML tags. See section 57.4.	
<b>Added by amsmath:</b>	(print and HTML)
<b>\label:</b> Execution is delayed until the math environment is completed.	
<b>\ltx@label:</b> L <sup>A</sup> T <sub>E</sub> X \label, (HTML: patched by lwarp,) later patched by cleveref.	
<b>Added by cleveref:</b>	(print and HTML)
<b>\refstepcounter:</b> Added: sets \cref@currentlabel.	
<b>\cref@currentlabel:</b> (<type>=<ctr> unless an alias is used): [<type>][\arabic{<ctr>}][<parent ctrs>]{\p@<ctr>\the<ctr>} Also see section 43.4 for use with footnotes.	
<b>\label:</b> Writes to the .aux file: \newlabel{<label>\cref}{\cref@currentlabel}{\thepage}}	
<b>\newlabel:</b> (Unchanged.) When the .aux file is read, sets \r@<label>\cref.	
<b>\r@&lt;label&gt;\cref:</b> Set to: {\cref@currentlabel}{\thepage}}	
<b>Utility functions:</b> See \cref@getlabel, \cref@gettype, \cref@getcounter, \cref@getprefix.	
<b>Cross-referencing names:</b> \crefname and \Crefname assign human-readable names for references to this counter type.	
<b>Additionally patched by lwarp:</b>	(HTML only)
<b>\cref, etc.:</b> Modified for lwarp. See section 67.	
<b>\label inside math:</b> See section 61.4.1.	
<b>Footnotes:</b> See \noteentry in section 43.4.	

---

Define a new property which has the name of the most recently declared section:

```
5346 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
5347 \zref@newprop{zLWR@htmlfilenumber}{%
5348 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilenumber}}%
5349 }%
```

Additional properties for lateximages:

```
5350 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
5351 \zref@newprop{zLWR@lateximagenumber}{\arabic{LWR@lateximagenumber}}
```

`zLWR@htmlfilenumber` property holds the file number or name

Add a `LWR@htmlfilenumber` property, and lateximage properties to special:

```
5352 \zref@addprop{special}{zLWR@name}
5353 \zref@addprop{special}{zLWR@htmlfilenumber}
5354 \zref@addprop{special}{zLWR@lateximagedepth}
5355 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
5356 \newcommand*{\LWR@spref}[2]{%
5357 \zref@extractdefault{#1}{#2}{??}}
```

`\LWR@nameref`  $\{\langle label \rangle\}$  Returns the section name for this label:

```
5358 \newcommand*{\LWR@nameref}[1]{%
5359 \LWR@spref{#1}{zLWR@name}}%
5360 }
```

`\LWR@htmlfileref`  $\{\langle label \rangle\}$  Returns the file number for this label:

```
5361 \newcommand*{\LWR@htmlfileref}[1]{%
5362 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
5363 \LWR@spref{#1}{zLWR@htmlfilenumber}}%
5364 }
```

`\LWR@lateximagedepthref`  $\{\langle label \rangle\}$  Returns the lateximagedepth for this label:

```
5365 \newcommand*{\LWR@lateximagedepthref}[1]{%
5366 \LWR@spref{#1}{zLWR@lateximagedepth}}%
5367 }
```

`\LWR@lateximagenumberref`  $\{\langle label \rangle\}$  Returns the `lateximagenumber` for this label:

```
5368 \newcommand*\LWR@lateximagenumberref}[1]{%
5369 \LWR@spref{#1}{\LWR@lateximagenumber}%
5370 }
```

`\LWR@splabel`  $\{\langle label \rangle\}$  Sanitize the name and then creates the label:

```
5371 \newcommand*\LWR@splabel}[1]{%
5372 \LWR@setlatestname{\@currentlabelname}%
5373 \zref@labelbylist{#1}{special}}
```

### 57.3 Labels

`\LWR@subsublabel`  $\{\langle label \rangle\}$  Creates an HTML id tag.

```
5374 \newcommand*\LWR@subsublabel}[1]{%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```
5375 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}{>}{0}}{%
5376 {}%
5377 {% not lateximage
```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```
5378 \ifbool{\LWR@doingstartpars}%
5379 {% pars allowed
5380 \ifbool{\LWR@doingapar}
5381 {% par started
5382 \LWR@htmltag{a id="#1"}{\LWR@htmltag{/a}%
5383 }% par started
5384 {% par not started
5385 \LWR@stoppars%
5386 \LWR@htmltag{a id="#1"}{\LWR@htmltag{/a}%
5387 \LWR@startpars%
5388 }% par not started
5389 }% pars allowed
5390 {% pars not allowed
5391 \LWR@htmltag{a id="#1"}{\LWR@htmltag{/a}%
5392 }% pars not allowed
5393 }% not lateximage
5394 }
```

`\LWR@newlabel`  $\{\langle label \rangle\}$  [ $\langle type \rangle$ ]



`\label` during HTML output when not in math mode, removing extra spaces around the label, as done by regular  $\text{\LaTeX}$  `\label`.

`cleveref` later encases this to add its own cross-referencing.

The optional  $\langle type \rangle$  is per the `ntheorem` package, and is ignored.

```
5395 \NewDocumentCommand{\LWR@newlabel}{m o}{%
5396 \LWR@traceinfo{\LWR@newlabel: starting}%
5397 \LWR@traceinfo{\LWR@newlabel: !#1!}%
5398 % \@bsphack%
```

Create a traditional LaTeX label, as modified by `cleveref`:

```
5399 \LWR@origlabel{#1}%
```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```
5400 \LWR@traceinfo{\LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
5401 \LWR@traceinfo{\LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
5402 \LWR@traceinfo{\LWR@newlabel: LWR@htmlfilenumber is \theLWR@htmlfilenumber}%
5403 \LWR@splabel{#1}%
5404 \LWR@subsublabel{#1}%
5405 % \@esphack%
5406 \LWR@traceinfo{\LWR@newlabel: done}%
5407 }
```

## 57.4 References

`\LWR@startref`  $\{\langle label \rangle\}$  (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, `#` character, and a label.

```
5408 \newcommand*{\LWR@startref}[1]
5409 {%
5410 \edef\LWR@lhref{\LWR@lateximagedepthref{#1}}%
5411 \LWR@traceinfo{\LWR@startref A: !#1!}%
```

Create the filename part of the link:

```
5412 \LWR@htmltag{a href="%
5413 \LWR@traceinfo{\LWR@startref B}%
5414 \LWR@htmlrefsectionfilename{#1}%
5415 \LWR@traceinfo{\LWR@startref C}%
5416 \#%
```

Create the destination id:

See if LWR@lateximagedepth is unknown:

```
5417 \LWR@traceinfo{LWR@startref D: !#1!}%
5418 \ifthenelse{\equal{\LWR@lidref}{??}}%
```

“??” if LWR@lateximagedepth is unknown, so create a link with an unknown destination:

```
5419 {%
5420     \LWR@traceinfo{LWR@startref D0: ??}%
5421     ??%
5422 }%
```

If LWR@lateximagedepth is known. Use a lateximage if the depth is greater than zero, or a regular link otherwise:

```
5423 {%
5424     \LWR@traceinfo{LWR@startref D1: \LWR@lidref}%
5425     \ifthenelse{\cinttest{\LWR@lidref}{>}{0}}%
5426     {%
5427         \LWR@traceinfo{LWR@startref D2: \LWR@lidref}%
5428         lateximage\LWR@lateximagenumberref{#1}%
5429     }%
5430     {%
5431         \LWR@traceinfo{LWR@startref D3}%
5432         #1%
5433     }%
5434 }%
5435 \LWR@traceinfo{LWR@startref E}%
```

Closing quote:

```
5436 "{}}%
5437 \LWR@traceinfo{LWR@startref F}%
5438 }
```

**\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)”.

```
5439 \NewDocumentCommand{\LWR@subnewref}{m m}{%
5440 \LWR@traceinfo{LWR@subnewref #1 #2}%
5441 \LWR@startref{#1}%
5442 \LWR@origref{#2}%
5443 \LWR@htmltag{/a}%
5444 }
```

```

\ref * {\label} \ref is \let to \LWR@newref

\LWR@newref * {\label} Create an internal document reference link, or without a link if
starred per hyperref.

5445 \NewDocumentCommand{\LWR@newref}{s m}{%
5446 \LWR@traceinfo{\LWR@newref #2}%
5447 \IfBooleanTF{#1}%
5448 {\LWR@origref{#2}}%
5449 {\LWR@subnewref{#2}{#2}}%
5450 }

\pagerefPageFor Text for starred page references.

5451 \newcommand*{\pagerefPageFor}{see }

\pageref * {\label} Create an internal document reference, or just the unlinked number
if starred, per hyperref.

5452 \NewDocumentCommand{\LWR@newpageref}{s m}{%
5453 \IfBooleanTF{#1}%
5454 {(\pagerefPageFor\LWR@origref{#2})}%
5455 {(\cpageref{#2})}%
5456 }

\nameref {\label}

5457 \newcommand*{\nameref}[1]{%
5458 \LWR@traceinfo{\nameref A}%
5459 \LWR@startref{#1}%
5460 \LWR@traceinfo{\nameref B}%
5461 \LWR@nameref{#1}%
5462 \LWR@traceinfo{\nameref C}%
5463 \LWR@htmltag{/a}%
5464 \LWR@traceinfo{\nameref D}%
5465 }

\Nameref {\label} In print, adds the page number. In HTML, does not.

5466 \let\Nameref\nameref

```

## 57.5 Hyper-references



Note that the code currently only sanitizes the underscore character. Additional

characters should be rendered inert as well. See the `hyperref.sty` definition of `\gdef\hyper@normalise` for an example.

Pkg `hyperref`

⚠ Do not tell other packages that `hyperref` is emulated. Some packages patch various commands if `hyperref` is present, which will probably break something, and the emulation already handles whatever may be emulated anyhow.

⚠ Any reference to `\usepackage{hyperref}` must be placed inside a `warpprint` environment.

```
5467 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
5468 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Create a link with a text name:

`\LWR@subhyperref`  $\{\langle URL \rangle\}$   $\{\langle text \rangle\}$

```
5469 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
5470 \LWR@htmltag{a href="#1" target="_{blank}"\LWR@orignewline}#2\LWR@htmltag{/a}%
5471 \LWR@ensuredoingapar%
5472 }
```

`\LWR@subhyperrefclass`  $\{\langle URL \rangle\}$   $\{\langle text \rangle\}$   $\{\langle htmlclass \rangle\}$

```
5473 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
5474 \LWR@htmltag{a href="#1"
5475 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
5476 \LWR@ensuredoingapar%
5477 }
```

`\href`  $[\langle options \rangle]$   $\{\langle URL \rangle\}$   $\{\langle text \rangle\}$

Create a link with accompanying text:

```
5478 \NewDocumentCommand{\LWR@hrefb}{0{} m +m}{%
5479 \LWR@subhyperref{#2}{#3}%
5480 \endgroup%
5481 \LWR@ensuredoingapar%
5482 }
5483
5484 \newcommand{\href}{%
5485 \LWR@ensuredoingapar%
5486 \begingroup%
5487 \catcode'\_ =12
5488 \LWR@hrefb%
5489 }
```

`\nolinkurl`  $\{\langle URL \rangle\}$

Print the name of the link without creating the link:

```
5490 \newcommand*{\LWR@nolinkurl}[1]{#1\endgroup\LWR@ensuredoingapar}
5491
5492 \newcommand{\nolinkurl}{%
5493 \LWR@ensuredoingapar%
5494 \begingroup\catcode'\_ =12
5495 \LWR@nolinkurl%
5496 }
```

`\url`  $\{\langle URL \rangle\}$

Create a link whose text name is the address of the link. The url package may redefine `\url`, so it is `\let` to `\LWR@urlahere` and also redefined by `lwarp-url`.

```
5497 \newcommand*{\LWR@urlb}[1]{%
5498 \href{#1}{#1}%
5499 \endgroup%
5500 \LWR@ensuredoingapar%
5501 }
5502
5503 \newcommand{\LWR@urla}{%
5504 \LWR@ensuredoingapar%
5505 \begingroup\catcode'\_ =12
5506 \LWR@urlb%
5507 }
5508
5509 \let\url\LWR@urla
```

`\LWR@subinlineimage`  $[\langle alttag \rangle] \{\langle class \rangle\} \{\langle filename \rangle\} \{\langle extension \rangle\} \{\langle style \rangle\}$

```
5510 \newcommand*{\LWR@subinlineimage}[5][ ]{%
5511 \ifthenelse{\equal{#1}{}}{%
5512 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"{}}}%
5513 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{}}}%
5514 }
5515 \end{warpHTML}
```

Table 10: Float data structures

---

For each `<type>` of float (figure, table, etc.) there exists the following:

---

**counter `<type>`:** A counter called `<type>`, such as `figure`, `table`.

**`\<type>name`:** Name. `\figurename` prints “Figure”, etc.

**`\ext@<type>`:** File extension. `\ext@figure` prints “lof”, etc.

**`\fps@<type>`:** Placement.

**`\the<type>`:** Number. `\thetable` prints the number of the table, etc.

**`\p@<type>`:** Parent’s number. Prints the number of the [within] figure, etc.

**`\fnum@<type>`:** Prints the figure number for the caption.

`\<type>name \the<type>`, “Figure 123”.

**`\<type>`:** Starts the float environment. `\figure` or `\begin{figure}`

**`\end<type>`:** Ends the float environment. `\endfigure` or `\end{figure}`

**`\tf@<ext>`:** The L<sup>A</sup>T<sub>E</sub>X file identifier for the output file.

**`LWR@have<type>`:** A boolean remembering whether a `\listof` was requested for a float of this type.

**File with extension `lo<f,t,a-z>`:** An output file containing the commands to build the `\listof<type><name>` “table-of-contents” structure.

**Cross-referencing names:** For `cleveref`’s `\cref` and related, `\crefname` and `\Crefname` assign human-readable names for references to this float type.

---

## 58 Floats

Floats are supported, although partially through emulation.

Table 10 shows the data structure associated with each `<type>` of float.

`\@makecaption` is redefined to print the float number and caption text, separated by `\CaptionSeparator`, which works with the `babel` package to adjust the caption separator according to the language. French, for example, uses an en-dash instead of a colon: “Figure 123 – Caption text”.

## 58.1 Float captions

for HTML output: 5516 \begin{warpHTML}

\LWR@floatbegin {<type>} [<placement>]

Begins a \newfloat environment.

```
5517 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
5518 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
5519
5520 === #1 begin
5521
5522 }{}%
5523 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
5524 \addtocounter{LWR@thisfloat}{1}%
5525 \booltrue{LWR@freezethisfloat}%
```

```
5526 \begingroup
```

Settings while inside the environment:

```
5527 \LWR@origraggedright
```

Open an HTML figure tag:

```
5528 \LWR@htmltag{figure id="autofloat-\arabic{LWR@thisfloat}" class="#1"}
```

```
5529 \renewcommand*{\@capttype}{#1}
5530 \caption@settype{#1}
5531 \LWR@startpars
5532 }
```

\@float Support packages which create floats directly.  
 \@dblfloat

```
5533 \let\@float\LWR@floatbegin
5534 \let\@dblfloat\LWR@floatbegin
```

\LWR@floatend Ends a \newfloat environment.

```
5535 \newcommand*{\LWR@floatend}{%
5536 \LWR@stoppars%
5537 \LWR@htmllementend{figure}%
```

```

5538 \endgroup%
5539 \boolfalse{LWR@freezethisfloat}%
5540 \LWR@startpars%
5541 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
5542
5543 === end
5544
5545 }{}%
5546 }

```

`\end@float` Support packages which create floats directly.  
`\end@dblfloat`

```

5547 \let\end@float\LWR@floatend
5548 \let\end@dblfloat\LWR@floatend

```

**Ctr** `LWR@thisfloat` A sequential counter for all floats and theorems. This is used to identify the float or theorem then reference it from the List of Figures and List of Tables.

```

5549 \newcounter{LWR@thisfloat}

```

**Bool** `LWR@freezethisfloat` Prevents multiple increments of `\LWR@thisfloat` inside a float.

```

5550 \newbool{LWR@freezethisfloat}
5551 \boolfalse{LWR@freezethisfloat}

```

`\LWR@maybeinthisfloat`

```

5552 \newcommand*{\LWR@maybeinthisfloat}{%
5553 \ifbool{LWR@freezethisfloat}{\addtocounter{LWR@thisfloat}{1}}%
5554 }

```

`\@capttype` Remembers which float type is in use.

```

5555 \newcommand*{\@capttype}{}

```

### 58.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```

5556 \AtBeginDocument{\providecommand*\CaptionSeparator{:-}}

```

`\@makecaption`  $\{\langle name \text{ and } num \rangle\} \{\langle text \rangle\}$

Prints the float type and number, the caption separator, and the caption text.

```

5557 \AtBeginDocument{\renewcommand{\@makecaption}[2]{\#1\CaptionSeparator\#2}}

```



### 58.1.2 Caption and LOF linking and tracking

When a new HTML file is marked in the  $\text{\LaTeX}$  PDF file, the  $\text{\LaTeX}$  page number at that point is stored in `LWR@latestautopage`, (and the associated filename is remembered by the special  $\text{\LaTeX}$  labels). This page number is used to generate an `autofloat` HTML `<id>` in the HTML output at the start of the new HTML file. Meanwhile, there is a float counter used to generate an HTML `autofloat <id>` at the start of the float itself in the HTML file. The `autopage` and `autofloat` values to use for each float are written to the `.lof`, etc. files just before each float's entry. These values are used by `\l@figure`, etc. to create the HTML links in the List of Figures, etc.

Ctr `LWR@nextautofloat` Tracks autofloat for floats. Tracks autopage for floats.

Ctr `LWR@nextautopage` These are updated per float as the `.lof` file is read.

```
5558 \newcounter{LWR@nextautofloat}
5559 \newcounter{LWR@nextautopage}
```

```
\LWRsetnextfloat {<autopage>} {<autofloat>}
```

This is written to the `.lof` file just before each float's usual entry. The `autopage` and `autofloat` are remembered for `\l@figure` to use when creating the HTML links.

```
5560 \newcommand*{\LWRsetnextfloat}[2]{%
5561 \setcounter{LWR@nextautopage}{#1}%
5562 \setcounter{LWR@nextautofloat}{#2}%
5563 }
```

Ctr `LWR@latestautopage` Updated each time a new HTML file is begun. `\LWRsetnextfloat` is written with this and the `autofloat` by the modified `\addcontentsline` just before each float's entry.

```
5564 \newcounter{LWR@latestautopage}
5565 \setcounter{LWR@latestautopage}{1}

5566 \let\LWR@origcaption@begin\caption@begin
5567 \let\LWR@origcaption@end\caption@end
5568 \let\LWR@orig@@par\@@par
```

`\LWR@caption@begin` Low-level patches to create HTML tags for captions.

```
5569 \newcommand{\LWR@caption@begin}
5570 {
5571 \LWR@traceinfo{LWR@caption@begin}%
```

Keep par and minipage changes local:

```
5572 \begingroup%
```

The caption code was not allowing the closing par tag:

```
5573 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a minipage or \parbox inside the caption:

```
5574 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{-}{-}%
5575 \RenewDocumentCommand{\parbox}{0{t} o 0{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
5576 \LWR@htmlblocktag{figcaption}%
5577 \LWR@origcaption@begin%
5578 }
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
5579 \newcommand{\LWR@caption@end}
5580 {%
5581 \LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
5582 \LWR@origvspace*{\baselineskip}%
```

Closing tag:

```
5583 \LWR@htmlblocktag{/figcaption}%
5584 \endgroup%
5585 % \leavevmode% avoid bad space factor (0) error
5586 \LWR@traceinfo{\LWR@caption@end: done}%
5587 }
```

`\caption@begin` Low-level patches to create HTML tags for captions.

```
\caption@end
5588 \AtBeginDocument{
5589 \let\caption@begin\LWR@caption@begin
5590 \let\caption@end\LWR@caption@end
5591 }
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```

5592 \let\LWR@origcaptionlistentry\captionlistentry
5593
5594 \renewcommand*{\captionlistentry}{%
5595 \LWR@maybeinthisfloat%
5596 \LWR@ensuredoingapar%
5597 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5598 \LWR@origcaptionlistentry%
5599 }
5600
5601 \def\LWR@LTcaptionlistentry{%
5602 \LWR@ensuredoingapar%
5603 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5604 \bgroup
5605 \@ifstar{\egroup\LWR@LTcaptionlistentry}% gobble *
5606 {\egroup\LWR@LTcaptionlistentry}}%
5607 \def\LWR@LTcaptionlistentry#1{%
5608 \caption@listentry\@firstoftwo[LTcaptype]{#1}}%

```

`\addcontentsline` Patched to write the autpage and autofloat before each float's entry. No changes if writing .toc For a theorem, automatically defines `\ext@<type>` as needed, to mimic and reuse the float mechanism.

```

5609 \let\LWR@origaddcontentsline\addcontentsline
5610
5611 \renewcommand*{\addcontentsline}[3]{%
5612 \ifthenelse{\equal{#1}{toc}}{ }{%
5613 \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{ }
5614 \addtocontents{\@nameuse{ext@#2}}{%
5615 \protect\LWRsetnextfloat%
5616 {\arabic{LWR@latestautopage}}%
5617 {\arabic{LWR@thisfloat}}%
5618 }% addtocontents
5619 }% not toc
5620 \LWR@origaddcontentsline{#1}{#2}{#3}%
5621 }

```

`\captionof` Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

5622 \AtBeginDocument{
5623 \let\LWR@origcaptionof\captionof
5624
5625 \renewcommand*{\captionof}{%
5626 \LWR@maybeinthisfloat%
5627 \LWR@stoppars
5628 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5629 \LWR@origcaptionof%
5630 }

```

```
5631 }
5632 \end{warpHTML}
```

## 59 Table of Contents, LOF, LOT

This section controls the generation of the TOC, LOF, LOT.

The `.toc`, `.lof`, and `.lot` files are named by the source code `\jobname`.

In HTML, the printed tables are placed inside a div of class `.toc`, `.lof`, or `.lot`.

A “sidetoc” is provided which prints a subset of the TOC on the side of each page other than the homepage.

The regular  $\text{\LaTeX}$  infrastructure is used for TOC, along with some patches to generate HTML output.

for HTML output: 5633 `\begin{warpHTML}`

### 59.1 Reading and printing the TOC

```
\LWR@myshorttoc {\toc/lof/lot}
```

Reads in and prints the TOC/LOF/LOT at the current position. While doing so, makes the `@` character into a normal letter to allow formatting commands in the section names.

Unlike in regular  $\text{\LaTeX}$ , the file is not reset after being read, since the TOC may be referred to again in each HTML page, and is used for the sideTOC.

```
5634 \newcommand*\LWR@myshorttoc}[1]{
5635 \LWR@ensuredoingapar
```

Only if the file exists:

```
5636 \IfFileExists{\jobname.#1}{
```



Make `@` a regular letter. Many of the commands in the file will have `@` characters in them, so `@` must be made a regular letter.



For `pdf $\text{\LaTeX}$` , also change to `latin1` encoding. When reading back a file with accented characters, the encoding change seems to be required, rather than leaving it `utf8`.

```

5637 \begingroup
5638 % \ifxetexorluatex%
5639 % \else
5640 % \inputencoding{latin1}% currently disabled
5641 % \fi
5642 \makeatletter

```

Read in the TOC file:

```

5643 \@input{\jobname.#1}
5644 % \makeatother
5645 \endgroup
5646 }%
5647 {}%
5648 }

```

`\LWR@subtableofcontents`  $\{\langle toc/lof/lot \rangle\} \{\langle sectionstarname \rangle\}$

Places a TOC/LOF/LOT at the current position.

```

5649 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%

```

Closes previous levels:

```

5650 \@ifundefined{chapter}
5651 {\LWR@closeprevious{\LWR@depthsection}}
5652 {\LWR@closeprevious{\LWR@depthchapter}}

```

Prints any pending footnotes so that they appear above the potentially large TOC:

```

5653 \LWR@printpendingfootnotes

```

Place the list into its own chapter (if defined) or section:

```

5654 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}

```

Create a new HTML nav containing the TOC/LOF/LOT:

```

5655 \LWR@htmlclass{nav}{#1}

```

Create the actual list:

```

5656 \LWR@myshorttoc{#1}

```

Close the nav:

```

5657 \LWR@htmlclassend{nav}{#1}
5658 }

```

Patch `\@starttoc` to encapsulate the TOC inside HTML tags:

```
5659 \let\LWR@orig@starttoc\@starttoc
5660
5661 \renewcommand{\@starttoc}[1]{
5662 \LWR@htmlelementclass{nav}-{#1}
5663 \LWR@orig@starttoc{#1}
5664 \LWR@htmlelementclassend{nav}-{#1}
5665 }
```

Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.

```
5666 \let\LWR@origtableofcontents\tableofcontents
5667 \let\LWR@origlistoffigures\listoffigures
5668 \let\LWR@origlistoftables\listoftables
5669
5670 \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:

```
5671 \ifbool{FormatWordProcessor}{}{
```

Copy the `.toc` file to `.sidetoc` for printing the sideTOC. The original `.toc` file is renewed when `\tableofcontents` is finished.

```
5672 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
5673 \LWR@printpendingfootnotes
5674 \LWR@origtableofcontents
5675 }
5676 }
5677 \renewcommand*{\listoffigures}{
5678 \ifbool{FormatWordProcessor}{}{
5679 \LWR@printpendingfootnotes
5680 \LWR@origlistoffigures
5681 }
5682 }
5683
5684 \renewcommand*{\listoftables}{
5685 \ifbool{FormatWordProcessor}{}{
5686 \LWR@printpendingfootnotes
5687 \LWR@origlistoftables
5688 }
5689 }
```

## 59.2 High-level TOC commands

`\listof`  $\{\langle type \rangle\}$   $\{\langle title \rangle\}$

Emulate the `\listof` command from the `float` package (section 106). Used to create lists of custom float types. Also used to redefine the standard L<sup>A</sup>T<sub>E</sub>X `\listoffigures` and `\listoftables` commands.

```
5690 \NewDocumentCommand{\listof}{m +m}{%
5691 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
5692 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
5693 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
5694     \jobname.\csuse{ext@#1}\relax
5695 }
```

## 59.3 Side TOC

The “side TOC” is a table-of-contents positioned to the side.

It may be renamed by redefining `\sidetocname`, and may contain paragraphs.

CSS may be used to format the sideTOC:

*CSS related to sideTOC:*

---

**nav.sidetoc:** The entire sideTOC.

**div.sidetoctitle:** The title.

**div.sidetoccontents:** The table of contents.

---

```
5696 \end{warpHTML}
```

**for HTML & PRINT:** 5697 `\begin{warpall}`

**Ctrl SideTOCDepth** Controls how deep the side-TOC gets. Use a standard L<sup>A</sup>T<sub>E</sub>X section level similar to `tocdepth`.

```
5698 \newcounter{SideTOCDepth}
5699 \setcounter{SideTOCDepth}{1}
```

`\sidetocname` Holds the default name for the sideTOC.

```
5700 \newcommand{\sidetocname}{Contents}
```

```
5701 \end{warpall}
```

for HTML output: 5702 \begin{warpHTML}

\LWR@sidetoc Creates the actual side-TOC.

```
5703 \newcommand*\LWR@sidetoc){
5704 \LWR@forcenewpage
5705 \LWR@stoppars
5706
```

The entire sideTOC is placed into a nav of class `sidetoc`.

```
5707 \LWR@htmlelementclass{nav}{sidetoc}
5708
5709 \setcounter{tocdepth}{\value{SideTOCDepth}}
5710
```

The title is placed into a div of class `sidetoctitle`, and may contain paragraphs.

```
5711 \begin{BlockClass}{sidetoctitle}
5712 \sidetocname
5713 \end{BlockClass}
```

The table of contents is placed into a div of class `sidetoccontents`.

```
5714 \begin{BlockClass}{sidetoccontents}
5715 \LinkHome
5716
5717 \LWR@myshorttoc{sidetoc}
5718 \end{BlockClass}
5719 \LWR@htmlelementclassend{nav}{sidetoc}
5720 }
```

## 59.4 Low-level TOC line formatting

\numberline {*{number}*}

(Called from each line in the `.aux`, `.lof` files.)

Record this section number for further use:

```
5721 \renewcommand*\numberline}[1]{%
5722 \LWR@sectionnumber{#1}%
5723 }
```



`\hypertoc`  $\{\langle 1: depth \rangle\} \{\langle 2: type \rangle\} \{\langle 3: name \rangle\} \{\langle 4: page \rangle\}$

Called by `\l@section`, etc. to create a hyperlink to a section.

The autopage label is always created just after the section.

**#1** is depth

**#2** is section, subsection, etc.

**#3** the text of the caption

**#4** page number

```
5724 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
5725 \ifthenelse{\cnttest{#1}{<=}}{\value{tocdepth}}{%
```

```
5726 \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```
5727 \LWR@subhyperrefclass{%
```

```
5728 \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
```

```
5729 \LWR@stoppars%
```

```
5730 }
```

```
5731 {}
```

```
5732 }
```

Ctrl `lofdepth` TOC depth for figures.

```
5733 \newcounter{lofdepth}
```

```
5734 \setcounter{lofdepth}{1}
```

Ctrl `lotdepth` TOC depth for tables.

```
5735 \newcounter{lotdepth}
```

```
5736 \setcounter{lotdepth}{1}
```

`\hypertocfloat`  $\{\langle 1: depth \rangle\} \{\langle 2: type \rangle\} \{\langle 3: ext\ of\ parent \rangle\} \{\langle 4: caption \rangle\} \{\langle 5: page \rangle\}$

**#1** is depth

**#2** is figure, table, etc.

**#3** is lof, lot, of the parent.

#4 the text of the caption

#5 page number

```
5737 \newcommand{\hypertocfloat}[5]{%
5738 \LWR@startpars
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
5739 \@ifundefined{c@#3depth}{%
5740 \newcounter{#3depth}%
5741 \setcounter{#3depth}{1}%
5742 }{}%
```

Respond to `lofdepth`, etc.:

```
5743 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
5744 \ifthenelse{\cnttest{#1}{<=}}{\arabic{#3depth}}{}%
5745 \LWR@startpars%
```

Create an HTML link to `filename#autofloat-(float number)`, with text of the caption, of the given HTML class.

```
5746 \LWR@subhyperrefclass{%
5747 \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
5748 \#autofloat-\arabic{LWR@nextautofloat}}%
5749 {#4}{toc#2}%
5750 \LWR@stoppars%
5751 }{}%
5752 }
```

Automatically called by `\contentsline`:

```
5753 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
5754 \DeclareDocumentCommand{\l@chapter}{m m}
5755 {\hypertoc{0}{chapter}{#1}{#2}}
5756 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
5757 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
5758 \renewcommand{\l@subsubsection}[2]
5759 {\hypertoc{3}{subsubsection}{#1}{#2}}
5760 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
5761 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
5762 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
5763 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

5764 \end{warpHTML}
```

## 60 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: 5765 `\begin{warpHTML}`

```
5766 \newcounter{LWR@autoindex}
5767 \setcounter{LWR@autoindex}{0}
5768
5769 \newcounter{LWR@autoglossary}
5770 \setcounter{LWR@autoglossary}{0}
```

`\LWR@indexsection` Controls whether the index will be in a section or a chapter, depending on the documentclass.

```
5771 \@ifundefined{chapter}
5772   {\newcommand*{\LWR@indexsection}{\section{\indexname}}}
5773   {\newcommand*{\LWR@indexsection}{\chapter{\indexname}}}
```

`\printindex`

```
5774 \let\LWR@origprintindex\printindex
5775
5776 \renewcommand*{\printindex}
5777 {
5778   \LWR@indexsection
5779   \LWR@startpars
5780   \LWR@origprintindex
5781 }
```

Env `theindex`

```
5782 \renewenvironment*{theindex}{%
5783 \let\item\LWR@indexitem%
5784 \let\subitem\LWR@indexsubitem%
5785 \let\subsubitem\LWR@indexsubsubitem%
5786 }{}
```

`\LWR@indexitem`

```

5787 \newcommand{\LWR@indexitem}{
5788
5789 \InlineClass{indexitem}{}
5790 }

```

\LWR@indexitem

```

5791 \newcommand{\LWR@indexsubitem}{
5792
5793 \InlineClass{indexsubitem}{}
5794 }

```

\LWR@indexitem

```

5795 \newcommand{\LWR@indexsubsubitem}{
5796
5797 \InlineClass{indexsubsubitem}{}
5798 }

```

\@wrindex {*{term}*}      Redefined to write the LWR@latestautopage counter instead of page

```

5799 \def\@wrindex#1{%
5800 \addtocounter{LWR@autoindex}{1}%
5801 \LWR@newlabel{LWRindex-\theLWR@autoindex}%
5802 \protected@write\@indexfile{%
5803 {\string\indexentry{#1}{\theLWR@autoindex}}%
5804 \endgroup
5805 \@esphack}

```

\@wrglossary {*{term}*}      Redefined to write the LWR@latestautopage counter instead of page

```

5806 \def\@wrglossary#1{%
5807 \addtocounter{LWR@autoglossary}{1}%
5808 \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
5809 \protected@write\@glossaryfile{%
5810 {\string\glossaryentry{#1}{\theLWR@autoglossary}}%
5811 \endgroup
5812 \@esphack}

```

\hyperindexref {*{autosecnumber}*}

\hyperindexref{web address} is inserted into \*.ind by the xindy style file  
lwarp.xdy

```

5813 \newcommand*\hyperindexref[1]{\nameref{LWRindex-#1}}

```

```
5814 \end{warpHTML}
```


**for PRINT output:** A null command for print mode, in case `hyperref` was not used:

```
5815 \begin{warpprint}
5816 \newcommand{\hyperindexref}[1]{#1}
5817 \end{warpprint}
```

**for HTML & PRINT:** For the `glossaries` package, try to prevent an error where `\glo@name` was not found:

```
5818 \begin{warpall}
5819 \providecommand{\glo@name}{}
5820 \end{warpall}
```

## 61 Math

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

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

**SVG math option** For SVG math, math is rendered as usual by  $\text{\LaTeX}$  into the initial PDF file using the current font<sup>9</sup>, 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  $\text{\LaTeX}$  with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML `alt` attribute carries the  $\text{\LaTeX}$  code which generated the math, allowing copy/paste of the  $\text{\LaTeX}$  math expression into other documents.

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

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

**SVG inline** Another approach would be to in-line the SVG files directly into the HTML. This avoids having a large number of files and potentially speeds loading the images,

<sup>9</sup>See section 194 regarding fonts and fractions.

but dis-allows the possibility of sharing one file among many instances without user intervention.

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

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

**MathJax math option** The popular MathJax alternative ([mathjax.org](http://mathjax.org)) may be used to display math.

Prog **MathJax**

When MathJax is enabled, math is rendered twice:

1. As regular  $\text{\LaTeX}$  PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of  $\text{\LaTeX}$ , and
2. As detokenized printed  $\text{\LaTeX}$  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  $\text{\LaTeX}$  values, and the MathJax cross-referencing system is ignored in favor of the  $\text{\LaTeX}$  internal system, seamlessly integrating with the rest of the  $\text{\LaTeX}$  code.

**MathJax limitations** Limitations when using MathJax include:

Prog **MathJax**

**chapter numbers**

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

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

**subequations**

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

**footnotes in math**


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

**lateximage**

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

**siunitx**

- Usage of `siunitx` inside a math equation is supported via a third-party MathJax

 **siunitx inside an equation**

extension. While inside a math expression, do not use `\SI` or `\si` inside `\text`, where it will be rendered as normal text.

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

**L<sup>A</sup>T<sub>E</sub>X macros**

- MathJax does not automatically support custom L<sup>A</sup>T<sub>E</sub>X macros, but they may be set up by the user.

**custom MathJax macros**

As an example of using custom L<sup>A</sup>T<sub>E</sub>X macros with MathJax, place the following at the start of the document, after `\begin{document}`:

---

```
\begin{warpHTML} % Only for HTML output,
\ifbool{mathjax} % and only for MathJax output:
{
  \(\ % New macros for MathJax are placed inside a math expression:
  \newcommand{\expval}[1]{\langle#1\rangle}
  \newcommand{\abs}[1]{\lvert#1\rvert}
  \)
  \{}
\end{warpHTML}
```

---

**for HTML output:** 5821 `\begin{warpHTML}`

`\$` Plain dollar signs appearing in the HTML output may be interpreted by MathJax to be math shifts. For a plain text dollar `\$`, print it inside a span to avoid it being interpreted by MathJax, unless are inside a `lateximage`, in which case it will not be seen by MathJax.

```
5822 \let\LWR@origtextdollar\$
5823
5824 \renewcommand*{\$}{%
5825 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{
5826 {\LWR@origtextdollar}%
5827 {\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
5828 }
```

**Ctrl LWR@externalfilecnt** Counter for the external files which are generated and then referenced from the HTML:

```
5829 \newcounter{LWR@externalfilecnt}
```

## 61.1 Inline and display math

```

5830 \LetLtxMacro\LWR@origdollar$
5831 \LetLtxMacro\LWR@secondorigdollar$% balance for editor syntax highlighting

5832 \LetLtxMacro\LWR@origopenparen\(  

5833 \LetLtxMacro\LWR@origcloseparen\  

5834 \LetLtxMacro\LWR@origopenbracket\  

5835 \LetLtxMacro\LWR@origclosebracket\]

```

**\$** Redefine the dollar sign to place math inside a `lateximage`, or use MathJax:  
**\$\$**

```

5836 \begingroup
5837 \catcode'\$=\active%
5838 \protected\gdef$ {\ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

**\LWR@doubledollar** Redefine the double dollar sign to place math inside a `lateximage`, or use MathJax:

```

5839 \gdef\LWR@doubledollar$#1$${
5840 \ifbool{mathjax}%

```

For MathJax, print the math between `\[` and `\]`:

```

5841 {\textbackslash\LWR@HTMLsanitize{#1}\textbackslash}}

```

For SVG, print the math inside a `lateximage`, with an ALT tag of the  $\LaTeX$  code:

```

5842 {% not mathjax
5843
5844 \begin{lateximage}%
5845 [\textbackslash{[] \LWR@HTMLsanitize{#1} \textbackslash{[]}]%
5846 \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
5847 \end{lateximage}
5848
5849 }
5850 }%

```

**\LWR@singledollar** Redefine the single dollar sign to place math inside a `lateximage`, or use MathJax:

```

5851 \gdef\LWR@subsingledollar#1{%
5852 \ifbool{mathjax}%

```

For MathJax, print the math between `\(` and `\)`:

```

5853 {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}

```

For SVG, print the math inside a `lateximage`, with an ALT tag of the  $\LaTeX$  code:

```

5854 {% not mathjax
5855 \begin{lateximage}%

```



```

5856    [\textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)]%
5857    \LWR@origdollar#1\LWR@origdollar%
5858    \end{lateximage}%
5859 }%
5860 }
5861
5862 \gdef\LWR@singledollar#1${%
5863 \LWR@subsingledollar{#1}%
5864 }

```

\( Redefine to the above dollar macros.  
 \)

```

5865 \gdef\(#1\){$#1$}
5866 \gdef\[#1\]{$$#1$$}
5867
5868 \endgroup
5869
5870
5871

```

\@ensuredmath  $\{expression\}$  Not yet adapted to lwarp.

```

5872 \let\LWR@orig@ensuredmath\@ensuredmath
5873
5874 \newcommand*{\LWR@ensuredmath}[1]{%
5875 \LWR@traceinfo{\@ensuredmath: page \thepage: \detokenize{#1}}%
5876 \LWR@subsingledollar{#1}%
5877 }

```

Remove the old math and displaymath environments:

```

5878 \let\math\relax
5879 \let\endmath\relax
5880 \let\displaymath\relax
5881 \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`.

```

5882 \NewEnviron{math}{\expandafter\(\BODY\)}

```

Env **displaymath** Set math mode then typeset the body of what was between the begin/end. See the `environ` package for `\BODY`.

```

5883 \NewEnviron{displaymath}{\expandafter\[\BODY\]\@ignoretrue}

```

## 61.2 MathJax support

Ctr LWR@nextequation Used to add one to compute the next equation number.

```
5884 \newcounter{LWR@nextequation}
```

\LWR@syncmathjax Sets the MathJax equation format and number for the following equations.

These MathJax commands are printed inside “\(" and “\)” characters. They are printed to HTML output, not interpreted by L<sup>A</sup>T<sub>E</sub>X.

```
5885 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
5886 \ifcsdef{thechapter}{
5887 \InlineClass{hidden}{
5888 \textbackslash(
5889 \textbackslash{}seteqsection \{\thechapter\}
5890 \textbackslash)
5891 }
5892 }
5893 {}}% not using chapters
```

MathJax doesn't allow setting the equation number to 1:

```
5894 \ifthenelse{\cnttest{\value{equation}}>0}
5895 {
```

Tell MathJax that the next set of equations begins with the current L<sup>A</sup>T<sub>E</sub>X equation number, plus one.

```
5896 \setcounter{LWR@nextequation}{\value{equation}}
5897 \addtocounter{LWR@nextequation}{1}
```

Place the MathJax command inside “\(" and “\)” characters, to be printed to HTML, not interpreted by L<sup>A</sup>T<sub>E</sub>X.

```
5898 \InlineClass{hidden}{
5899 \textbackslash(
5900 \textbackslash{}seteqnumber \{\arabic{LWR@nextequation}\}
5901 \textbackslash)
5902 }
5903 {}}% not eq > 0
5904 }
```

`\LWR@restoreorigformatting` While producing SVG math or a `lateximage`, use regular L<sup>A</sup>T<sub>E</sub>X formatting instead of HTML tags.

```

5905 \newcommand*{\LWR@restoreorigformatting}{%
5906 \LWR@traceinfo{\LWR@restoreorigformatting}%
5907 \LetLtxMacro{\hspace}{\LWR@orighspace}%
5908 \LetLtxMacro\rule\LWR@origrule%
5909 \let\,\LWR@origcomma% disable HTML short unbreakable space
5910 \LetLtxMacro{\textrm}{\LWR@origtextrm}%
5911 \LetLtxMacro{\textsf}{\LWR@origtextsf}%
5912 \LetLtxMacro{\texttt}{\LWR@origtexttt}%
5913 \LetLtxMacro{\textbf}{\LWR@origtextbf}%
5914 \LetLtxMacro{\textmd}{\LWR@origtextmd}%
5915 \LetLtxMacro{\textit}{\LWR@origtextit}%
5916 \LetLtxMacro{\textsl}{\LWR@origtextsl}%
5917 \LetLtxMacro{\textsc}{\LWR@origtextsc}%
5918 \LetLtxMacro{\textup}{\LWR@origtextup}%
5919 \LetLtxMacro{\textnormal}{\LWR@origtextnormal}%
5920 \LetLtxMacro{\emph}{\LWR@origemph}%
5921 \LetLtxMacro{\rmfamily}{\LWR@origrmfamily}%
5922 \LetLtxMacro{\sffamily}{\LWR@origsffamily}%
5923 \LetLtxMacro{\ttfamily}{\LWR@origttfamily}%
5924 \LetLtxMacro{\bfseries}{\LWR@origbfseries}%
5925 \LetLtxMacro{\mdseries}{\LWR@origmdseries}%
5926 \LetLtxMacro{\upshape}{\LWR@origupshape}%
5927 \LetLtxMacro{\slshape}{\LWR@origslshape}%
5928 \LetLtxMacro{\scshape}{\LWR@origscshape}%
5929 \LetLtxMacro{\itshape}{\LWR@origitshape}%
5930 \LetLtxMacro{\em}{\LWR@origem}%
5931 \LetLtxMacro{\normalfont}{\LWR@orignormalfont}%
5932 \let\sp\LWR@origsp%
5933 \let\sb\LWR@origsb%
5934 \let\textsuperscript\LWR@origtextsuperscript%
5935 \let\textsubscript\LWR@origtextsubscript%
5936 \let\textdegree\LWR@origtextdegree%
5937 \let\textcelsius\LWR@origtextcelsius%
5938 \let\textohm\LWR@origtextohm%
5939 \let\textmu\LWR@origtextmu%
5940 \let~\LWR@origtilde%
5941 \let\enskip\LWR@origenskip%
5942 \let\quad\LWR@origquad%
5943 \let\qqquad\LWR@origqqquad%
5944 \LetLtxMacro{\tabular}{\LWR@origtabular}%
5945 \LetLtxMacro{\endtabular}{\LWR@origendtabular}%
5946 \LetLtxMacro\toprule\LWR@origtoprule%
5947 \LetLtxMacro\midrule\LWR@origmidrule%
5948 \LetLtxMacro\cmidrule\LWR@origcmidrule%
5949 \LetLtxMacro\bottomrule\LWR@origbottomrule%
5950 \LetLtxMacro\addlinespace\LWR@origaddlinespace%
5951 \LetLtxMacro\morecmidrules\LWR@origmorecmidrules%
```

```

5952 \LetLtxMacro\specialrule\LWR@origspecialrule%
5953 \let\newline\LWR@orignewline%
5954 \LetLtxMacro{\raisebox}{\LWR@origraisebox}%
5955 \LetLtxMacro{\scalebox}{\LWR@origscalebox}%
5956 \LetLtxMacro{\rotatebox}{\LWR@origrotatebox}%
5957 \let\reflectbox\LWR@origreflectbox%
5958 \let\framebox\LWR@origframebox%
5959 \let\makebox\LWR@origmakebox%
5960 \let\fbbox\LWRprint@fbbox%
5961 \let\fbboxBlock\LWRprint@fbbox%
5962 \LetLtxMacro{fminipage}{\LWRprint@fminipage}%
5963 \LetLtxMacro{endfminipage}{\endLWRprint@fminipage}%
5964 \LetLtxMacro{minipage}{\LWR@origminipage}%
5965 \let\endminipage\LWR@endminipage%
5966 \LetLtxMacro{\parbox}{\LWR@origparbox}%
5967 \LWR@restoreorigxcolor%
5968 \LWR@restoremoreorigxcolor%
5969 \LWR@FBCancel%
5970 }

```

`\LWR@hidelatexequation`  $\{ \langle environment \rangle \} \{ \langle contents \rangle \}$

Creates the L<sup>A</sup>T<sub>E</sub>X version of the equation inside an HTML comment.

```

5971 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%

```

Stop HTML paragraph handling and open an HTML comment:

```

5972 \LWR@stoppars
5973 \LWR@htmlopencomment
5974

```

Start the L<sup>A</sup>T<sub>E</sub>X math environment inside the HTML comment:

```

5975 \begingroup
5976 \csuse{\LWR@orig#1}

```

While in the math environment, restore various commands to their L<sup>A</sup>T<sub>E</sub>X meanings.

```

5977 \LWR@restoreorigformatting

```

See `\LWR@htmlmathlabel` in section [61.4.1](#).

Print the contents of the equation:

```

5978 #2

```

End the L<sup>A</sup>T<sub>E</sub>X math environment inside the HTML comment:

```

5979 \csuse{LWR@origend#1}
5980 \endgroup
5981

```

Close the HTML comment and resume HTML paragraph handling:

```

5982 \LWR@htmlclosecomment
5983 \LWR@startpars
5984 }

```

```
\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<sup>A</sup>T<sub>E</sub>X.

```
5985 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
```

Enclose the MathJax environment inside printed “\ (“ and “\)” characters.

```

5986 \LWR@origtilde\LWR@orignewline
5987 \textbackslash{}begin\{#1\}

```

Print the contents, sanitizing for HTML special characters.

```
5988 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```

5989 \textbackslash{}end\{#1\}
5990 \LWR@orignewline
5991 }

```

## 61.3 Equation environment

Remember existing equation environment:

```

5992 \let\LWR@origequation\equation
5993 \let\LWR@origendequation\endequation

```

Remove existing equation environment:

```

5994 \let\equation\relax
5995 \let\endequation\relax

```

**Env** **equation** The new **equation** environment is created with `\NewEnviron` (from the `environ` package), which stores the contents of its environment in a macro called `\BODY`.

For SVG math output, the contents are typeset using the original `equation` inside a `lateximage`, along with an ALT tag containing a detokenized copy of the  $\LaTeX$  source for the math.

For MathJax output, the contents are typeset in an original `equation` environment placed inside a HTML comment, with special processing for `\labels`. The contents are also printed to the HTML output for processing by the MathJax script.

```
5996 \NewEnviron{equation}{%
5997
5998 \ifbool{mathjax}
```

MathJax output:

```
5999 {
```

Print commands to synchronize MathJax's equation number and format to the current  $\LaTeX$  chapter/section and equation number:

```
6000 \LWR@syncmathjax
```

Print the  $\LaTeX$  math inside an HTML comment:

```
6001 \LWR@hidelatexequation{equation}{\BODY}
6002 }
```

SVG output: Create the `lateximage` along with an HTML ALT tag having an equation number, the  $\LaTeX$  equation environment commands, and the contents of the environment's `\BODY`.

```
6003 {% not mathjax
```

Begin the `lateximage` with an ALT tag containing the math source:

```
6004 \begin{lateximage}[(\theequation) \textbackslash{begin\{equation\}}}%
6005 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}}}%
6006 \textbackslash{end\{equation\}}}% alt tag
```

Create the actual  $\LaTeX$ -formatted equation inside the `lateximage` using the contents of the environment.

```
6007 \LWR@origequation
6008 \BODY% contents collected by NewEnviron
6009 \LWR@origendequation
6010 \end{lateximage}%
6011 }
6012
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
6013 }[\ifbool{mathjax}{\LWR@addmathjax{equation}{\BODY}}{}]
```

## 61.4 AMS Math environments

### 61.4.1 Support macros

Bool `LWR@amsmultline` True if processing a multiline environment.

To compensate for multiline-specific code, `LWR@amsmultline` is used to add extra horizontal space in `\LWR@htmlmathlabel` if is used in an `amsmath` environment which is not a `multiline` environment and not an `equation`.

```
6014 \newbool{LWR@amsmultline}
6015 \boolfalse{LWR@amsmultline}
```

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

`lwarp` points `\ltx@label` here. This is used by `\label` when inside a  $\text{\LaTeX}$  AMS math environment's math display environment.

`\LWR@origltx@label` points to the  $\text{\LaTeX}$  original, modified by `lwarp`, then by `amsmath`, then by `cleveref`.

```
6016 \newcommand*{\LWR@htmlmathlabel}[1]{%
6017 \LWR@traceinfo{\LWR@htmlmathlabel #1}%
6018 \ifbool{mathjax}{%
```

The combined  $\text{\LaTeX}$  & HTML label is printed in a `\text` field:

```
6019 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
6020 \ifbool{LWR@amsmultline}{\hspace*{\totwidth@}}
```

Temporarily end the HTML comment, insert the  $\text{\LaTeX}$  & HTML label, then resume the HTML comment. `\@firstofone` is required to remove extra braces introduced by the `amsmath` package.)

```
6021 \LWR@htmlclosecomment%
6022 \LWR@origltx@label{#1}%
```

```

6023 \LWR@htmlopencomment%
6024 }% text
6025 }% mathjax
6026 {%
6027 \LWR@origltx@label{#1}%
6028 }%
6029 }

```

`\LWR@beginhideamsmath` Starts hiding L<sup>A</sup>T<sub>E</sub>X math inside an HTML comment.

```

6030 \newcommand*{\LWR@beginhideamsmath}{
6031 \LWR@stoppars
6032 \LWR@origtilde\LWR@orignewline
6033 \LWR@htmlopencomment
6034
6035 \begingroup
6036 \LWR@restoreorigformatting
6037 }

```

`\LWR@endhideamsmath` Ends hiding L<sup>A</sup>T<sub>E</sub>X math inside an HTML comment.

```

6038 \newcommand*{\LWR@endhideamsmath}{
6039 \endgroup
6040
6041 \LWR@htmlclosecomment
6042 \LWR@orignewline
6043 \LWR@startpars
6044 }

```

### 61.4.2 Environment patches

The following `amsmath` environments already collect their contents in `\@envbody` for further processing. `eqnarray` is not an  $\mathcal{AMS}$  package, and thus requires special handling.

For SVG math: Each environment is encapsulated inside a `lateximage` environment, along with a special `LWRAMSMATHBODY` argument telling `lateximage` to use as the HTML ALT tag the environment's contents which were automatically captured by the  $\mathcal{AMS}$  environment.

For MathJax: Each environment is syched with L<sup>A</sup>T<sub>E</sub>X's equation numbers, typeset with L<sup>A</sup>T<sub>E</sub>X inside an HTML comment, then printed to HTML output for MathJax to process.



Env `eqnarray` This environment is not an  $\mathcal{AMS}$  environment and thus its body is not automatically captured, so the `environ` package is used to capture the environment into `\BODY`.

```
6045 \let\LWR@origeqnarray\eqnarray
6046 \let\LWR@origendeqnarray\endeqnarray
```

To remember whether the starred environment was used, and thus whether to number the equations:

```
6047 \newbool{LWR@numbereqnarray}
6048 \booltrue{LWR@numbereqnarray}
```

Common code used by `eqnarray` and `Beqnarray` (from `fancybox`):

```
6049 \newcommand{\LWR@eqnarrayfactor}{%
```

Different behavior depending on MathJax vs SVG math:

```
6050 \ifbool{mathjax}
6051 {
```

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:

```
6052 \LWR@syncmathjax
6053 \boolfalse{LWR@amsmultline}
6054 \ifbool{LWR@numbereqnarray}
6055 {
```

If numbering the equations, execute a copy inside an HTML comment block:

```
6056 \LWR@beginhideamsmath
6057 \LWR@origeqnarray
6058 \BODY
6059 \LWR@origendeqnarray
6060 \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MathJax to interpret:

```
6061 \LWR@addmathjax{eqnarray}{\BODY}
6062 }%
6063 {% not LWR@numbereqnarray
```

If not numbering equations, just create the contents for MathJax:

```
6064 \LWR@addmathjax{eqnarray*}{\BODY}
6065 }% LWR@numbereqnarray
```

```

6066 }% mathjax
6067 {% not mathjax
6068     \ifbool{LWR@numbreqnarray}
6069     {

```

For numbered SVG equations, first create a `lateximage` with an `alt` attribute containing sanitized copy of the source code:

```

6070         \begin{lateximage}[\LWR@addmathjax{eqnarray*}{\BODY}]

```

Then create the image contents using an actual `eqnarray`:

```

6071         \LWR@origeqnarray
6072         \BODY
6073         \LWR@origendeqnarray
6074         \end{lateximage}
6075     }%
6076     {% not LWR@numbreqnarray

```

If not numbered, do the same, but an extra `\nonumber` seems to be required:

```

6077         \begin{lateximage}[\LWR@addmathjax{eqnarray*}{\BODY}]
6078         \LWR@origeqnarray
6079         \BODY
6080         \nonumber
6081         \LWR@origendeqnarray
6082         \end{lateximage}
6083     }% LWR@numbreqnarray
6084 }% not mathjax

```

Default to number equations in the future:

```

6085 \booltrue{LWR@numbreqnarray}
6086 }

```

`eqnarray` itself is made with a blank line before and after to force it to be on its own line:

```

6087 \RenewEnviron{eqnarray}
6088 {%
6089
6090 \LWR@eqnarrayfactor
6091
6092 }

```

The starred version is patched to turn off the numbering:

```

6093 \csgpreto{eqnarray*}{\boolfalse{LWR@numbreqnarray}}

```

The following  $\mathcal{AMS}$  environments are more easily patched in-place:

Env `multline`

```

6094 \BeforeBeginEnvironment{multline}{
6095
6096 \ifbool{mathjax}
6097 {
6098   \LWR@syncmathjax
6099   \booltrue{LWR@amsmultline}
6100   \LWR@beginhideamsmath
6101 }
6102 {
6103   \lateximage[LWRAMSMATHBODY]
6104 }
6105 }
6106
6107 \AfterEndEnvironment{multline}{
6108
6109 \ifbool{mathjax}
6110 {
6111   \LWR@endhideamsmath
6112   \boolfalse{LWR@amsmultline}
6113   \LWR@addmathjax{multline}{\the\@envbody}
6114 }
6115 {\endlateximage}
6116
6117 }
```

Env `multline*`

```

6118 \BeforeBeginEnvironment{multline*}{
6119
6120 \ifbool{mathjax}
6121 {
6122   \LWR@syncmathjax
6123   \booltrue{LWR@amsmultline}
6124   \LWR@beginhideamsmath
6125 }
6126 {
6127   \lateximage[LWRAMSMATHBODY]
6128 }
6129 }
6130
6131 \AfterEndEnvironment{multline*}{
6132
6133 \ifbool{mathjax}
6134 {
6135   \LWR@endhideamsmath
```

```

6136     \boolfalse{LWR@amsmultline}
6137     \LWR@addmathjax{multline*}{\the\@envbody}
6138 }
6139 {\endlateximage}
6140
6141 }
6142

```

Env **gather**

```

6143 \BeforeBeginEnvironment{gather}{
6144
6145 \ifbool{mathjax}
6146 {
6147     \LWR@syncmathjax
6148     \boolfalse{LWR@amsmultline}
6149     \LWR@beginhideamsmath
6150 }
6151 {
6152     \lateximage[LWRAMSMATHBODY]
6153 }
6154 }
6155
6156 \AfterEndEnvironment{gather}{
6157
6158 \ifbool{mathjax}
6159 {
6160     \LWR@endhideamsmath
6161     \LWR@addmathjax{gather}{\the\@envbody}
6162 }
6163 {\endlateximage}
6164
6165 }

```

Env **gather\***

```

6166 \BeforeBeginEnvironment{gather*}{
6167
6168 \ifbool{mathjax}
6169 {
6170     \LWR@syncmathjax
6171     \boolfalse{LWR@amsmultline}
6172     \LWR@beginhideamsmath
6173 }
6174 {
6175     \lateximage[LWRAMSMATHBODY]
6176 }
6177 }
6178

```

```

6179 \AfterEndEnvironment{gather*}{
6180
6181 \ifbool{mathjax}
6182 {
6183     \LWR@endhideamsmath
6184     \LWR@addmathjax{gather*}{\the\@envbody}
6185 }
6186 {\endlateximage}
6187
6188 }

```

Env **align**

```

6189 \BeforeBeginEnvironment{align}{
6190
6191 \ifbool{mathjax}
6192 {
6193     \LWR@syncmathjax
6194     \boolfalse{LWR@amsmultline}
6195     \LWR@beginhideamsmath
6196 }
6197 {
6198     \lateximage[LWRAMSMATHBODY]
6199 }
6200 }
6201
6202 \AfterEndEnvironment{align}{
6203
6204 \ifbool{mathjax}
6205 {
6206     \LWR@endhideamsmath
6207     \LWR@addmathjax{align}{\the\@envbody}
6208 }
6209 {\endlateximage}
6210
6211 }

```

Env **align\***

```

6212 \BeforeBeginEnvironment{align*}{
6213
6214 \ifbool{mathjax}
6215 {
6216     \LWR@syncmathjax
6217     \boolfalse{LWR@amsmultline}
6218     \LWR@beginhideamsmath
6219 }
6220 {
6221     \lateximage[LWRAMSMATHBODY]

```

```

6222 }
6223 }
6224
6225 \AfterEndEnvironment{align*}{
6226
6227 \ifbool{mathjax}
6228 {
6229     \LWR@endhideamsmath
6230     \LWR@addmathjax{align*}{\the\@envbody}
6231 }
6232 {\endlateximage}
6233
6234 }

```

Env **flalign**

```

6235 \BeforeBeginEnvironment{flalign}{
6236
6237 \ifbool{mathjax}
6238 {
6239     \LWR@syncmathjax
6240     \boolfalse{LWR@amsmultline}
6241     \LWR@beginhideamsmath
6242 }
6243 {
6244     \lateximage[LWRAMSMATHBODY]
6245 }
6246 }
6247
6248 \AfterEndEnvironment{flalign}{
6249
6250 \ifbool{mathjax}
6251 {
6252     \LWR@endhideamsmath
6253     \LWR@addmathjax{flalign}{\the\@envbody}
6254 }
6255 {\endlateximage}
6256
6257 }

```

Env **flalign\***

```

6258 \BeforeBeginEnvironment{flalign*}{
6259
6260 \ifbool{mathjax}
6261 {
6262     \LWR@syncmathjax
6263     \boolfalse{LWR@amsmultline}
6264     \LWR@beginhideamsmath

```

```

6265 }
6266 {
6267     \lateximage[LWRAMSMATHBODY]
6268 }
6269 }
6270
6271 \AfterEndEnvironment{flalign*}{
6272
6273 \ifbool{mathjax}
6274 {
6275     \LWR@endhideamsmath
6276     \LWR@addmathjax{flalign*}{\the\@envbody}
6277 }
6278 {\endlateximage}
6279
6280 }

6281 \end{warpHTML}

```

## 62 Lateximages

A `\lateximage` is typeset on its own PDF page inside an HTML comment which starts on the preceding page and ends on following page, and instructions are written to `lateximage.txt` for `lwarpmk` to extract the `\lateximage` from the page of the PDF file then generate an accompanying `.svg` file image file. Meanwhile, instructions to show this image are placed into the HTML file after the comment.

An HTML span is created to hold both the HTML comment, which will have the `pdftotext` conversion, and also the link to the final `.svg` image.

A  $\LaTeX$  label is used to remember which PDF page has the image. A label is used because footnotes, endnotes, and pagenotes may cause the image to appear at a later time. The label is declared along with the image, and so it correctly remembers where the image finally ended up.

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

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

**for HTML output:** 6282 `\begin{warpHTML}`

Ctr `LWR@lateximagenumber` Sequence the images.

```

6283 \newcounter{LWR@lateximagenumber}
6284 \setcounter{LWR@lateximagenumber}{0}

```

Ctr LWR@lateximagedepth Do not create \lateximage inside of \lateximage.

```
6285 \newcounter{LWR@lateximagedepth}
6286 \setcounter{LWR@lateximagedepth}{0}
```

Declare the \LWR@file for writing to generate file lateximages.txt:

```
6287 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
```

A few utility macros to write special characters:

```
6288 \edef\LWR@hashmark{\string#} % for use in \write
6289 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctr LWR@Lipage Used to reference the PDF page number of a lateximage to be written into lateximages.txt.

```
6290 \newcounter{LWR@Lipage}
```

```
6291 \end{warpHTML}
```

for HTML & PRINT: 6292 \begin{warpall}

\LateximageFontSizeName Declares how large to write text in the \lateximage. The .svg file text size should blend well with the surrounding HTML text size.

*Do not include the leading backslash in the name.*

```
6293 \newcommand*{\LateximageFontSizeName}{large}
```

```
6294 \end{warpall}
```

for HTML output: 6295 \begin{warpHTML}

\LWR@HTMLsanitize {\text}

Math expressions are converted to lateximages, and some math environments may contain “&”, “<”, or “>”, which should not be allowed inside an HTML ALT tag, so must convert them to HTML entities.

Two versions follow, depending on expansion needs. There may be a better way...

```
6296 \newcommand{\LWR@HTMLsanitize}[1]{%
6297 \begingroup%
6298 \LWR@FBcancel%
6299 \protect\StrSubstitute{\detokenize{#1}}{%
6300 {\detokenize{&}}%
```



```

6301 {\detokenize{&}}[\LWR@strresult]%
6302 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6303 {\detokenize{<}}%
6304 {\detokenize{&lt;}}%
6305 [\LWR@strresult]%
6306 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6307 {\detokenize{>}}%
6308 {\detokenize{&gt;}}%
6309 [\LWR@strresult]%
6310 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6311 {\detokenize{##}}%
6312 {\#}%
6313 [\LWR@strresult]%
6314 \LWR@strresult%
6315 \endgroup%
6316 }

```

`\LWR@HTMLsanitizeexpand`  $\{ \langle text \rangle \}$

This version expands the argument before sanitizing it.

```

6317 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
6318 \begingroup%
6319 \LWR@FBCancel%
6320 \protect\StrSubstitute{\detokenize\expandafter{#1}}%
6321 {\detokenize{&}}%
6322 {\detokenize{&}}%
6323 [\LWR@strresult]%
6324 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6325 {\detokenize{<}}%
6326 {\detokenize{&lt;}}%
6327 [\LWR@strresult]%
6328 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6329 {\detokenize{>}}%
6330 {\detokenize{&gt;}}%
6331 [\LWR@strresult]%
6332 \LWR@strresult%
6333 \endgroup%
6334 }

```

Env `lateximage`  $[\langle alttag \rangle]$

```

6335 \catcode'\$=\active%
6336
6337 \NewDocumentEnvironment{lateximage}{O{image}}
6338 {%
6339 \LWR@traceinfo{lateximage: starting on page \arabic{page}}%
6340 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%

```

If nesting inside an already-existing lateximage, simply record one more level:

```
6341 {%
6342     \addtocounter{LWR@lateximagedepth}{1}%
6343 }%
```

Otherwise, this is the outer-most lateximage:

```
6344 {% start of outer-most lateximage
```

Starting a new lateximage:

```
6345     \addtocounter{LWR@lateximagenumber}{1}%
6346     \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%
```

While inside a lateximage, locally do not use mathjax:

```
6347     \boolfalse{mathjax}
```

Be sure that are doing a paragraph:

```
6348     \LWR@ensuredoingapar%
```

Next file:

```
6349     \addtocounter{LWR@externalfilecnt}{1}%
6350     \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%
```

Figure out what the next page number will be:

```
6351     \setcounter{pageref}{LWR@Lipage}\theLWR@lateximagenumber}%
6352     \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:

```
6353     \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
6354     class="lateximagesource">{}} \LWR@orignewline
```

Write instructions to the `lateximages.txt` file:

```
6355     \LWR@traceinfo{lateximage: about to write to lateximages.txt}%
6356     \immediate\write\LWR@file{|\theLWR@Lipage|\theLWR@externalfilecnt|}%
```

Place an open comment tag at the bottom of page; footnotes will be above this tag. This will hide any traces of the lateximage PDF page which were picked up by `pdftotext`.

```
6357 \LWR@traceinfo{lateximage: about to create open comment}%
6358 \LWR@htmlopencomment%
```

One level deeper:

```
6359 \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
6360 \LWR@traceinfo{lateximage: about to create new page}%
6361 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
6362 \LWR@traceinfo{lateximage: about to create minipage}%
6363 \LWR@originminipage{6in}%
6364 \csuse{LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for `\hspace`, etc. inside a `lateximage`.

```
6365 \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
6366 \LWR@restoreorigformatting%
```

Use full-page footnotes instead of minipage footnotes. These become HTML footnotes.

```
6367 \def\@mpfn{footnote}%
6368 \def\thempfn{\thefootnote}%
6369 \let\@footnotetext\LWR@footnotetext%
```

Create the `LWRlateximage<number>` label:

```
6370 \LWR@traceinfo{lateximage: about to create label}%
6371 \LWR@origlabel{LWRlateximage\arabic{LWR@lateximagenumber}}%
6372 \LWR@traceinfo{lateximage: finished creating the label}%
```

Enable print-mode math functions:

```
6373 \LetLtxMacro$\LWR@origdollar%
6374 \catcode'\$=3% math shift
6375 \LetLtxMacro\(\LWR@origopenparen%
6376 \LetLtxMacro\)\LWR@origcloseparen%
6377 \LetLtxMacro\[ \LWR@origopenbracket%
6378 \LetLtxMacro\] \LWR@origclosebracket%
6379 \let\@ensuredmath\LWR@orig@ensuredmath%
6380 }% end of outer-most lateximage
6381 \LWR@traceinfo{lateximage: finished start of environment}%
6382 }% end of \begin{lateximage}
```

When the environment closes:

```
6383 {% start of \end{lateximage}
6384 \LWR@traceinfo{lateximage: starting end of environment}%
```

Nested more than one deep?

```
6385 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
6386 {\addtocounter{LWR@lateximagedepth}{-1}}%
```

If this is the outer-most lateximage:

```
6387 {% end of outer-most lateximage
```

Finish the lateximage minipage and start a new PDF page:

```
6388 \LWR@origendminipage%
6389 \LWR@orignewpage%
6390 \LWR@origscriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by `pdftotext`:

```
6391 \LWR@htmlclosecomment{}\LWR@orignewline%
6392 \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}%
```

Create a link to the lateximage, allowing its natural height:

If the alt tag is given as “LWRAMSMATHBODY”, then use the text collected by the `amsmath` `multline`, `gather`, or `align` environments.

```
6393 \ifthenelse{\equal{#1}{LWRAMSMATHBODY}}%
6394 {%
6395 \LWR@subinlineimage[%
6396 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}%
6397 ]%
6398 {lateximage}%
6399 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}%
6400 {svg}%
6401 {}%
6402 }%
6403 {%
6404 \LWR@subinlineimage[#1]{lateximage}%
6405 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}{svg}{}%
6406 }%
6407 % \LWR@orignewline% Removed to prevent extra space.
```

Be sure that are doing a paragraph:

```
6408 \LWR@ensuredoingapar%
```

Close the HTML span which has the `pdftotext` comment and also the link to the `.svg` image:

```
6409 \LWR@htmltag{/span}%
6410 \ifbool{HTMLDebugComments}{%
6411 \LWR@htmlcomment{End of lateximage}%
6412 }{}%
6413 % \LWR@orignewline% Removed to prevent extra space.
```

Undo one `lateximage` level:

```
6414 \addtocounter{LWR@lateximagedepth}{-1}%
6415 }% end of outer-most lateximage
6416 \LWR@traceinfo{lateximage: done}%
6417 }%
6418 \catcode'\$=3% math shift
6419 \end{warpHTML}
```

```
for PRINT output: 6420 \begin{warpprint}
6421 % \newenvironment{lateximage}[1][\minipage{\linewidth}]{\endminipage}
6422 \newenvironment{lateximage}[1][\minipage{\linewidth}]{\endminipage}
6423 \end{warpprint}
```

## 63 center, flushleft, flushright

```
for HTML output: 6424 \begin{warpHTML}
```

Env **center** Replace `center` functionality with CSS tags:

```
6425 \renewenvironment*{center}
6426 {
6427 \LWR@forcenewpage
6428 \BlockClass{center}
6429 }
6430 {\endBlockClass}
```

Env **flushright**

```
6431 \renewenvironment*{flushright}
6432 {
```

```

6433 \LWR@forcenewpage
6434 \BlockClass{flushright}
6435 }
6436 {\endBlockClass}

```

Env **flushleft**

```


6437 \renewenvironment*{flushleft}
6438 {
6439 \LWR@forcenewpage
6440 \BlockClass{flushleft}
6441 }
6442 {\endBlockClass}

6443 \end{wrapHTML}

```

## 64 Siunitx

Pkg **siunitx**

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

Place the `\SI` expression inside math mode for anything which requires math mode units.

for HTML output: 6444 `\begin{wrapHTML}`

Options for `siunitx`:

```

6445 \PassOptionsToPackage{
6446     detect-mode=true,
6447     per-mode=symbol,% fraction is not seen by pdftotext
6448 %     text-celsius = {\protect\LWRsiunitx@degree}{C},
6449 %     text-degree = {\protect\LWRsiunitx@degree},
6450 }{siunitx}
6451

6452 \end{wrapHTML}

```

## 65 Graphics

Pkg **graphics**

Pkg **graphicx**

- ⚠ **graphics vs. graphicx** If using the older **graphics** syntax, use both optional arguments for `\includegraphics`. A single optional parameter is interpreted as the newer **graphicx** syntax. Note that
- ⚠ **viewports** viewports are not supported by **warp**; the entire image will be shown.
- ⚠ **\graphicspath** `\graphicspath` only works for a single directory; all graphics must be in this directory.
- units** For `\includegraphics`, avoid **px** and **%** units for width and height, or enclose them inside **warpHTML** environments. For font-proportional image sizes, use **ex** or **em**. For fixed-sized images, use **cm**, **mm**, **in**, **pt**, or **pc**. Using the keys `width=.5\linewidth`, or similar for `\textwidth` or `\textheight` to give fixed-sized images proportional to a 6 by 9 inch text area.
- options** `\includegraphics` accepts **width** and **height**, **origin**, **rotate** and **scale**, plus a new **class** key.
- HTML class** With HTML output, `\includegraphics` accepts an optional `class=xyz` keyval combination, and if this is given then the HTML output will include that class for the image. The class is ignored for print output.
- ⚠ **image file types** For `\includegraphics` the user should provide both **.pdf** and **.svg** images, but always refer to **.pdf** images in the document source. All `\includegraphics` references to **.pdf** will automatically be changed to **.svg** for HTML output, and will be left as **.pdf** for print output. Images may also be **.jpg** and **.png**, and will be used as-is for either output.
- \rotatebox** `\rotatebox` accepts the optional **origin** key.
- ⚠ **browser support** `\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The CSS3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike **L<sup>A</sup>T<sub>E</sub>X**, so expect some ugly results for scaling and rotating.
- for HTML output:** 6453 `\begin{warpHTML}`

## 65.1 \graphicspath

`\graphicspath`  $\{\langle path \rangle\}$

6454 `\newcommand*\thisgraphicspath{\{}}`

6455 `\renewcommand*\graphicspath{[1]{\renewcommand*\thisgraphicspath{\#1}}}`

`\DeclareGraphicsExtensions`  $\{\langle list \rangle\}$

```
\DeclareGraphicsRule {<>} {<>} {<>} {<>}
```

```
6456 \renewcommand*{\DeclareGraphicsExtensions}[1]{%
6457 \renewcommand*{\DeclareGraphicsRule}[4]{%}
```

## 65.2 Length conversions and graphics options



whitespace

A scaled image in L<sup>A</sup>T<sub>E</sub>X 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.

```
6458 \renewcommand*{\unitspace}{%}
```

Used to store the user's selected dimensions and HTML class.

The class defaults to “inlineimage” unless changed by a `class=xyx` option.

```
6459 \newlength{\LWR@igwidth}%
6460 \newlength{\LWR@igheight}%
6461 \newcommand*{\LWR@igwidthstyle}{%
6462 \newcommand*{\LWR@igheightstyle}{%
6463 \newcommand*{\LWR@igorigin}{%
6464 \newcommand*{\LWR@igangle}{%
6465 \newcommand*{\LWR@igxscale}{1}%
6466 \newcommand*{\LWR@igyyscale}{1}%
6467 \newcommand*{\LWR@igclass}{inlineimage}
```

Set the actions of each of the key/value combinations for `\includegraphics`. Many are ignored.

If an optional width was given, set an HTML style:

```
6468 \define@key{igraph}{width}{%
6469 \setlength{\LWR@igwidth}{#1}%
6470 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}{%
6471 {%}
```

Default to use the converted fixed length given:

```
6472 \uselengthunit{PT}%
6473 \renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%
```

If ex or em dimensions were given, use those instead:

```
6474 \IfEndWith{#1}{ex}%
6475 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes ex
```



```

6476     {}% not ex
6477     \IfEndWith{#1}{em}%
6478     {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes em
6479     {}% not em
6480     \IfEndWith{#1}{\}%
6481     {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes percent
6482     {}% not percent
6483     \IfEndWith{#1}{px}%
6484     {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
6485     {}% not px
6486 }{}% end of length > Opt
6487 }

```

If an optional height was given, set an HTML style:

```

6488 \define@key{igraph}{height}{%
6489 \setlength{\LWR@igheight}{#1}%
6490 \ifthenelse{\lengthtest{\LWR@igheight > Opt}}%
6491 {%

```

Default to use the converted fixed length given:

```

6492     \uselengthunit{PT}%
6493     \renewcommand*{\LWR@igheightstyle}{%
6494     height:\rndprintlength{\LWR@igheight} %
6495     }%

```

If ex or em dimensions were given, use those instead:

```

6496     \IfEndWith{#1}{ex}%
6497     {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
6498     {}% not ex
6499     \IfEndWith{#1}{em}%
6500     {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
6501     {}% not em
6502     \IfEndWith{#1}{\}%
6503     {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
6504     {}% not percent
6505     \IfEndWith{#1}{px}%
6506     {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
6507     {}% not px
6508 }{}% end of length > Opt
6509 }

```

Handle origin key:

```

6510 \define@key{igraph}{origin}{%
6511 \renewcommand*{\LWR@igorigin}{#1}%
6512 }

```

Handle angle key:

```
6513 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}
```

Handle class key:

```
6514 \define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}
6515
```

It appears that `graphicx` does not have separate keys for `xscale` and `yscale`. `scale` adjusts both at the same time.

```
6516 \define@key{igraph}{scale}{%
6517 \renewcommand*{\LWR@igxscale}{#1}%
6518 \renewcommand*{\LWR@igyscale}{#1}}
```

Numerous ignored keys:

```
6519 \define@key{igraph}{bb}{}
6520 \define@key{igraph}{bblx}{}
6521 \define@key{igraph}{bbly}{}
6522 \define@key{igraph}{bburx}{}
6523 \define@key{igraph}{bbury}{}
6524 \define@key{igraph}{natwidth}{}
6525 \define@key{igraph}{natheight}{}
6526 \define@key{igraph}{hiresbb}{}
6527 \define@key{igraph}{viewport}{}
6528 \define@key{igraph}{trim}{}
6529 \define@key{igraph}{totalheight}{}
6530 \define@key{igraph}{keepaspectratio}{}
6531 \define@key{igraph}{clip}{}
6532 \define@key{igraph}{draft}{}
6533 \define@key{igraph}{type}{}
6534 \define@key{igraph}{ext}{}
6535 \define@key{igraph}{read}{}
6536 \define@key{igraph}{command}{}

```

```
\LWR@rotstyle {<prefix>} {<degrees>}
```

Prints the rotate style with the given prefix.

`prefix` is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:rotate` style.

```
6537 \newcommand*{\LWR@rotstyle}[2]{%
6538 #1transform:rotate(-#2deg);
6539 }
```

```
\LWR@scalestyle {<prefix>} {<xscale>} {<yscale>}
```

Prints the scale style with the given prefix.

**prefix** is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:scale` style.

```
6540 \newcommand*\LWR@scalestyle}[3]{%
6541   #1transform:scale(#2,#3);
6542 }
```

### 65.3 \includegraphics

Bool **LWR@infloatrow** Used to compute `\linewidth`.

```
6543 \newbool{LWR@infloatrow}
6544 \boolfalse{LWR@infloatrow}
```

`\LWR@opacity` may be set by the transparent package. For HTML it is only used for `\includegraphics`.

```
6545 \def\LWR@opacity{1}
```

```
6546 \newcommand*\LWR@imageextension{}
6547 \newcommand*\LWR@expgraphicsfilename{}
```

`\LWR@includegraphicsb` \* [*2: options*] [*3: options*] {*4: filename*}

graphics syntax is `\includegraphics * [llx,lly] [urx,ury] {file}`

graphicx syntax is `\includegraphics [key values] {file}`

If #3 is empty, only one optional argument was given, thus `graphicx` syntax.

```
6548 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
6549 {%
```

Start the image tag on a new line, allow PDF output word wrap:

```
6550 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
6551 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
6552 {%
6553   \ifbool{LWR@infloatrow}%
6554   {}
```

```

6555     {% not in a minipage or a floatrow:
6556         \setlength{\linewidth}{6in}%
6557         \setlength{\textwidth}{6in}%
6558         \setlength{\textheight}{9in}%
6559     }%
6560 }{}%
```

See if can find the image by adding an extension:

Preference is svgz, then svg, gif, png, and jpg.

`\detokenize\expandafter` allows underscore characters in filenames.

```

6561 \edef\LWR@expgraphicsfilename{#4}
6562 \renewcommand*{\LWR@imageextension}{}%
6563 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.jpg}%
6564 {\renewcommand*{\LWR@imageextension}{.jpg}}{}%
6565 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.JPG}%
6566 {\renewcommand*{\LWR@imageextension}{.JPG}}{}%
6567 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.png}%
6568 {\renewcommand*{\LWR@imageextension}{.png}}{}%
6569 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.PNG}%
6570 {\renewcommand*{\LWR@imageextension}{.PNG}}{}%
6571 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.gif}%
6572 {\renewcommand*{\LWR@imageextension}{.gif}}{}%
6573 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.GIF}%
6574 {\renewcommand*{\LWR@imageextension}{.GIF}}{}%
6575 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svg}%
6576 {\renewcommand*{\LWR@imageextension}{.svg}}{}%
6577 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVG}%
6578 {\renewcommand*{\LWR@imageextension}{.SVG}}{}%
6579 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svgz}%
6580 {\renewcommand*{\LWR@imageextension}{.svgz}}{}%
6581 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVGZ}%
6582 {\renewcommand*{\LWR@imageextension}{.SVGZ}}{}%
```

Convert a PDF extension to SVG, leave the result in `\LWR@strresult`:

Must also `\detokenize .pdf` and `.svg` comparison strings.

```

6583 \StrSubstitute{\detokenize\expandafter{\LWR@expgraphicsfilename}}%
6584 {\detokenize{.pdf}}{\detokenize{.svg}}[\LWR@strresult]%
6585 %
6586 \StrSubstitute{\LWR@strresult}%
6587 {\detokenize{.PDF}}{\detokenize{.SVG}}[\LWR@strresult]%
```

For correct em sizing during the width and height conversions:

```

6588 \large%
```

Reset some defaults, possibly will be changed below if options were given:

```
6589 \setlength{\LWR@igwidth}{0pt}%
6590 \setlength{\LWR@igheight}{0pt}%
6591 \renewcommand*\LWR@igwidthstyle{}%
6592 \renewcommand*\LWR@igheightstyle{}%
6593 \renewcommand*\LWR@igorigin{}%
6594 \renewcommand*\LWR@igangle{}%
6595 \renewcommand*\LWR@igxscale{1}%
6596 \renewcommand*\LWR@igyyscale{1}%
6597 \renewcommand*\LWR@igclass{inlineimage}%
```

If #3 is empty, only one optional argument was given, thus `graphicx` syntax:

```
6598 \IfValueTF{#3}{}%
6599 \IfValueTF{#2}%
6600 {\setkeys{igraph}{#2}}%
6601 {\setkeys{igraph}{}}%
6602 }%
```

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.

```
6603 \href{\thisgraphicspath\LWR@strresult\LWR@imageextension}%
6604 {% start of href
6605 \LWR@htmltag{% start of image tags
6606 img src="\thisgraphicspath\LWR@strresult\LWR@imageextension" \LWR@orignewline
6607 \LWR@origtilde{} alt="\LWR@strresult" \LWR@orignewline
```

Only include a style tag if a width, height, angle, or scale was given:

```
6608 \ifthenelse{
6609   \NOT\equal{\LWR@igwidthstyle}{} \OR
6610   \NOT\equal{\LWR@igheightstyle}{} \OR
6611   \NOT\equal{\LWR@igorigin}{} \OR
6612   \NOT\equal{\LWR@igangle}{} \OR
6613   \NOT\equal{\LWR@igxscale}{1} \OR
6614   \NOT\equal{\LWR@igyyscale}{1}
6615 }%
6616 {\LWR@origtilde{} style="%
6617 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{}%
6618 {\LWR@igwidthstyle;}{}%
6619 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{}%
6620 {\LWR@igheightstyle;}{}%
6621 \ifthenelse{\NOT\equal{\LWR@igorigin}{}%
6622 {\LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{}%
```

```

6623 \ifthenelse{\NOT\equal{\LWR@igangle}{}}%
6624 {%
6625 \LWR@rotstyle{-ms-}{\LWR@igangle}%
6626 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
6627 \LWR@rotstyle{}{\LWR@igangle%
6628 }}{}%
6629 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%
6630 \NOT\equal{\LWR@igyscale}{1}}%
6631 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}%
6632 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
6633 \LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyscale}}{}%
6634 %
6635 \ifthenelse{\NOT\equal{\LWR@opacity}{1}}%
6636 {opacity:\LWR@opacity;}%
6637 {}%
6638 %
6639 " \LWR@orignewline}{}%

```

Set the class:

```

6640 \LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%
6641 }% end of image tags
6642 }% end of href
6643 \endgroup

```

Return to small-sized output:

```

6644 \LWR@origscriptsize
6645 }

```

`\includegraphics` [*(key=val)*] {*(filename)*}

Handles width and height, converted to fixed width and heights.

Converts any .pdf references to .svg for HTML

The user should always refer to .pdf in the document source.

```

6646 \renewcommand*\includegraphics{
6647 {%

```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```

6648 \LWR@ensuredoingapar%
6649 \begingroup%
6650 \LWR@includegraphicsb%
6651 }

```

```
6652 \end{warpHTML}
```

for **PRINT output**: For print output, accept and then discard the new `class` key:

```
6653 \begin{warpprint}
6654 \define@key{Gin}{class}{}
6655 \end{warpprint}
```

## 65.4 \rotatebox, \scalebox, \reflectbox

for **HTML output**: 6656 \begin{warpHTML}

\LWR@rotboxorigin Holds the origin key letters.

```
6657 \newcommand*\LWR@rotboxorigin{}
```

\LWR@originname {\langle letter \rangle}

Given one L<sup>A</sup>T<sub>E</sub>X origin key value, translate into an HTML origin word:

```
6658 \newcommand*\LWR@originname[1]{%
6659 \ifthenelse{\equal{#1}{t}}{top}{}%
6660 \ifthenelse{\equal{#1}{b}}{bottom}{}%
6661 \ifthenelse{\equal{#1}{c}}{center}{}%
6662 \ifthenelse{\equal{#1}{l}}{left}{}%
6663 \ifthenelse{\equal{#1}{r}}{right}{}%
6664 }
```

\LWR@originnames {\langle letters \rangle}

Given one- or two-letter L<sup>A</sup>T<sub>E</sub>X origin key values, translate into HTML origin words:

```
6665 \newcommand*\LWR@originnames[1]{%
6666 \StrChar{#1}{1}[\LWR@strresult]%
6667 \LWR@originname{\LWR@strresult}
6668 \StrChar{#1}{2}[\LWR@strresult]%
6669 \LWR@originname{\LWR@strresult}
6670 }
```

Handle the origin key for \rotatebox:

```
6671 \define@key{krotbox}{origin}{%
6672 \renewcommand*\LWR@rotboxorigin{#1}%
6673 }
```

These keys are ignored:

```
6674 \define@key{krotbox}{x}{}
6675 \define@key{krotbox}{y}{}
6676 \define@key{krotbox}{units}{}

```

`\rotatebox` [*keyval list*] {*angle*} {*text*}

Will `\let\rotatebox\LWR@rotatebox` at `\LWR@LwarpStart`, in case `\rotatebox` was over-written by a later package load.

```
6677 \LetLtxMacro\LWR@origrotatebox\rotatebox
6678
6679 \NewDocumentCommand{\LWR@rotatebox}{O{} m +m}{%

```

Reset the origin to “none-given”:

```
6680 \renewcommand*\LWR@rotboxorigin{}

```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
6681 \setkeys{krotbox}{#1}%

```

Select `inline-block` so that HTML will transform this span:

```
6682 \LWR@htmltagc{span style="display: inline-block; %

```

If an origin was given, translate and print the origin information:

```
6683 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}{%
6684 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}}%

```

Print the rotation information:

```
6685 \LWR@rotstyle{-ms-}{#2} %
6686 \LWR@rotstyle{-webkit-}{#2} %
6687 \LWR@rotstyle{}{#2} %
6688 "{}\LWR@orignewline%

```

Print the text to be rotated:

```
6689 \begin{LWR@nestspan}%
6690 #3%

```

Close the span:

```
6691 \LWR@htmltagc{/span}%
6692 \end{LWR@nestspan}%
6693 }

```



`\scalebox`  $\{\langle h\text{-}scale\rangle\} [\langle v\text{-}scale\rangle] \{\langle text\rangle\}$

Will `\let\scalebox\LWR@scalebox` at `\LWR@LwarpStart`, in case `\scalebox` was over-written by a later package load.

```
6694 \LetLtxMacro\LWR@origscalebox\scalebox
6695
6696 \NewDocumentCommand{\LWR@scalebox}{m o m}{%
```

Select `inline-block` so that HTML will transform this span:

```
6697 \LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
6698 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6699 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6700 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6701 "}}%
```

Print the text to be scaled:

```
6702 \begin{LWR@nestspan}%
6703 #3%
```

Close the span:

```
6704 \LWR@htmltagc{/span}%
6705 \end{LWR@nestspan}%
6706 }
```

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

Will `\let\reflectbox\LWR@reflectbox` at `\LWR@LwarpStart`, in case `\reflectbox` was over-written by a later package load.

```
6707 \let\LWR@origreflectbox\reflectbox
6708
6709 \newcommand{\LWR@reflectbox}[1]{\LWR@scalebox{-1}[1]{#1}}
6710 \end{warpHTML}
```

## 65.5 Null functions

These functions are not supported by lwarp's HTML conversion.

for HTML output: 6711 `\begin{warpHTML}`

```
\resizebox {<h-length>} {<v-length>} {<text>}
```

Simply prints its text argument.

```
6712 \renewcommand{\resizebox}[3]{#3}
```

```
6713 \end{warpHTML}
```

## 66 Xcolor defaults

The following are used when xcolor is not loaded:

### 66.1 Defaults without xcolor

for HTML & PRINT: 6714 \begin{warpall}

The following is used if xcolor is not loaded, and is replaced when lwarp-xcolor is loaded.

If xcolor is not loaded, there is no xcolor formatting to restore:

```
6715 \newcommand*\LWR@restoreorigxcolor{}
6716 \newcommand*\LWR@restoremorigxcolor{}
6717 \end{warpall}
```

### 66.2 xcolor print-mode additions

Print-mode versions of new xcolor defintions. These are defined inside `warpall` because they are also used for HTML while inside a `lateximage`. They are defined `\AtBeginDocument` so that the xcolor originals may first be 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: 6718 \begin{warpall}

After xparse may have been loaded ...

```
6719 \AtBeginDocument{
```

... and *only* if xcolor was loaded:

```
6720 \@ifpackageloaded{xcolor}{
6721 \LWR@traceinfo{patching xcolor}
```

`\colorboxBlock` `\colorboxBlock` is the same as `\colorbox`:

```
6722 \LetLtxMacro\colorboxBlock\colorbox
```

In HTML mode, the following is done when xcolor is loaded. Following is the print-mode action:

```
6723 \warpprintonly{
6724 \LetLtxMacro\LWRprint@colorboxBlock\colorbox
6725 \LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox
6726 \LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox
6727 }
```

`\fcolorbox` [`\framecolor`] [`\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`.)

```
6728 \newsavebox{\LWR@colorminipagebox}
6729
6730 \DeclareDocumentCommand{\LWRprint@fcolorbox}{o m o m +m}{%
6731 \LWR@traceinfo{\LWRprint@fcolorbox #2 #4}%
```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
6732 \begin{lrbox}{\LWR@colorminipagebox}%
6733 #5%
6734 \end{lrbox}%
```

Sort out the various optional arguments and the background color of `none`. In each case, the LRbox is placed inside a `\fcolorbox`.

```
6735 \ifthenelse{\equal{#4}{none}}{%
6736 {% #4 none
6737   \LWR@traceinfo{background is none}%
6738   {% scope the \colorlet
6739     \colorlet{\LWR@currentcolor}{.}%
6740     \color{#2}%
6741     \fbox{%
6742       \color{\LWR@currentcolor}%
```

```

6743         \usebox{\LWR@colorminipagebox}%
6744     }% fbox
6745     }% colorlet
6746 }% #4 none
6747 {% #4 not none
6748 \LWR@traceinfo{background not none}%
6749 \IfValueTF{#1}%
6750 {%
6751     \IfValueTF{#3}%
6752     {\LWRorigprint@fcolorbox{#1}{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
6753     {\LWRorigprint@fcolorbox{#1}{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
6754 }%
6755 {% no value #1
6756     \IfValueTF{#3}%
6757     {\LWRorigprint@fcolorbox{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
6758     {\LWRorigprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
6759 }% no value #1
6760 }% #4 not none
6761 \LWR@traceinfo{\LWRprint@fcolorbox done}%
6762 }

```

`\fcolorboxBlock` [*framemodel*] [*framecolor*] [*boxmodel*] [*boxcolor*] [*text*]

In print mode, `\fcolorboxBlock` is the same as `\fcolorbox`.

```
6763 \LetLtxMacro\LWRprint@fcolorboxBlock\LWRprint@fcolorbox
```

Env `fcolorminipage` [*1:framemodel*] [*2:framecolor*] [*3:boxmodel*] [*4:boxcolor*] [*5:align*]  
 [*6:height*] [*7:inner-align*] [*8:width*]

In print mode, becomes a `\fcolorbox` containing a minipage:

```

6764 \NewDocumentEnvironment{LWRprint@fcolorminipage}{o m o m O{c} O{ } o m}
6765 {%
6766 \LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%

```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
6767 \begin{lrbox}{\LWR@colorminipagebox}%
```

If inner alignment is not given, use the outer alignment instead:

```

6768 \IfValueTF{#7}%
6769 {\begin{minipage}[#5][#6][#7]{#8}}%
6770 {\begin{minipage}[#5][#6][#5]{#8}}%
6771 }%
6772 {%
6773 \end{minipage}%

```

```
6774 \end{lrbox}%
6775 \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.

```
6776 \ifthenelse{\equal{#4}{none}}%
6777 {% #4 none
6778     {% scope the \colorlet
6779         \colorlet{LWR@currentcolor}{.}%
6780         \color{#2}%
6781         \fbox{%
6782             \color{LWR@currentcolor}%
6783             \usebox{\LWR@colorminipagebox}}%
6784         }% fbox
6785     }% colorlet
6786 }% #4 none
6787 {% #4 not none
6788     \IfValueTF{#1}%
6789     {%
6790         \IfValueTF{#3}%
6791         {\LWRorigprint@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
6792         {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
6793     }%
6794     {% no value #1
6795         \IfValueTF{#3}%
6796         {\LWRorigprint@fcolorbox{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
6797         {\LWRorigprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
6798     }% no value #1
6799 }% #4 not none
6800 \LWR@traceinfo{*** finished end fcolorminipage}%
6801 }
```

`\LWR@restoremoreorigxcolor` Used to activate print-mode additions for xcolor. In print mode, this is used immediately following. In HTML mode, this is used inside a `lateximage`.

```
6802 \renewcommand*{\LWR@restoremoreorigxcolor}{%
6803 \LWR@traceinfo{\LWR@restoremoreorigxcolorformatting}%
6804 \LetLtxMacro\colorboxBlock\LWRprint@colorboxBlock%
6805 \LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
6806 \LetLtxMacro\fcolorboxBlock\LWRprint@fcolorboxBlock%
6807 \LetLtxMacro\fcolorminipage\LWRprint@fcolorminipage%
6808 \LetLtxMacro\endfcolorminipage\endLWRprint@fcolorminipage%
6809 }
```

If print mode, immediately activate the print-mode enhancements for xcolor:

```
6810 \warpprintonly{\LWR@restoremoreorigxcolor}
```

```

6811
6812 \LWR@traceinfo{xcolor patches done}
6813 }{}% xcolor
6814 }% AtBeginDocument

6815 \end{warpall}

```

## 67 Cleveref

**Pkg** `cleveref` `cleveref` package is used as-is with minor patches.

**loading order** `cleveref` and the following associated macro patches are automatically preloaded at the end of the preamble via `\AtEndPreamble` and `\AfterEndPreamble`. This is done because the HTML conversion requires `cleveref`. The user's document may not require `cleveref`, thus the user may never explicitly load it, so during HTML output `lwarp` loads it last. If the user's document preamble uses `cleveref` options, or functions such as `\crefname`, then `cleveref` may be loaded in the user's preamble near the end, and `lwarp`'s additional loading of `cleveref` will have no effect.

Table 9 on 278 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:** 6816 `\begin{warpHTML}`

`\AtEndPreamble` forces `cleveref` to be loaded last:

```

6817 \AtEndPreamble{
6818 \RequirePackage{cleveref}
6819 }

```

The following patches are applied after `cleveref` has loaded, and after `\AtBeginDocument`:

```

6820 \AfterEndPreamble{
6821 \LWR@traceinfo{Patching cleveref.}

```

```
\@@setcref {<kindofref>} {<label>}
```

```
6822 \renewcommand*{\@@setcref}[2]{#1{\ref{#2}}}{}}
```

```
\@@setcrefrange {<text>} {<label>} {<label>}
```

```
6823 \renewcommand{\@@setcrefrange}[3]{%
6824 #1{\ref{#2}}{\ref{#3}}{}{}{}}
```

`\cpagerefFor` Redefinable word between “page(s)” and the page numbers.

```
6825 \newcommand*\cpagerefFor{for}
```

`\@@setcpageref`  $\langle typeofref \rangle$   $\langle label \rangle$ , where *typeofref* is “page” or “pages”

```
6826 \renewcommand*\@@setcpageref}[2]{%
6827 #1{\cpagerefFor\ \cref{#2}}{}{}%
6828 }

6829 \renewcommand{\@@setcpagerefrange}[3]{%
6830 #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}{}{}}
6831 }% AfterEndPreamble
```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

```
6832 \let\LWR@origlabel\label
6833 \let\label\LWR@newlabel
6834 \let\LWR@origref\ref
6835 \let\ref\LWR@newref% \end{ syntax highlighting
6836 \let\LWR@origpageref\pageref
6837 \let\pageref\LWR@newpageref
6838
6839
6840
6841 \end{warpHTML}
```

## 68 Picture

Env `picture` The picture environment is enclosed inside a `\lateximage`.

for HTML output: 6842 `\begin{warpHTML}`


Env `picture`

```
6843 \BeforeBeginEnvironment{picture}{\lateximage}
6844
6845 \AfterEndEnvironment{picture}{\endlateximage}

6846 \end{warpHTML}
```

## 69 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 `<span>`. An HTML `<div>` cannot appear inside a `<span>`. While in a `<span>`, 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<sup>A</sup>T<sub>E</sub>X PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

**for HTML output:** 6847 `\begin{warphTML}`

### 69.1 Counters and lengths

**Ctr** `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
6848 \newcounter{LWR@minipagedepth}
6849 \setcounter{LWR@minipagedepth}{0}
```

**Len** `\WR@minipagewidth` Used to convert the width into printable units.

```
6850 \newlength{\LWR@minipagewidth}
```



Len `\WR@minipageheight` Used to convert the height into printable units.

```
6851 \newlength{\LWR@minipageheight}
```

## 69.2 Footnote handling

Also see section 43 for other forms of footnotes. Minipage footnotes are gathered in section 43.5, and then placed into the document in section 69.3.

## 69.3 Minipage handling

`\LWR@endminipage` Used to close a minipage.

Copied the L<sup>A</sup>T<sub>E</sub>X definition and modified to create a `mpfootnotes` div class:

```
6852 \def\LWR@endminipage{%
6853   \par
6854   \unskip
6855   \ifvoid\@mpfootins\else
6856     \vskip\skip\@mpfootins
6857     \normalcolor
6858     \LWR@htmldivclass{mpfootnotes}
6859     \LWR@origmedskip
6860     \unvbox\@mpfootins
6861     \LWR@htmldivclassend{mpfootnotes}
6862   \fi
6863   \@minipagefalse
6864   \color@endgroup
6865   \egroup
6866   \expandafter\@iiiparbox\@mpargs{\unvbox\@tempboxa}}
```

`\LWR@subminipage` Used to create a PDF minipage without creating an HTML minipage. This allows footnotes to appear at the bottom of the minipage instead of the bottom of the HTML page.

```
6867 \newcommand*{\LWR@subminipage}{%
6868 \LWR@stoppars
6869 \LWR@origminipage{6in}
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
6870 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
6871 \LWR@startpars%
6872 }
```

`\LWR@endsubminipage` Closes the subminipage.

```
6873 \newcommand*{\LWR@endsubminipage}{%
6874 \LWR@stoppars%
6875 \LWR@endminipage% The following empty line is required:
6876
6877 }
```

Bool `LWR@minipagefullwidth` Should the next minipage have no HTML width?

```
6878 \newbool{LWR@minipagefullwidth}
6879 \boolfalse{LWR@minipagefullwidth}
```

`\minipagefullwidth` Requests that the next minipage have no width tag in HTML:

**for HTML output:**

```
6880 \newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}
6881 \end{warppHTML}
```

**for PRINT output:**

```
6882 \begin{warpprint}
6883 \newcommand*{\minipagefullwidth}{}
6884 \end{warpprint}
```

**for HTML output:**

```
6885 \begin{warppHTML}
```

Bool `LWR@minipagethispar` Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
6886 \newbool{LWR@minipagethispar}
6887 \boolfalse{LWR@minipagethispar}
```

Env `minipage` [*<vert position>*] [*<height>*] [*<inner vert position>*] {*<width>*}

The vertical positions may be 'c', 't', or 'b'. The inner position may also be 's'.

When using `\linewidth`, `\textwidth`, or `\textheight`, these are scaled proportionally to a 6×9 inch text area.

```
6888 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}
6889 {%
6890 \uselengthunit{in}%
6891 \setlength{\LWR@minipagewidth}{#4}%
6892 \ifthenelse{\cinttest{\value{LWR@minipagedepth}}{=}{0}}{%
```

```

6893 \addtolength{\LWR@minipagewidth}{3em}% room for frames
6894 \setlength{\linewidth}{6in}%
6895 \setlength{\textwidth}{6in}%
6896 \setlength{\textheight}{9in}%
6897 }{}%
6898 \LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}
6899 \addtocounter{LWR@minipagedepth}{1}%
6900 \setlength{\LWR@minipageheight}{\textheight}% default unless specified
6901 \IfValueTF{#2}{\setlength{\LWR@minipageheight}{#2}}{}%

```

L<sup>A</sup>T<sub>E</sub>X wants to start a paragraph for the new minipage, then start a paragraph again for the contents of the minipage, so cancel the paragraph tag handling until the minipage has begun.

```

6902 \LWR@stoppars%

```

Create the <div> tag with optional alignment style:

```

6903 \LWR@traceinfo{minipage: creating div class}%
6904 \LWR@orignewpage%
6905 \LWR@htmltag{div class="minipage" style="%
6906 \ifthenelse{\equal{#1}{t}}{vertical-align:bottom ; }{}%
6907 \ifthenelse{\equal{#1}{c}}{vertical-align:middle ; }{}%
6908 \ifthenelse{\equal{#1}{b}}{vertical-align:top ; }{}%
6909 \ifthenelse{\equal{#3}{t}}{justify-content:flex-start ; }{}%
6910 \ifthenelse{\equal{#3}{c}}{justify-content:center ; }{}%
6911 \ifthenelse{\equal{#3}{b}}{justify-content:flex-end ; }{}%
6912 \ifthenelse{\equal{#3}{s}}{justify-content:space-between ; }{}%

```

Print the width and optional height styles:

```

6913 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
6914 \uselengthunit{PT}%
6915 \ifbool{LWR@minipagefullwidth}%
6916 {\boolfalse{LWR@minipagefullwidth}}%
6917 {%
6918 \ifthenelse{\lengthtest{#4}=\linewidth}%
6919 {}%
6920 {width:\rndprintlength{\LWR@minipagewidth} ; }%
6921 }%
6922 \LWR@traceinfo{minipage: about to print the height}%
6923 \IfValueTF{#2}{height:\rndprintlength{\LWR@minipageheight} ; }{}%
6924 "%}%

```

Finish with an empty line to start L<sup>A</sup>T<sub>E</sub>X minipage processing on a new line. Use a large minipage area to avoid the unnecessary wrapping of tags.

```

6925
6926 \LWR@origminipage{6in}% The preceding empty line is required.

```

Set the user-accessible minipage and text width and height values inside the minipage. These do not affect the actual size of the large minipage created by `\LWR@origminipage` above, but are used by any reference to `\linewidth`, etc. inside the PDF minipage being created here.

```
6927 \setlength{\linewidth}{#4}% the original width
6928 \setlength{\textwidth}{6in}%
6929 \setlength{\textheight}{9in}%
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
6930 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
6931 \LWR@startpars%
6932 \LWR@traceinfo{minipage: finished starting the minipage}%
6933 }
```

End the environment with L<sup>A</sup>T<sub>E</sub>X processing and closing tag:

```
6934 {%
6935 \LWR@stoppars%
6936 \LWR@endminipage% The following empty line is required:
6937
6938 \LWR@htmldivclassend{minipage}%
6939 \LWR@origvspace{1\baselineskip}% required for subcaption
6940 \addtocounter{LWR@minipagedepth}{-1}%
6941 \LWR@startpars%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
6942 \global\booltrue{LWR@minipagethispar}%
6943 }
```

## 69.4 Parbox, makebox, framebox, fbox, raisebox

for HTML output:  
`\parbox`

```
[\langle pos \rangle] [\langle height \rangle] [\langle inner-pos \rangle] {\langle width \rangle} {\langle text \rangle}
```

A parbox uses the minipage code:

```
6944 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}
6945 {
6946 \LWR@traceinfo{parbox of width #4}%
6947 \begin{minipage}[#1][#2][#3]{#4}
```

```

6948 #5
6949 \end{minipage}
6950 }

```

`\makebox` ( $\langle()$ posn) [ $\langle width\rangle$ ] [ $\langle pos\rangle$ ] { $\langle text\rangle$ }

```

6951 \LetLtxMacro\LWR@origmakebox\makebox
6952
6953 \RenewDocumentCommand{\makebox}{d() o o m}{%

```

Check for the optional width:

```

6954 \IfValueTF{#2}%
6955 {%

```

Check for the horizontal text alignment. For `stretched`, the best HTML can do is `justified` alignment.

```

6956     {% scope
6957     \def\LWR@align{center}%
6958     \ifstrequal{#3}{l}{\def\LWR@align{left}}{}%
6959     \ifstrequal{#3}{r}{\def\LWR@align{right}}{}%
6960     \ifstrequal{#3}{s}{\def\LWR@align{justify}}{}%

```

To print the width argument:

```

6961     \setlength{\LWR@tempwidth}{#2}%

```

`inline-block` allows width and text-alignment to be used in a `<span>`.

```

6962     \uselengthunit{PT}%
6963     \InlineClass[%
6964         display:inline-block ; %
6965         text-align:\LWR@align\ ; %
6966         width:\rndprintlength{\LWR@tempwidth}%
6967     ]%
6968     {makebox}%

```

Without a width argument, the text is simply used inline:

```

6969     {\mbox{#4}}%
6970     }% scope
6971 }%
6972 {\mbox{#4}}%
6973 }

```

`\framebox` [ $\langle width\rangle$ ] [ $\langle pos\rangle$ ] { $\langle text\rangle$ }

```

6974 \LetLtxMacro\LWR@origframebox\framebox
6975
6976 \RenewDocumentCommand{\framebox}{o o m}{%
6977 \fbox{\makebox[#1][#2]{#3}}}%
6978 }

```

`\LWR@forceminwidth`  $\{\langle legth \rangle\}$

Sets `\LWR@tempwidth` to be at least 1pt.

```

6979 \newlength{\LWR@atleastonept}
6980
6981 \newcommand*{\LWR@forceminwidth}[1]{%
6982 \setlength{\LWR@atleastonept}{#1}%
6983 \ifthenelse{%
6984     \lengthtest{\LWR@atleastonept>0pt}\AND%
6985     \lengthtest{\LWR@atleastonept<1pt}}%
6986 }%
6987 {\setlength{\LWR@atleastonept}{1pt}}%
6988 {}%
6989 }

```

`\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.

```

6990 \newcommand*{\LWR@blackborderpadding}{%
6991 \uselengthunit{PT}}%
6992 border:\rndprintlength{\LWR@atleastonept} solid black ; %
6993 padding:\rndprintlength{\fboxsep}%
6994 }

```

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

Creates a framed inline span enclosing the text.

Remember the print-mode version:

```

6995 \let\LWRprint@fbox\fbox

```

Create a new HTML version, but don't use it until after `xcolor` may have loaded:

```

6996 \newcommand{\LWRhtml@fbox}[1]{%
6997 \LWR@traceinfo{HTML fbox}
6998 \LWR@forceminwidth{\fboxrule}%
6999 \InlineClass[%
7000 \LWR@blackborderpadding%
7001 ]{fbox}{#1}
7002 }

```

`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`:

```
7003 \AtBeginDocument{\let\fbox\LWRhtml@fbox}
```

`\fboxBlock`  $\{ \langle text \rangle \}$  Creates a framed HTML `<div>` of the text.

A print-output version is also supplied below.

```
7004 \newcommand{\fboxBlock}[1]{%
7005 \LWR@forceminwidth{\fboxrule}%
7006 \begin{BlockClass}[%
7007 \LWR@blackborderpadding%
7008 ]{\fboxBlock}
7009 #1
7010 \end{BlockClass}
7011 }
```

Env `fminipage`  $[ \langle align \rangle ] [ \langle height \rangle ] [ \langle align \rangle ] \{ \langle width \rangle \}$

Creates a framed HTML `<div>` around its contents.

A print-output version is also supplied below.

```
7012 \NewDocumentEnvironment{fminipage}{0{t} o 0{t} m}
7013 {%
7014 \LWR@traceinfo{fminipage #1 #2 #3 #4}%
7015 \LWR@forceminwidth{\fboxrule}%
7016 \setlength{\LWR@tempwidth}{#4}%
7017 \IfValueTF{#2}{\setlength{\LWR@tempheight}{#2}}{}%
7018 \begin{BlockClass}[%
7019 \LWR@blackborderpadding ; %
7020 \uselengthunit{PT}%
7021 \IfValueTF{#2}{height:\rndprintlength{\LWR@tempheight} ; }{}%
7022 width:\rndprintlength{\LWR@tempwidth}%
7023 ]{fminipage}%
7024 }
7025 {%
7026 \end{BlockClass}%
7027 \LWR@traceinfo{fminipage done}%
7028 }
```

`\raisebox`  $\{ \langle raiselen \rangle \} [ \langle height \rangle ] [ \langle depth \rangle ] \{ \langle text \rangle \}$

```
7029 \LetLtxMacro{\LWR@origraisebox}{\raisebox}
7030
7031 \RenewDocumentCommand{\raisebox}{m o o m}{%
7032 #4%
```

```
7033 }
```

```
7034 \end{warpHTML}
```

**for HTML & PRINT:** 7035 \begin{warpall}

LWRprint@fminipage is defined inside warpall. For print output, it is \let to fminipage. For HTML output, the HTML version of fminipage is used instead, but the print version is still available for use inside a lateximage.

```
Env LWRprint@fminipage [\langle 1:align \rangle] [\langle 2:height \rangle] [\langle 3:inner-align \rangle] {\langle 4:width \rangle}
```

Creates a frame around its contents.

```
7036 \newsavebox{\LWR@fminipagebox}
7037
7038 \NewDocumentEnvironment{LWRprint@fminipage}{0{t} o 0{t} m}
7039 {%
```

An outer minipage will be used for vertical alignment. An inner minipage will be framed with \fbbox.

If the optional inner alignment is not given, use the outer instead:

```
7040 \IfValueTF{#3}%
7041 {\def\LWR@thisalign{#3}}
7042 {\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.

```
7043 \IfValueTF{#2}%
7044 {\minipage[#1][#2+2\fbboxsep+2\fbboxrule][\LWR@thisalign]{#4+2\fbboxsep+2\fbboxrule}}%
7045 {\minipage[#1]{#4+2\fbboxsep+2\fbboxrule}}%
```

Capture the contents of the environment:

```
7046 \begin{lrbox}{\LWR@fminipagebox}%
```

Nest the contents inside an inner minipage of the desired size:

```
7047 \IfValueTF{#2}%
7048 {\minipage[#1][#2][\LWR@thisalign]{#4}}%
7049 {\minipage[#1]{#4}}%
7050 }
7051 {%
```



Close the inner minipage and the LR box with the contents:

```
7052 \endminipage%
7053 \end{lrbox}%
```

Create a frame around the contents of the environment:

```
7054 \fbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```
7055 \endminipage%
7056 }
```

```
7057 \end{warppall}
```

**for PRINT output:** 7058 \begin{warpprint}

For print output, the following are \let to become active.

```
\fboxBlock  {\langle text \rangle}
```

Creates a framed HTML <div> around the text.

```
7059 \let\fboxBlock\fbox
```

```
Env  fminipage  [\langle align \rangle] [\langle height \rangle] [\langle align \rangle] {\langle width \rangle}
```

Creates a frame around its contents.

```
7060 \LetLtxMacro{fminipage}{\LWRprint@fminipage}
7061 \LetLtxMacro{endfminipage}{\endLWRprint@fminipage}
```

```
7062 \end{warpprint}
```

## 70 Direct formatting

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

For high-level block and inline custom CSS classes, see section [37.8](#).

**for HTML output:** 7063 \begin{warppHTML}

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

```
7064 \DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{#1}}
7065 \DeclareRobustCommand{\LWR@nullemph}[1]{#1}
7066 \LetLtxMacro{\emph}{\LWR@HTMLemph}
```

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

```
7067 \DeclareRobustCommand{\LWR@HTMLtextmd}[1]{\LWR@htmlspanclass{textmd}{#1}}
7068 \DeclareRobustCommand{\LWR@nulltextmd}[1]{#1}
7069 \LetLtxMacro{\textmd}{\LWR@HTMLtextmd}
```

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

```
7070 \DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{#1}}
7071 \DeclareRobustCommand{\LWR@nulltextbf}[1]{#1}
7072 \LetLtxMacro{\textbf}{\LWR@HTMLtextbf}
```

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

```
7073 \DeclareRobustCommand{\LWR@HTMLtextrm}[1]{\LWR@htmlspanclass{textrm}{#1}}
7074 \DeclareRobustCommand{\LWR@nulltextrm}[1]{#1}
7075 \LetLtxMacro{\textrm}{\LWR@HTMLtextrm}
```

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

```
7076 \DeclareRobustCommand{\LWR@HTMLtextsf}[1]{\LWR@htmlspanclass{textsf}{#1}}
7077 \DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
7078 \LetLtxMacro{\textsf}{\LWR@HTMLtextsf}
```

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

```
7079 \DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
7080 \DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
7081 \LetLtxMacro{\texttt}{\LWR@HTMLtexttt}
```

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

```
7082 \DeclareRobustCommand{\LWR@HTMLtextup}[1]{\LWR@htmlspanclass{textup}{#1}}
7083 \DeclareRobustCommand{\LWR@nulltextup}[1]{#1}
7084 \LetLtxMacro{\textup}{\LWR@HTMLtextup}
```

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

```
7085 \DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}}
```

```
7086 \DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
7087 \LetLtxMacro{\textit}{\LWR@HTMLtextit}
```

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

```
7088 \DeclareRobustCommand{\LWR@HTMLtextsc}[1]{\LWR@htmlspanclass{textsc}{#1}}
7089 \DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}
7090 \LetLtxMacro{\textsc}{\LWR@HTMLtextsc}
```

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

```
7091 \DeclareRobustCommand{\LWR@HTMLtextsl}[1]{\LWR@htmlspanclass{textsl}{#1}}
7092 \DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}
7093 \LetLtxMacro{\textsl}{\LWR@HTMLtextsl}
```

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

```
7094 \DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
7095 \DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
7096 \LetLtxMacro{\textnormal}{\LWR@HTMLtextnormal}
```

```
7097 \DeclareRobustCommand{\LWR@nullrmfamily}{%}
7098 \DeclareRobustCommand{\LWR@nullsfamily}{%}
7099 \DeclareRobustCommand{\LWR@nullttfamily}{%}
7100 \DeclareRobustCommand{\LWR@nullbfseries}{%}
7101 \DeclareRobustCommand{\LWR@nullmdseries}{%}
7102 \DeclareRobustCommand{\LWR@nullupshape}{%}
7103 \DeclareRobustCommand{\LWR@nullslshape}{%}
7104 \DeclareRobustCommand{\LWR@nullscshape}{%}
7105 \DeclareRobustCommand{\LWR@nullitshape}{%}
7106 \DeclareRobustCommand{\LWR@nullem}[1]{%}
7107 \DeclareRobustCommand{\LWR@nullnormalfont}{%}
```

`\LWR@nullfonts` Removes formatting during filename operations.

```
7108 \newcommand*{\LWR@nullfonts}{%}
7109 \LetLtxMacro{\emph}{\LWR@nullemph}%
7110 \LetLtxMacro{\textmd}{\LWR@nulltextmd}%
7111 \LetLtxMacro{\textbf}{\LWR@nulltextbf}%
7112 \LetLtxMacro{\textrm}{\LWR@nulltextrm}%
7113 \LetLtxMacro{\textsf}{\LWR@nulltextsf}%
7114 \LetLtxMacro{\texttt}{\LWR@nulltexttt}%
7115 \LetLtxMacro{\textup}{\LWR@nulltextup}%
7116 \LetLtxMacro{\textit}{\LWR@nulltextit}%
7117 \LetLtxMacro{\textsc}{\LWR@nulltextsc}%
7118 \LetLtxMacro{\textsl}{\LWR@nulltextsl}%
7119 \LetLtxMacro{\textnormal}{\LWR@nulltextnormal}%}
```

```

7120 \LetLtxMacro{\rmfamily}{\LWR@nullrmfamily}%
7121 \LetLtxMacro{\sffamily}{\LWR@nullsffamily}%
7122 \LetLtxMacro{\ttfamily}{\LWR@nullttfamily}%
7123 \LetLtxMacro{\bfseries}{\LWR@nullbfseries}%
7124 \LetLtxMacro{\mdseries}{\LWR@nullmdseries}%
7125 \LetLtxMacro{\upshape}{\LWR@nullupshape}%
7126 \LetLtxMacro{\slshape}{\LWR@nullslshape}%
7127 \LetLtxMacro{\scshape}{\LWR@nullscshape}%
7128 \LetLtxMacro{\itshape}{\LWR@nullitshape}%
7129 \LetLtxMacro{\em}{\LWR@nullem}%
7130 \LetLtxMacro{\normalfont}{\LWR@nullnormalfont}%
7131 \renewcommand*{\HTMLunicode}[1]{}%
7132 % \renewcommand*{\HTMLunicode}[1]{%
7133 % \renewcommand*{\HTMLentity}[1]{##1}
7134 \renewcommand*{\HTMLentity}[1]{}%

```

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

```

7135 \renewcommand*{\&}{and}%
7136 \renewcommand{\textsuperscript}[1]{##1}%
7137 \renewcommand{\textsubscript}[1]{##1}%
7138 \RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
7139 \DeclareExpandableDocumentCommand{\InlineClass}{o m +m}{##3}%
7140 }

```

`\mdseries`

```

7141 \renewcommand*{\mdseries}{}

```

`\bfseries`

```

7142 \renewcommand*{\bfseries}{}

```

`\rmfamily`

```

7143 \renewcommand*{\rmfamily}{}

```

`\sffamily`

```

7144 \renewcommand*{\sffamily}{}

```

`\ttfamily`

```

7145 \renewcommand*{\ttfamily}{}

```

`\upshape`

```
7146 \renewcommand*{\upshape}{}
```

`\itshape`

```
7147 \renewcommand*{\itshape}{}
```

`\scshape`

```
7148 \renewcommand*{\scshape}{}
```

`\normalfont`

```
7149 \renewcommand*{\normalfont}{} 
```

`\sp`  $\langle text \rangle$

For siunitx. Must work in math mode.

```
7150 \renewcommand{\sp}[1]{\text{<sup>#1</sup>}} 
```

`\sb`  $\langle text \rangle$

For siunitx. Must work in math mode.

```
7151 \renewcommand{\sb}[1]{\text{<sub>#1</sub>}} 
```

`\textsuperscript`  $\langle text \rangle$

```
7152 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

`\textsubscript`  $\langle text \rangle$

```
7153 \renewcommand{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
```

`\up`  $\langle text \rangle$  Prints superscript.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
7154 \AtBeginDocument{\let\up\textsuperscript}
```

`\fup`  $\{\langle text \rangle\}$  Prints superscript.

Supports `fntcount` package.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
7155 \AtBeginDocument{\let\fup\textsuperscript}
```

`\hfill`

```
7156 \renewcommand*{\hfill}{\quad}
```

`\hrulefill`

```
7157 \renewcommand*{\hrulefill}{\rule{1in}{1pt}}
```

`\dotfill`

```
7158 \renewcommand*{\dotfill}{\dots}
```

```
7159 \end{warpHTML}
```

## 71 Skips, spaces, font sizes

for HTML output: 7160 `\begin{warpHTML}`

`\`, must be redefined after `\RequirePackage{printlen}`

```
7161 \let\LWR@origcomma\,
7162 \let\LWR@origtilde~
7163 \let\LWR@origenskip\enskip
7164 \let\LWR@origquad\quad
7165 \let\LWR@origqqquad\qqquad
7166 \let\LWR@orighspace\hspace
7167 \let\LWR@origvspace\vspace
7168 \let\LWR@origrule\rule
7169 \let\LWR@origmedskip\medskip
```

Direct-formatting space commands become HTML entities:

```
7170 \renewcommand*{\,}{\HTMLunicode{202f}} % HTML thin non-breakable space
```

```
7171
```

```

7172 \renewcommand*{~}{\HTMLentity{nbsp}}
7173
7174 \renewcommand*{\textellipsis}{\HTMLunicode{2026}}

```

Direct-formatting font sizes are ignored:

```

7175 \let\LWR@orignormalsize\normalsize
7176 \let\LWR@origsmall\small
7177 \let\LWR@origfootnotesize\footnotesize
7178 \let\LWR@origscriptsize\scriptsize
7179 \let\LWR@origtiny\tiny
7180 \let\LWR@origlarge\large
7181 \let\LWR@origLarge\Large
7182 \let\LWR@origLARGE\LARGE
7183 \let\LWR@orighuge\huge
7184 \let\LWR@origHuge\Huge
7185 \renewcommand*{\normalsize}{}
7186 \renewcommand*{\small}{}
7187 \renewcommand*{\footnotesize}{}
7188 \renewcommand*{\scriptsize}{}
7189 \renewcommand*{\tiny}{}
7190 \renewcommand*{\large}{}
7191 \renewcommand*{\Large}{}
7192 \renewcommand*{\LARGE}{}
7193 \renewcommand*{\huge}{}
7194 \renewcommand*{\Huge}{}
7195
7196 \renewcommand*{\onecolumn}{}
7197
7198 \renewcommand{\twocolumn}[1][{}]{
7199
7200 #1
7201
7202 }

```

`\newline` Uses the HTML `<br />` element.

```

7203 \newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
7204 \let\newline\LWR@newlinebr

```

`\\` Redefined to `\LWR@endoffline` or `\LWR@tabularendoffline`.

`\LWR@endoffline` \* [*len*]

`\\` is assigned to `\LWR@endoffline` at `\LWR@LwarpStart`.

Inside `tabular`, `\\` is temporarily changed to `\LWR@tabularendoffline`.

```

7205 \LetLtxMacro\LWR@origendofline\
7206 \NewDocumentCommand{\LWR@endofline}{s o}
7207 {%
7208 \newline%
7209 }

```

**\LWR@minipagestartpars** Minipages are often placed side-by-side inside figures, with a bit of horizontal space to separate them. Since HTML does not allow a `<div>` to be inside a `p`, paragraphs must be turned off during the generation of the minipage, then turned on after the minipage is complete. When this occurs between side-by-side minipages, **lwarp** correctly suppresses the paragraph tags between the minipages, unless some other text is between the minipages. Such text forms its own paragraph, resulting in text after a minipage to be on its own line. Since people often place small horizontal space between minipages, it is desirable to maintain this space if possible. **lwarp** tries to do this by remembering that a minipage has been seen, in which case paragraph tags are suppressed around **\hspace**, **\enskip**, **\quad**, and **\qqquad** until the end of the paragraph, when the closing `p` tag is created.

**\hspace**  
**\enskip**  
**\quad**  
**\qqquad**

When a minipage is seen, the boolean **LWR@minipagethispar** is set, telling the following horizontal whitespace commands to try to suppress their surrounding paragraph tags. **LWR@minipagethispar** is cleared at the next end of paragraph, when the HTML paragraph closing tag is generated.

Placed just before **\hspace**, **\quad**, or **\qqquad**'s HTML output.

```

7210 \newcommand*{\LWR@minipagestartpars}{%
7211 \ifbool{LWR@minipagethispar}{\LWR@startpars}{}%
7212 }

```

**\LWR@minipagestoppars** Placed just after **\hspace**, **\quad**, or **\qqquad**'s HTML output.

```

7213 \newcommand*{\LWR@minipagestoppars}{%
7214 \ifbool{LWR@minipagethispar}{\LWR@stoppars}{}%
7215 }

```

**\quad** Handles special minipage & horizontal space interactions.

```

7216 \renewcommand*{\quad}{%
7217 \LWR@minipagestoppars%
7218 \HTMLUnicode{2001}%
7219 \LWR@minipagestartpars%
7220 }

```

**\qqquad** Handles special minipage & horizontal space interactions.

```

7221 \renewcommand*{\qqquad}{\quad\quad}

```



`\enskip` Handles special minipage & horizontal space interactions.

```
7222 \renewcommand*{\enskip}{%
7223 \LWR@minipagestoppars%
7224 \HTMLUnicode{2000}%
7225 \LWR@minipagestartpars%
7226 }
```

Len `\WR@tempwidth` Used to compute span width, height, raise for `\hspace` and `\rule`:

```
Len \WR@tempheight 7227 \newlength{\LWR@tempwidth}
Len \WR@tempraise 7228 \newlength{\LWR@tempheight}
7229 \newlength{\LWR@tempraise}
```

`\LWR@hspace` \*  $\langle length \rangle$

Handles special minipage & horizontal space interactions.

Prints a span of a given width. Ignores the optional star.

`\hspace{\fill}` is converted to `\hspace{2em}`, equal to `\qquad`.

```
7230 \NewDocumentCommand{\LWR@hspace}{s m}{%
7231 \setlength{\LWR@tempwidth}{#2}%
```

If `\fill`, change to `\qquad`:

```
7232 \ifnum\gluestretchorder\LWR@tempwidth>0%
7233 \setlength{\LWR@tempwidth}{2em}%
7234 \fi%
```

Only if the width is not zero:

```
7235 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{0pt}}{ }{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7236 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
7237 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{.16667em}}
7238 {%
7239 \HTMLUnicode{2009}% thin breakable space
7240 }%
```

Print the span with the converted width. Not rounded.

```

7241    {%
7242    \uselengthunit{PT}%
7243    \LWR@htmltagc{%
7244    span style="width:\printlength{\LWR@tempwidth}; display:inline-block;"%
7245    }%
7246    \LWR@htmltagc{/span}%
7247    }%

```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

7248    \LWR@minipagestartpars%
7249 }% width not 0
7250 }

```

`\LWR@nohspace` \*  $\langle length \rangle$

Used to disable `\hspace` while creating description `\items`.

```
7251 \NewDocumentCommand{\LWR@nohspace}{s m}{}
```

`\hspace` \*  $\langle length \rangle$

Handles special minipage & horizontal space interactions.

```
7252 \LetLtxMacro{\hspace}{\LWR@hspace}
```

`\LWR@vspace` \*  $\langle length \rangle$  Nullified `vspace`.

```
7253 \NewDocumentCommand{\LWR@vspace}{s m}{}

```

`\vspace` \*  $\langle length \rangle$  Nullified.

```
7254 \let\vspace\LWR@vspace

```

`\linebreak` [ $\langle num \rangle$ ] Inserts an HTML `br` tag.

```
7255 \renewcommand*{\linebreak}[1][]{\newline}

```

`\nolinebreak` [ $\langle num \rangle$ ]

```
7256 \renewcommand*{\nolinebreak}[1][]{}
```

`\pagebreak` [ $\langle num \rangle$ ] Starts a new paragraph.

```

7257 \renewcommand*{\pagebreak}[1][]{
7258
7259 }

```

`\nopagebreak` [*num*]

```

7260 \renewcommand*{\nopagebreak}[1][]{

```

`\enlargethispage` \* {*len*}

```

7261 \RenewDocumentCommand{\enlargethispage}{s m}{}

```

`\LWR@currenttextcolor` The color to use for text and `\rule`, defaulting to black:

```

7262 \newcommand*{\LWR@currenttextcolor}{black}

```

`\LWR@rule` [*raise*] {*width*} {*height*}

Handles special minipage & horizontal space interactions.

Creates a span of a given width and height. Ignores the optional star.

`\fill` is zero-width, so `\hspace{\fill}` is ignored.

```

7263 \NewDocumentCommand{\LWR@rule}{o m m}{%

```

The width is copied into a temporary L<sup>A</sup>T<sub>E</sub>X length, from which comparisons and conversions may be made:

```

7264 \setlength{\LWR@tempwidth}{#2}%

```

If it's zero-width then skip the entire rule:

```

7265 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}{
7266 {}}% zero- width
7267 {}% non-zero width

```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```

7268 \ifthenelse{%
7269 \lengthtest{\LWR@tempwidth>0pt}\AND%
7270 \lengthtest{\LWR@tempwidth<1pt}}%
7271 }%
7272 {\setlength{\LWR@tempwidth}{1pt}}{}%

```

Likewise with height:

```

7273     \setlength{\LWR@tempheight}{#3}%
7274     \ifthenelse{%
7275         \lengthtest{\LWR@tempheight>0pt}\AND%
7276         \lengthtest{\LWR@tempheight<1pt}%
7277     }%
7278     {\setlength{\LWR@tempheight}{1pt}}{}%
```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```

7279     \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<sup>A</sup>T<sub>E</sub>X code.

```

7280     \uselengthunit{PT}%
7281     \LWR@htmltagc{%
7282         span
7283         style="%
```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```

7284     background:\LWR@currenttextcolor; %
```

The width and height are printed, converted to PT:

```

7285     width:\printlength{\LWR@tempwidth}; %
7286     height:\printlength{\LWR@tempheight}; %
```

The raise height is converted to a CSS transform. The `*2` raise multiplier is to approximately match HTML output's X height. Conversion to a L<sup>A</sup>T<sub>E</sub>X length allows a typical L<sup>A</sup>T<sub>E</sub>X expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a L<sup>A</sup>T<sub>E</sub>X length limits the allowable syntax. To do: A superior method would compute a ratio of L<sup>A</sup>T<sub>E</sub>X ex height, then print that to HTML with an ex unit.

```

7287     \IfValueTF{#1}%
7288     {%
7289         \setlength{\LWR@tempraise}{0pt-#1}%
7290         \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
7291         \LWR@orignewline%
7292         -ms-transform: translate(0pt,\printlength{\LWR@tempraise}); %
7293         \LWR@orignewline%
7294         -webkit-transform: translate(0pt,\printlength{\LWR@tempraise}); %
7295         \LWR@orignewline%
```

```

7296         transform: translate(0pt,\printlength{\LWR@tempraise}); %
7297         \LWR@orignewline%
7298     }{}%

```

Display inline-block to place the span inline with the text:

```

7299     display:inline-block;"%
7300 }%
7301 \LWR@htmltagc{/span}%

```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

7302     \LWR@minipagestartpars%
7303 }% non-zero width
7304 }

```

```
\rule [⟨raise⟩] {⟨width⟩} {⟨height⟩}
```

Handles special minipage & horizontal space interactions.

```

7305 \renewcommand{\rule}{\LWR@rule}
7306 \end{warpHTML}

```

## 72 \phantomsection

for HTML output: 7307 \begin{warpHTML}

`\phantomsection` Emulate the hyperref `\phantomsection` command, often used to insert the bibliography into table of contents:

```

7308 \newcommand*{\phantomsection}{\section*{}}
7309 \end{warpHTML}

```

## 73 \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>

## 73.1 HTML logos

for HTML output: 7310 \begin{warpHTML}

\TeX     $\mathrm{T}_{\mathrm{E}}\mathrm{X}$

latexlogo is a CSS class used to properly typeset the E and A in  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$  and friends.

latexlogofont is a CSS class used to select the font for the rest of the logo in  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}_{\mathrm{E}}$ ,  $\mathrm{C}^{\mathrm{O}}\mathrm{n}\mathrm{T}_{\mathrm{E}}\mathrm{X}_{\mathrm{t}}$ , etc.

```
7311 \newcommand*{\LWR@TeX}
7312 {\InlineClass{latexlogofont}%
7313 {\InlineClass{latexlogo}{T\textsubscript{e}X}}}
```

\LaTeX     $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}_{2\epsilon}$

\LaTeXe

```
7314 \newcommand*{\LWR@LaTeX}
7315 {\InlineClass{latexlogofont}%
7316 {\InlineClass{latexlogo}%
7317 {L\textsuperscript{a}T\textsubscript{e}X}}
7318
7319 \renewcommand*{\LaTeXe}
7320 {\LaTeX\InlineClass{latexlogofont}%
7321 {\,\,2\textsubscript{\textit{\HTMLUnicode{3B5}}}}}
```

\LuaTeX     $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}_{\mathrm{E}}$

\LuaLaTeX

```
7322 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
7323 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

\XeTeX     $\mathrm{X}_{\mathrm{E}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{X}_{\mathrm{E}}\mathrm{T}_{\mathrm{E}}\mathrm{X}_{\mathrm{E}}$

\XeLaTeX

xetexlogo is a CSS class which aligns the backwards E in  $\mathrm{X}_{\mathrm{E}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$  and spaces  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  appropriately.

xelatexlogo is a CSS class which aligns the backwards E in  $\mathrm{X}_{\mathrm{E}}\mathrm{T}_{\mathrm{E}}\mathrm{X}_{\mathrm{E}}$  and spaces  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$  appropriately.

```
7324 \newcommand*{\Xe}
7325 {X\textsubscript{\HTMLUnicode{18e}}}
```

```

7326 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
7327 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}

```

`\ConTeXt`   `ConTEXt`

```

7328 \newcommand*{\LWR@ConTeXt}
7329 {\InlineClass{latexlogofont}{Con}\TeX{}}%
7330 \InlineClass{latexlogofont}{t}}

```

`\BibTeX`   `BIBTEX`, *MakeIndex*  
`\MakeIndex`

```

7331 \providecommand*{\BibTeX}
7332 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}
7333
7334 \newcommand*{\MakeIndex}
7335 {\InlineClass{latexlogofont}{\textit{MakeIndex}}}

```

`\AmS`   `AMS`

`amslogo` is a CSS class used for the `AMSlogo`.

```

7336 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}
7337 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}

```

`\MiKTeX`   `MiKTEX`

```

7338 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}

```

`\LyX`   `LyX`

`lyxlogo` is a CSS class used for the `LyXlogo`.

```

7339 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}

7340 \end{warpHTML}

```

## 73.2 Print logos

for PRINT output:

```

7341 \begin{warpprint}
7342 \newcommand*{\XeTeXrevE}
7343 {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
7344 \providecommand*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
7345 \providecommand*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
7346 \providecommand*{\AmS}{%

```

```

7347 \leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex%
7348 \hbox{$\mathcal M$\kern-.2em\mathcal S$}}
7349 \newcommand*\LyX{\textsf{LyX}}
7350 \providecommand*\LuaTeX{\mbox{Lua\TeX}}
7351 \providecommand*\LuaLaTeX{\mbox{Lua\LaTeX}}
7352 \providecommand*\BibTeX{\mbox{B\textsc{ib}\TeX}}
7353 \providecommand*\MakeIndex{\mbox{\textit{MakeIndex}}}
7354 \providecommand*\ConTeXt{\mbox{Con\TeXt}}
7355 \providecommand*\MiKTeX{\mbox{MiK\TeX}}
7356 \end{warpprint}

```

## 74 \AtBeginDocument, \AtEndDocument

for HTML output: 7357 \begin{warppHTML}

\LWR@LwarpStart Automatically sets up the HTML-related actions for the start and end of the  
 \LWR@LwarpEnd document.

```

7358 \AfterEndPreamble{\LWR@LwarpStart}
7359 \AtEndDocument{\LWR@LwarpEnd}

7360 \end{warppHTML}

```



## 75 Trademarks

- $\text{\TeX}$  is a trademark of American Mathematical Society.
- Adobe® and Adobe Framemaker® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.
- Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Mac OS® is a trademark of Apple Inc.
- MadCap Flare™ is the property of MadCap Software, Inc.
- MathJax is copyright 2009 and later. The MathJax Consortium is a joint venture of the American Mathematical Society (AMS) and the Society for Industrial and Applied Mathematics (SIAM) to advance mathematical and scientific content on the web.
- Microsoft®, Encarta, MSN, and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- UNIX® is a registered trademark of The Open Group.

## Package 2

# lwarp-abstract.sty

## 76 Abstract

*(Based on original code by PETER WILSON.)*

`Pkg abstract` abstract is supported and patched by lwarp.

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

for HTML output: Accept all options for lwarp-abstract:

```

1 \LWR@ProvidesPackagePass{abstract}

2 \AtBeginDocument{
3 \BeforeBeginEnvironment{abstract}{
4 \LWR@forcenewpage
5 \BlockClass{abstract}
6 }
7 \AfterEndEnvironment{abstract}{\endBlockClass}
8 }
9
10 \renewcommand{\@bsrunintitle}{%
11 \hspace*{\abstitleskip}%
12 {\abstractnamefont%
13 \InlineClass{abstractrunintitle}{\abstractname}%
14 \@bslabeldelim}%
15 }
16
17 \if@titlepage
18 \renewenvironment{abstract}{%
19 % \titlepage
20 \null\vfil
21 \@beginparpenalty\@lowpenalty
22 \if@bsrunin
23 \else
24 \if@bsstyle
25 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
26 \else
27 \ifnumber@bs
28 \num@bs

```

---

```

29         \else
30         \begin{\absnamepos}%
31 \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
32         \endparpenalty\@M
33         \end\absnamepos%
34 %%         \vspace{\abstitleskip}%
35         \fi
36     \fi
37     \vspace{\abstitleskip}%
38 \fi
39 \put@bsintoc%
40 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
41 {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
42 }
43 \else
44 \renewenvironment{abstract}{%
45 \if@bsrunin
46 \else
47 \if@bsstyle
48 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
49 \else
50 \ifnumber@bs
51 \num@bs
52 \else
53 \begin{\absnamepos}%
54 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
55 \end\absnamepos%
56 %%         \vspace{\abstitleskip}%
57         \fi
58     \fi
59     \vspace{\abstitleskip}%
60 \fi
61 \put@bsintoc%
62 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
63 {\par\end{@bstr@ctlist}}
64 \fi
65

```

## Package 3

# lwarp-afterpage.sty

## 77 Afterpage

Pkg afterpage Emulated.

for **HTML output**: Discard all options for lwarp-afterpage:

```
1 \LWR@ProvidesPackageDrop{afterpage}
```

```
2 \newcommand{\afterpage}[1]{#1}
```

## Package 4

# lwarp-algorithmicx.sty

## 78 Algorithmicx

Pkg algorithmicx algorithmicx is supported with minor adjustments.

for HTML output: 1 \LWR@ProvidesPackagePass{algorithmicx}

Inside the `algorithmic` environment, level indenting is converted to a `<span>` of the required length, and comments are placed inside a `<span>` which is floated right.

 package conflicts If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section [187.1](#).

for HTML output: 2 \begin{warpHTML}

```

3 \AtBeginEnvironment{algorithmic}{%
4 %
5 \let\origALG@doentity\ALG@doentity%
6 %
7 \renewcommand*{\ALG@doentity}{%
8 \origALG@doentity%
9 \uselengthunit{PT}%
10 \LWR@htmltagc{%
11 span style="width:\rndprintlength{\ALG@thistlm}; display:inline-block;"%
12 }%
13 \LWR@htmltagc{/span}%
14 }%
15 %
16 \let\origComment\Comment%
17 %
18 \renewcommand{\Comment}[1]{\InlineClass{floatright}{\origComment{#1}}}%
19 }

20 \end{warpHTML}
```

## Package 5

# lwarp-alltt.sty

## 79 Alltt

Pkg alltt alltt is patched for use by lwarp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{alltt}

2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching alltt.}
4 \AtBeginEnvironment{alltt}{%
5 \LWR@forcenewpage
6 \LWR@atbeginverbatim{alltt}\unskip\LWR@origvspace*{-\baselineskip}%
7 }
8 \AfterEndEnvironment{alltt}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
9 }
```

## Package 6

# lwarp-amsthm.sty

## 80 AMSthm

(Based on original code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)

Pkg amsthm amsthm is patched for use by lwarp.

### CSS styling of theorems and proofs:

**Theorem:** <div> of class amsthmbody<theoremstyle>  
**Theorem Name:** <span> of class amsthmname<theoremstyle>  
**Theorem Number:** <span> of class amsthmnumber<theoremstyle>  
**Theorem Note:** <span> of class amsthmnote<theoremstyle>  
**Proof:** <div> of class amsthmproof  
**Proof Name:** <span> of class amsthmproofname  
 where <theoremstyle> is plain, definition, etc.

for HTML output: 1 \LWR@ProvidesPackagePass{amsthm}

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
3 \renewcommand{\theoremstyle}[1]{%
4   \@ifundefined{th@#1}{%
5     \PackageWarning{amsthm}{Unknown theoremstyle '#1'}%
6     \thm@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% new
8   }{%
9     \thm@style{#1}%
10    \renewcommand{\LWR@newtheoremstyle}{#1}% new
11  }%
12 }
```

Patched to remember the style for this theorem type:

```

13 \def\xnthm#1#2{%
14   \csedef{LWR@thmstyle#2}{\LWR@newtheoremstyle}% new
15   \let\@tempa\relax
16   \exp\@ifdefinable\csname #2\endcsname{%
17     \global\exp\let\csname end#2\endcsname\@endtheorem
18     \ifx *#1% unnumbered, need to get one more mandatory arg
19       \edef\@tempa##1{%
20         \gdef\exp\@nx\csname#2\endcsname{%
21           \@nx\@thm{\exp\@nx\csname th@\the\thm@style\endcsname}%
22           }\{##1}\}%
23       \else % numbered theorem, need to check for optional arg
24         \def\@tempa{\@oparg{\ynthm{#2}}{}}%
25       \fi
26       \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#2}}}% new
27   }%
28   \@tempa
29 }

```

Patched to enclose with CSS:

```

30 \newcommand{\LWR@haveamsthmname}{
31 \renewcommand{\thmname}[1]{\InlineClass{amsthmname\LWR@thisthmstyle}{##1}}
32 }
33
34 \newcommand{\LWR@haveamsthmnumber}{
35 \renewcommand{\thmnumber}[1]{\InlineClass{amsthmnumber\LWR@thisthmstyle}{##1}}
36 }
37
38 \newcommand{\LWR@haveamsthmnote}{
39 \renewcommand{\thmnote}[1]{\InlineClass{amsthmnote\LWR@thisthmstyle}{##1}}
40 }
41
42 \LWR@haveamsthmname
43 \LWR@haveamsthmnumber
44 \LWR@haveamsthmnote

```

Patches for CSS:

```

45 \def\@begintheorem#1#2[#3]{%
46 \LWR@forcenewpage% new
47 \BlockClass{amsthmbody\LWR@thisthmstyle}% new
48 \deferred@thm@head{
49   \the\thm@headfont \thm@indent
50   \@ifempty{#1}{\let\thmname\@gobble}{\LWR@haveamsthmname}% new
51   \@ifempty{#2}{\let\thmnumber\@gobble}{\LWR@haveamsthmnumber}% new
52   \@ifempty{#3}{\let\thmnote\@gobble}{\LWR@haveamsthmnote}% new
53   \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
54   \the\thm@headpunct~
55   \thmheadnl % possibly a newline.
56   \hskip\thm@headsep

```



```

57 }%
58 \ignorespaces}

```

Patched for CSS:

```

59 \def\@endtheorem{\endBlockClass\endtrivlist\@endpefalse }

```

Proof QED symbol:

```

60 \AtBeginDocument{
61 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
62 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
63 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
64 }

```

Patched for CSS:

```

65 \renewenvironment{proof}[1][\proofname]{\par
66 \LWR@forcenewpage% new
67   \BlockClass{amsthmproof}% new
68   \pushQED{\qed}%
69   \normalfont \topsep6\p@\@plus6\p@\relax
70   \trivlist
71   \item[\hskip\labelsep
72     \InlineClass{amsthmproofname}{#1\@addpunct{.}}]\ignorespaces% changes
73 }{%
74   \InlineClass{theoremdendmark}{\popQED}\endtrivlist%
75   \endBlockClass% new
76   \@endpefalse
77 }

```

## Package 7

# lwarp-balance.sty

## 81 Balance

Pkg    **balance**    Emulated.

**for HTML output:**    Discard all options for lwarp-balance:

```
1 \LWR@ProvidesPackageDrop{balance}
```

```
2 \newcommand*{\balance}{}
3 \newcommand*{\nobalance}{}

```

## Package 8

# lwarp-bookmark.sty

## 82 Bookmark

Pkg bookmark bookmark is emulated during HTML output, and the bookmark package is ignored.

for HTML output: Discard all options for lwarp-bookmark:

```
1 \LWR@ProvidesPackageDrop{bookmark}

2 \newcommand*{\bookmarksetup}[1]{}
3 \newcommand*{\bookmarksetupnext}[1]{}
4 \newcommand*{\bookmark}[2] [] {}
5 \newcommand*{\bookmarkdefinestyle}[2] {}
6 \newcommand*{\bookmarkget}[1] {}
7 \newcommand{\BookmarkAtEnd}[1] {}
```

## Package 9

# lwarp-booktabs.sty

## 83 Booktabs

Pkg booktabs booktabs is emulated during HTML output, and used as-is during print output and inside an HTML lateximage.

for HTML output: 1 \LWR@ProvidesPackagePass{booktabs}

Booktabs emulation is spread among the tabular code. The original definitions are saved here for use in HTML lateximages. The HTML versions temporarily overwrite these print versions when tabular is started.

```
2 \LetLtxMacro\LWR@origtoprule\toprule
3 \LetLtxMacro\LWR@origmidrule\midrule
4 \LetLtxMacro\LWR@origcmidrule\cmidrule
5 \LetLtxMacro\LWR@origbottomrule\bottomrule
6 \LetLtxMacro\LWR@origaddlinespace\addlinespace
7 \LetLtxMacro\LWR@origmorecmidrules\morecmidrules
8 \LetLtxMacro\LWR@origspecialrule\specialrule
```

## Package 10

# lwarp-boxedminipage.sty

## 84 Boxedminipage

Pkg boxedminipage boxedminipage is superceded by boxedminipage2e.  
for HTML output: 1 \LWR@loadnever{boxedminipage}{boxedminipage2e}

## Package 11

# lwarp-boxedminipage2e.sty

## 85 Boxedminipage2e

Pkg boxedminipage2e boxedminipage2e is emulated.  
for HTML output: Discard all options for lwarp-boxedminipage2e:  
1 \LWR@ProvidesPackageDrop{boxedminipage2e}  
  
2 \newenvironment{boxedminipage}{%  
3 \begin{BlockClass}{framebox}%  
4 \minipage%  
5 }  
6 {  
7 \endminipage%  
8 \end{BlockClass}  
9 }

## Package 12

# lwarp-ccaption.sty

## 86 Ccaption

Pkg ccaption ccaption is not used. The user is recommended to use caption instead.

for HTML output: 1 \LWR@loadnever{ccaption}{caption}

## Package 13

# lwarp-changepage.sty

## 87 Changepage

Pkg changepage changepage is ignored.

for HTML output: Discard all options for lwarp-changepage:

```
1 \LWR@ProvidesPackageDrop{changepage}

2 \newif\ifoddpage
3 \DeclareRobustCommand{\checkoddpage}{\oddpagetrue}
4 \DeclareRobustCommand{\changetext}[5]{}
5 \DeclareRobustCommand{\changepage}[9]{}
6 \newenvironment{adjustwidth}[2]{}{}
7 \newenvironment{adjustwidth*}[2]{}{}
```

## Package 14

# lwarp-crop.sty

## 88 Crop

Pkg crop Emulated.

for HTML output: Discard all options for lwarp-crop:

```
1 \LWR@ProvidesPackageDrop{crop}
```

```
2 \newcommand*{\crop}[1] [] {}
```

```
3 \newcommand*{\cropdef}[6] [] {}
```

## Package 15

# lwarp-cutwin.sty

## 89 Cutwin

Pkg cutwin Emulated.

for HTML output: Discard all options for lwarp-cutwin:

```

1 \LWR@ProvidesPackageDrop{cutwin}

2 \newcommand*{\opencutleft}{\}
3 \newcommand*{\opencutright}{\}
4 \newcommand*{\opencutcenter}{\}
5 \newcommand*{\cutfuzz}{\}
6
7 \newenvironment{cutout}[4]
8 {\marginpar{\windowpagestuff}}
9 {}
10
11 \newcommand*{\windowpagestuff}{\}
12
13 \newcommand*{\pageinwindow}{\%
14 % \begin{minipage}{.3\linewidth}
15 \windowpagestuff
16 % \end{minipage}
17 }
18
19 \newenvironment{shapedcutout}[3]
20 {\marginpar{\picinwindow}}
21 {}
22
23 \newcommand*{\putstuffinpic}{\}
24
25 \newcommand*{\picinwindow}{\%
26 \begin{picture}(0,0)
27 \putstuffinpic
28 \end{picture}}
```

## Package 16

# lwarp-dcolumn.sty

## 90 Dcolumn

Pkg dcolumn dcolumn is emulated during HTML output, and the dcolumn package is ignored.

```
1 \LWR@ProvidesPackageDrop{dcolumn}
```



## Package 17

# lwarp-draftwatermark.sty

## 91 Draftwatermark

Pkg `draftwatermark` `draftwatermark` is emulated during HTML output, and the `draftwatermark` package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{draftwatermark}

2 \newcommand{\SetWatermarkAngle}[1]{}
3 \newcommand{\SetWatermarkColor}[1]{}
4 \newcommand{\SetWatermarkLightness}[1]{}
5 \newcommand{\SetWatermarkFontSize}[1]{}
6 \newcommand{\SetWatermarkScale}[1]{}
7 \newcommand{\SetWatermarkHorCenter}[1]{}
8 \newcommand{\SetWatermarkVertCenter}[1]{}
9 \newcommand{\SetWatermarkText}[1]{}

```

## Package 18

# lwarp-ellipsis.sty

## 92 Ellipsis

Pkg `ellipsis` `ellipsis` is emulated during HTML output, and the `ellipsis` package is ignored.

```
1 \LWR@ProvidesPackageDrop{ellipsis}
2
3 \newcommand{\ellipsisgap}{0.1em}

```

## Package 19

# lwarp-emptypage.sty

## 93 Emptypage

Pkg emptypage emptypage is ignored.

for **HTML output**: Discard all options for lwarp-emptypage:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

## Package 20

# lwarp-endnotes.sty

## 94 Endnotes

*(Based on original code by JOHN LAVAGNINO.)*

Pkg endnotes

```
for HTML output: 1 \LWR@ProvidesPackagePass{endnotes}

2 \def\enoteformat{%
3 % \rightskip\z@ \leftskip\z@ \parindent=1.8em
4 \leavevmode
5 % \llap{
6 \makeenmark
7 % }
8 }
9
10 \def\@makeenmark{\hbox{\LWR@htmlspan{sup}{\normalfont\theenmark}}}
11 \def\makeenmark{\@makeenmark}
```

## Package 21

# lwarp-enumerate.sty

## 95 Enumerate

Pkg **enumerate** **enumerate** is supported with no changes.

This package is only required because it was used in the past to drop and then emulate the package. It cannot be removed because an older version which dropped the package may still remain, for example in a local vs. distribution directory, but it is now supported directly by **lwarp** and thus must no longer be dropped.

for HTML output: 1 \LWR@ProvidesPackagePass{enumerate}

## Package 22

# lwarp-enumitem.sty

## 96 Enumitem

Pkg `enumitem` `enumitem` is supported with minor adjustments.

for HTML output: `1 \LWR@ProvidesPackagePass{enumitem}`

for HTML output: `2 \begin{warpHTML}`

```
\newlist {<name>} {<type>} {<maxdepth>}
\renewlist {<name>} {<type>} {<maxdepth>}
```

For `enumitem` lists, new lists must have the start and end actions assigned to the new environment. Renewed lists already have their actions assigned, and thus need no changes.

```
3 \let\LWR@orignewlist\newlist
4
5 \renewcommand*{\newlist}[3]{%
6 \LWR@orignewlist{#1}{#2}{#3}%
7 \AtBeginEnvironment{#1}{\csuse{LWR@#2start}}}%
8 \AtEndEnvironment{#1}{\csuse{LWR@#2end}}}%
9 }

10 \end{warpHTML}
```

## Package 23

# lwarp-epigraph.sty

## 97 Epigraph

Pkg epigraph epigraph is emulated during HTML output, and the epigraph package is ignored.

```
for HTML output: 1 \LWR@ProvidesPackageDrop{epigraph}

2 \newcommand{\qitem}[2]
3 {
4 \begin{BlockClass}{qitem}
5 #1
6 \begin{BlockClass}{epigraphsource}
7 #2
8 \end{BlockClass}
9 \end{BlockClass}
10 }

11 \newcommand{\epigraph}[2]
12 {
13 \begin{BlockClass}{epigraph}
14 \qitem{#1}{#2}
15 \end{BlockClass}
16 }
17
18 \newenvironment*{epigraphs}
19 {\BlockClass{epigraph}}
20 {\endBlockClass}
```

Use CSS to format epigraphs.

The following are null commands for source compatibility:

```
21 \newlength{\epigraphwidth}
22 \setlength{\epigraphwidth}{.5\linewidth}
23 \newenvironment*{flushepinormal}{}{}
24 \newcommand{\textflush}[1]{flushepinormal}
25 \newcommand{\epigraphflush}[1]{flushright}
26 \newcommand{\sourceflush}[1]{flushright}
27 \newcommand*{\epigraphsize}{\small}
28 \newlength{\epigraphrule}
29 \newlength{\beforeepigraphskip}
30 \newlength{\afterepigraphskip}
31 \newcommand{\epigraphhead}[2][0]{#2}
```

```

32 \newcommand{\dropchapter}[1]{}
33 \newcommand*{\undodrop}{}
34 \newcommand{\cleartoevenpage}[1] [] {}

```

## Package 24

# lwarp-eso-pic.sty

## 98 Eso-pic

Pkg eso-pic eso-pic is emulated during HTML output, and the eso-pic package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{eso-pic}

2 \newcommand*{\LenToUnit}{}
3 \newcommand{\AtPageUpperLeft}[1]{}
4 \newcommand{\AtPageLowerLeft}[1]{}
5 \newcommand{\AtPageCenter}[1]{}
6 \newcommand{\AtStockLowerLeft}[1]{}
7 \newcommand{\AtStockUpperLeft}[1]{}
8 \newcommand{\AtStockCenter}[1]{}
9 \newcommand{\AtTextUpperLeft}[1]{}
10 \newcommand{\AtTextLowerLeft}[1]{}
11 \newcommand{\AtTextCenter}[1]{}
12 \NewDocumentCommand{\AddToShipoutPictureBG}{s +m}{}
13 \renewcommand{\AddToShipoutPicture}{\AddToShipoutPictureBG}
14 \NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
15 \newcommand*{\ClearShipoutPictureBG}{}
16 \newcommand*{\ClearShipoutPicture}{}
17 \newcommand*{\ClearShipoutPictureFG}{}
18 \newcommand{\gridSetup}[6] [] {}

```

## Package 25

# lwarp-everypage.sty

## 99 Everypage

Pkg `everypage` `everypage` is emulated during HTML output, and the `everypage` package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{everypage}

2 \newcommand*{\AddEverypageHook}[1]{}
3 \newcommand*{\AddThispageHook}[1]{}

```



## Package 26

# lwarp-everyshi.sty

## 100 Everyshi

Pkg everyshi Emulated.

for HTML output: Discard all options for lwarp-everyshi:

```
1 \LWR@ProvidesPackageDrop{everyshi}

2 \newcommand*{\EveryShipout}[1]{}
3 \newcommand*{\AtNextShipout}[1]{}

```

## Package 27

# lwarp-extramarks.sty

## 101 Extramarks

Pkg extramarks extramarks is not used.

for HTML output: Discard all options for lwarp-extramarks:

```
1 \LWR@ProvidesPackageDrop{extramarks}

2 \newcommand*{\extramarks}[2]{}
3 \newcommand*{\firstleftxmark}{}
4 \newcommand*{\lastleftxmark}{}
5 \newcommand*{\firstrightxmark}{}
6 \newcommand*{\lastrightxmark}{}
7 \newcommand*{\firstxmark}{}
8 \newcommand*{\lastxmark}{}
9 \newcommand*{\topxmark}{}
10 \newcommand*{\topleftxmark}{}
11 \newcommand*{\firstleftmark}{}
12 \newcommand*{\lastrightmark}{}

```

## Package 28

# lwarp-fancybox.sty

## 102 Fancybox

(Based on original code by TIMOTHY VAN ZANDT.)

Pkg **fancybox** fancybox is supported with some patches.

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

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

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

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

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

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

```

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

```

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

for HTML output: `1 \begin{warpHTML}`

```

2 \LWR@ProvidesPackagePass{fancybox}

3 \renewcommand*{\@shadowbox}[1]{\InlineClass{shadowbox}{#1}}
4
5 \renewcommand*{\@doublebox}[1]{\InlineClass{doublebox}{#1}}
6
7 \renewcommand*{\@ovalbox}[2]{%
8 \ifthenelse{\isequivalentto{#1}{\thinlines}}
9 {\InlineClass{ovalbox}{#2}}
10 {\InlineClass{ovalbox}{#2}}
11 }

```

Convert minipages, parboxes, and lists into linear text using the `LWR@nestspan` environment:

```

12 \let\LWR@origSbox\Sbox
13
14 \def\Sbox{\LWR@origSbox\LWR@nestspan}
15
16
17 \let\LWR@origendSbox\endSbox
18
19 \def\endSbox{\endLWR@nestspan\LWR@origendSbox}

```

`Beqnarray` is adapted for MathJax or enclosed inside a `lateximage`:

```

20 \RenewEnviron{Beqnarray}
21 {\LWR@eqnarrayfactor}
22
23 \csgpreto{Beqnarray*}{\boolfalse{LWR@numbereqnarray}}

```

`\GenericCaption` is enclosed in an HTML block:

```

24 \renewcommand{\GenericCaption}[1]{%

```

```

25 \LWR@htmlblocktag{figcaption}%
26 #1%
27 \LWR@htmlblocktag{/figcaption}%
28 }

```

Btrivlist is enclosed in an HTML block:

```

29 \RenewDocumentEnvironment{Btrivlist}{m o}
30 {\begin{BlockClass}{Btrivlist}\tabular{#1}}
31 {\endtabular\end{BlockClass}}

```

Btrivlist is also neutralized when used inside a span:

```

32 \AtBeginEnvironment{LWR@nestspan}{%
33 \RenewDocumentEnvironment{Btrivlist}{m o}{-}{-}%
34 \RenewDocumentCommand{\LWR@origitem}{d()}{\LWRFB@origitem}%
35 }

```

lwarp's handling of \item is patched to accept fancybox's optional arguments:

```

36 \let\LWRFB@origitemizeitem\LWR@itemizeitem
37 \let\LWRFB@origdescitem\LWR@descitem
38 \LetLtxMacro{\LWRFB@origitem}{\LWR@origitem}
39
40 \RenewDocumentCommand{\LWR@itemizeitem}{d()}{\LWRFB@origitemizeitem}
41 \RenewDocumentCommand{\LWR@descitem}{d()}{\LWRFB@origdescitem}

```

The various boxed lists become regular lists:

```

42 \renewenvironment{Bitemize}[1] [] {\begin{itemize}}{\end{itemize}}
43 \renewenvironment{Benumerate}[1] [] {\begin{enumerate}}{\end{enumerate}}
44 \renewenvironment{Bdescription}[1] [] {\begin{description}}{\end{description}}

```

\boxput simply prints one then the other argument, side-by-side instead of above and behind:

```

45 \RenewDocumentCommand{\boxput}{s d() m m}{%
46 \IfBooleanTF{#1}{#3\quad#4}{#4\quad#3}%
47 }

```

Neutralized commands:

```

48 \RenewDocumentCommand{\fancyput}{s d() m}{-}
49 \RenewDocumentCommand{\thisfancyput}{s d() m}{-}
50
51 \RenewDocumentCommand{\fancypage}{m m}{-}
52 \RenewDocumentCommand{\thisfancypage}{m m}{-}

```

```

53
54 \def\LandScape#1{}
55 \def\endLandScape{}
56 \def\@Landscape#1#2#3{}
57 \def\endLandscape{}

```

Low-level patches for Verbatim, VerbatimInput, UseVerbatim:

```

58 \let\LWRFB@UseVerbatim\UseVerbatim
59 \renewcommand*{\UseVerbatim}[1]{%
60 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
61 \LWRFB@UseVerbatim{#1}%
62 \LWR@afterendverbatim%
63 }
64
65 \let\LWRFB@LUseVerbatim\LUseVerbatim
66
67 \renewcommand*{\LUseVerbatim}[1]{%
68 \LWR@atbeginverbatim{LVerbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
69 \LWRFB@LUseVerbatim{#1}%
70 \LWR@afterendverbatim%
71 }
72
73 \def\@BUseVerbatim[#1]#2{%
74 \LWR@atbeginverbatim{BVerbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
75 \LWRFB@UseVerbatim{#2}%
76 \LWR@afterendverbatim%
77 }
78 \end{warpHTML}

```

## Package 29

# lwarp-fancyhdr.sty

## 103 Fancyhdr

Pkg fancyhdr fancyhdr is nullified.

for HTML output: Discard all options for lwarp-fancyhdr:

```

1 \LWR@ProvidesPackageDrop{fancyhdr}

2 \newcommand*{\fancyhead}[2][{}]{

```

---

```
3 \newcommand*{\fancyfoot}[2] [] {}
4 \newcommand*{\fancyhf}[2] [] {}
5 \newcommand*{\fancyhead}[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}
```

## Package 30

# lwarp-fancyvrb.sty

## 104 Fancyvrb

(Based on original code by TIMOTHY VAN ZANDT.)

**Pkg fancyvrb** fancyvrb is supported with some patches.

**for HTML output:** 1 \RequirePackage{xcolor}% for \convertcolorspec  
2 \LWR@ProvidesPackagePass{fancyvrb}

**for HTML output:** 3 \begin{warpHTML}

Initial default patch for fancyvrb:

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

For \VerbatimFootnotes:

```
5 \renewcommand{\VerbatimFootnotes}{
6 \PackageError{lwarp}
7 {Verbatim footnotes are not yet supported by lwarp.}
8 {This may be improved some day.}
9 }
```

After the preamble is loaded, after any patches to Verbatim:

```
10 \AfterEndPreamble{
11 \LWR@traceinfo{Patching Verbatim.}
```

Remember the original defintion of Verbatim:

```
12 \let\LWRFV@origVerbatim\Verbatim
```

**Env Verbatim** Patched to place the environment in a **fancyvrb** div, and the label in a **fancyvrblabel** div. Also corrects the left margin for line numbers. Also uses **VerbatimHTMLWidth** to control placement of line numbers on the right. Aligning the right margin requires knowing the width.

```
13 \renewcommand*{\Verbatim}{%
14 \LWR@forcenewpage
15 \LWRFV@origVerbatim%
16 }
```

`\LWR@FVstyle` Holds the style of the verbatim.

```
17 \newcommand*{\LWR@FVstyle}{}
```

The following patches to `Verbatim` are executed at the start and end of the environment, depending on the choice of `frame`. Original code is from the `fancyvrb` package.

```
18 \newcommand*{\LWR@fvstartnone}{%
19 \LWR@traceinfo{fvstartnone}%
20 \hbox to\z@{\LWR@atbeginverbatim[\LWR@FVstyle]{verbatim}}%
21 }
22
23 \newcommand*{\LWR@fvendnone}{%
24 \LWR@traceinfo{fvendnone}%
25 \hbox to\z@{\LWR@afterendverbatim}%
26 }
27
28 \newcommand*{\LWR@fvstartsingle}{%
29 \LWR@traceinfo{fvstartsingle}%
30 \LWR@fvstartnone%
31 \FV@BeginListFrame@Single%
32 }
33
34 \newcommand*{\LWR@fvendsingle}{%
35 \LWR@traceinfo{fvendsingle}%
36 \FV@EndListFrame@Single%
37 \LWR@fvendnone%
38 }
39
40 \newcommand*{\LWR@fvstartline}{%
41 \LWR@traceinfo{fvstartline}%
42 \LWR@fvstartnone%
43 \FV@BeginListFrame@Lines%
44 }
45
46 \newcommand*{\LWR@fvendline}{%
47 \LWR@traceinfo{fvendline}%
48 \FV@EndListFrame@Lines%
49 \LWR@fvendnone%
50 }
```

The following patches select the start/left/right/end behaviors depending on `frame`. Original code is from the `fancyvrb` package.

```
51
52 \newcommand*{\LWR@FVfindcurrentcolor}{%
53 \protect\colorlet{\LWR@current@color}{.}%
54 \protect\convertcolorspec{named}{\LWR@current@color}{HTML}{\LWR@tempcolor%
```



```

55 }
56
57 \newcommand*{\LWR@FVtextstyle}{%
58 \LWR@FVfindcurrentcolor
59 \ifdefstring{\LWR@tempcolor}{000000}%
60 {}%
61 {color: \#\LWR@tempcolor ; }%
62 }
63
64
65 \newcommand*{\LWR@FVfindbordercolor}{%
66 \FancyVerbRuleColor%
67 \LWR@FVfindcurrentcolor
68 \color{black}
69 }
70
71 % border width of \FV@FrameRule
72 \newcommand*{\LWR@FVborderstyle}[1]{%
73 padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ;
74 \LWR@FVfindbordercolor
75 border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt solid \#\LWR@tempcolor ;
76 }
77
78 \def\FV@Frame@none{%
79 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle}%
80 \let\FV@BeginListFrame\LWR@fvstartnone%
81 \let\FV@LeftListFrame\relax%
82 \let\FV@RightListFrame\relax%
83 \let\FV@EndListFrame\LWR@fvendnone}
84
85 \FV@Frame@none% default values
86
87 \def\FV@Frame@single{%
88 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{}}%
89 \let\FV@BeginListFrame\LWR@fvstartsingle%
90 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
91 \let\FV@RightListFrame\FV@RightListFrame@Single%
92 \let\FV@EndListFrame\LWR@fvendsingle}
93
94 \def\FV@Frame@lines{%
95 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-top}\LWR@FVborderstyle{-bottom}}
96 \let\FV@BeginListFrame\LWR@fvstartline%
97 \let\FV@LeftListFrame\relax%
98 \let\FV@RightListFrame\relax%
99 \let\FV@EndListFrame\LWR@fvendline}
100
101 \def\FV@Frame@topline{%
102 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-top}}%
103 \let\FV@BeginListFrame\LWR@fvstartline%
104 \let\FV@LeftListFrame\relax%

```

```

105 \let\FV@RightListFrame\relax%
106 \let\FV@EndListFrame\LWR@fvendnone}
107
108 \def\FV@Frame@bottomline{%
109 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-bottom}}}%
110 \let\FV@BeginListFrame\LWR@fvstartnone%
111 \let\FV@LeftListFrame\relax%
112 \let\FV@RightListFrame\relax%
113 \let\FV@EndListFrame\LWR@fvendline}
114
115 \def\FV@Frame@leftline{%
116 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-left}}}%
117 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
118 \ifx\FancyVerbFillColor\relax%
119 \let\FV@FrameFillLine\relax%
120 \else%
121 \@tempdima\FV@FrameRule\relax%
122 \multiply\@tempdima-\tw@%
123 \edef\FV@FrameFillLine{%
124 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}%
125 \kern-\number\@tempdima sp}}%
126 \fi%
127 \let\FV@BeginListFrame\LWR@fvstartnone%
128 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
129 \let\FV@RightListFrame\relax%
130 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the fancyvrb package.

```

131 \def\FV@SingleFrameLine#1{%
132 \hbox to\z@{%
133 % \kern\leftmargin
134 \ifnum#1=\z@\relax
135 \let\FV@Label\FV@LabelBegin
136 \else
137 \let\FV@Label\FV@LabelEnd
138 \fi
139 \ifx\FV@Label\relax
140 % \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
141 \else
142 \ifnum#1=\z@
143 % \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
144 \ifx\FV@LabelPositionTopLine\relax
145 \else
146 \LWR@FVfindbordercolor
147 \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
148 \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
149 \LWR@htmltagc{/div}

```

```

150      \fi
151    \else
152    %      \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
153      \ifx\FV@LabelPositionBottomLine\relax
154      \else
155      \LWR@FVfindbordercolor
156      \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
157      \LWR@origtextrm{\FV@LabelEnd}
158      \LWR@htmltagc{/div}
159      \fi
160    \fi
161
162    \fi
163    \hss
164  }
165 }

```

Processes each line, adding optional line numbers. Original code is from the fancyvrb package.

```

166 \def\FV@ListProcessLine#1{%
167   \hbox to \hsize{%
168     %      \kern\leftmargin
169     \hbox to \VerbatimHTMLWidth {%
170       \ifcvoid{FV@LeftListNumber}{\kern 2.5em}%
171       \FV@LeftListNumber%
172     %      \FV@LeftListFrame
173     \FancyVerbFormatLine{#1}%
174     \hss%
175     %      \FV@RightListFrame
176     \FV@RightListNumber%
177   }%
178   \hss% required to avoid underfull hboxes
179 }
180 }

```

Env BVerbatim

```

181 \AtBeginEnvironment{BVerbatim}
182 {
183   \LWR@forcenewpage
184   \LWR@atbeginverbatim{bverbatim}
185
186 }
187
188 \AfterEndEnvironment{BVerbatim}
189 {
190   \leavevmode\par\LWR@origvspace{-\baselineskip}
191   \LWR@afterendverbatim

```

192 }

End of the modifications to make at the end of the preamble:

193 } % \AfterEndPreamble

194 \end{warpHTML}

## Package 31

# lwarp-figcaps.sty

## 105 Figcaps

Pkg figcaps Emulated.

for HTML output: Discard all options for lwarp-figcaps:

```
1 \LWR@ProvidesPackageDrop{figcaps}

2 \newcommand*{\figcaption}{}
3 \newcommand*{\figcaptionoff}{}
4 \newcommand*{\printfigures}{}
5 \newcommand*{\figmarkon}{}
6 \newcommand*{\figmarkoff}{}
7 \def\figurecaptionname{Figure Captions}
8 \def\tablepagenamename{Tables}
9 \def\figurepagenamename{Figures}
```

## Package 32

# lwarp-float.sty

## 106 Float and \newfloat

`Pkg float` float is emulated during HTML output, and the float package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{float}[2016/03/04]`

See section 59.2 for the `\listof` command.

`\newfloat`  $\{\langle 1: type \rangle\} \{\langle 2: placement \rangle\} \{\langle 3: ext \rangle\} [\langle 4: within \rangle]$

Emulates the `\newfloat` command from the float package.

“placement” is ignored.

```
2 \NewDocumentCommand{\newfloat}{m m m o}{%
3 \IfValueTF{#4}
4 {\DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}}
5 {\DeclareFloatingEnvironment[fileext=#3]{#1}}
```

`newfloat` package automatically creates the `\listof` command for new floats, but float does not, so remove `\listof` here in case it is manually created later.

```
6 \cslet{listof#1s}\relax
7 \cslet{listof#1es}\relax
8 }
```

`\floatname`  $\{\langle type \rangle\} \{\langle name \rangle\}$

Sets the text name of the float, such as “Figure”.

```
9 \NewDocumentCommand{\floatname}{m +m}{%
10 \SetupFloatingEnvironment{#1}{name=#2}%
11 }
```

`\floatplacement`  $\{\langle type \rangle\} \{\langle placement \rangle\}$

Float placement is ignored.

```
12 \newcommand*{\floatplacement}[2]{%
13 \SetupFloatingEnvironment{#1}{placement=#2}%
14 }
```

`\floatstyle`  $\{\langle style \rangle\}$

Float styles are ignored.

```
15 \newcommand{\floatstyle}[1]{%  
16 }
```

`\restylefloat`  $* \{\langle style \rangle\}$

Float styles are ignored.

```
17 \NewDocumentCommand{\restylefloat}{s m}{%  
18 }
```

## Package 33

# lwarp-floatflt.sty

## 107 Floatflt

Pkg floatflt Emulated.

for HTML output: Discard all options for lwarp-floatflt:

```
1 \LWR@ProvidesPackageDrop{floatflt}
```

Borrowed from the lwarp version of keyfloat:

```
2 \NewDocumentEnvironment{KFLTfloatflt@marginfloat}{0{-1.2ex} m}
3 {% start
4 \LWR@maybeinthisfloat%
5 \LWR@forcenewpage
6 \LWR@stoppars%
7 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
8 \LWR@startpars%
9 \captionsetup{type=#2}%
10 }
11 {
12 \LWR@htmldivclassend{div}
13 }
14
15 \DeclareDocumentEnvironment{floatingfigure}{o m}
16 {\begin{KFLTfloatflt@marginfloat}{figure}}
17 {\end{KFLTfloatflt@marginfloat}}
18
19 \DeclareDocumentEnvironment{floatingtable}{o +m}
20 {\begin{KFLTfloatflt@marginfloat}{table}#2}
21 {\end{KFLTfloatflt@marginfloat}}
```



## Package 34

# lwarp-floatpag.sty

## 108 Floatpag

Pkg floatpag Emulated.

for **HTML output**: Discard all options for lwarp-floatpag:

```
1 \LWR@ProvidesPackageDrop{floatpag}

2 \newcommand*{\floatpagestyle}[1]{}
3 \newcommand*{\rotfloatpagestyle}[1]{}
4 \newcommand*{\thisfloatpagestyle}[1]{}

```


## Package 35


# lwarp-floatrow.sty

## 109 Floatrow

`Pkg floatrow` floatrow is emulated during HTML output, and the floatrow package is ignored.

**for HTML output:** `1 \LWR@ProvidesPackageDrop{floatrow}`

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

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

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

After everything has loaded, remember whether subcaption was loaded. If not, it is assumed that subfig is used instead:

```
2 \newbool{LWR@subcaptionloaded}
3
4 \AtBeginDocument{
5 \@ifpackageloaded{subcaption}
6 {\booltrue{LWR@subcaptionloaded}}
7 {\boolfalse{LWR@subcaptionloaded}}
8 }
```

```
\floatbox [1 preamble] {2 captype} [3 width] [4 height] [5 vert pos]
{6 caption} {7 object}
```

Only parameters for captype, width, caption, and object are used.

LWR@insubfloatrow is true if inside a subfloatrow environment.

There are two actions, depending on the use of `subcaption` or `subfig`.

```

9 \NewDocumentCommand{\floatbox}{o m o o o +m +m}{%
10 \ifbool{LWR@subcaptionloaded}%
11 {% subcaption

```

For `subcaption`:

```

12 \ifbool{LWR@insubfloatrow}%
13 {% subcaption in a subfloatrow

```

`subfigure` and `subtable` environments take width as an argument.

```

14 \IfValueTF{#3}%
15 {\@nameuse{sub#2}{#3}}%
16 {\@nameuse{sub#2}{\linewidth}}%
17 }% subcaption in a subfloatrow
18 {% subcaption not in subfloatrow

```

`figure` and `table` environments do not take a width argument.

```

19 \@nameuse{#2}%
20 }% subcaption not in subfloatrow
21 #6
22
23 #7

```

End the environments:

```

24 \ifbool{LWR@insubfloatrow}%
25 {\@nameuse{endsub#2}}%
26 {\@nameuse{end#2}}%
27 }% subcaption
28 {% assume subfig

```

For `subfig`:

```

29 \ifbool{LWR@insubfloatrow}%
30 {% subfig in a subfloatrow

```

`\subfloat` is a macro, not an environment.

Package `subfig`'s `\subfloat` command takes an optional argument which is the caption, but `\floatbox` argument `#6` contains commands to create the caption and label, not the caption itself. Thus, `\caption` is temporarily disabled to return its own argument without braces.

```

31 \begingroup
32 \let\caption\@firstofone

```

```

33   \subfloat[#6]{#7}
34   \endgroup
35 }% subfig in a subfloatrow
36 {% subfig package, but not a subfig

```

figure and table are environments:

```

37 \@nameuse{#2}
38 #6
39
40 #7
41 \@nameuse{end#2}
42 }% subfig package, but not a subfig
43 }% assume subfig
44 }

```

Not used:

```

45 \newcommand*{\nocapbeside}{}
46 \newcommand*{\capbeside}{}
47 \newcommand*{\captop}{}
48 \newlength{\FBwidth}
49 \setlength{\FBwidth}{.3\linewidth}
50 \newlength{\FBheight}
51 \setlength{\FBheight}{2in}
52 \newcommand*{\useFCwidth}{}
53 \newcommand{\floatsetup}[2][{}]{
54 \newcommand{\thisfloatsetup}[1]{
55 \newcommand{\clearfloatsetup}[1]{
56 \newcommand*{\killfloatstyle}{}

```

Preamble and default width are ignored.

```

57 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
58 \@namedef{#1}{
59 \floatbox{#2}
60 }
61 }

```

Preamble and default width are ignored.

```

62 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
63 \@namedef{#1}{%
64 \floatbox{#2}
65 }
66 }

67 \newfloatcommand{ffigbox}{figure}[\nocapbeside][

```

```
68 \newfloatcommand{ttabbox}{table}[\captop][\FBwidth]
```

```
69 \newfloatcommand{fcapside}{figure}[\capbeside][]
```

The row of floats is placed into a <div> of class `floatrow`.

```
70 \newenvironment*{floatrow}[1][2]
```

```
71 {
```

```
72 \LWR@forcenewpage
```

```
73 \BlockClass{floatrow}
```

While inside the `floatrow`, divide the `\linewidth` by the number of floats.

```
74 \booltrue{LWR@infloatrow}
```

```
75 \setlength{\linewidth}{6in/#1}
```

```
76 }
```

```
77 {
```

```
78 \boolfalse{LWR@infloatrow}
```

```
79 \endBlockClass
```

```
80 }
```

Keys for `\DeclareNewFloatType`:

```
81 \newcommand*{\LWR@frowkeyplacement}{}%
```

```
82 \newcommand*{\LWR@frowkeyname}{}%
```

```
83 \newcommand*{\LWR@frowkeyfileext}{}%
```

```
84 \newcommand*{\LWR@frowkeywithin}{}%
```

```
85 \newcommand*{\LWR@frowkeycapstyle}{}%
```

```
86
```

```
87 \define@key{frowkeys}{placement}{}%
```

```
88 \define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}%
```

```
89 \define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}%
```

```
90 \define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}%
```

```
91 \define@key{frowkeys}{relatedcapstyle}{}%
```

Use `\listof{type}{Title}` to print a list of the floats.

```
92 \newcommand*{\DeclareNewFloatType}[2]{%
```

Reset key values:

```
93 \renewcommand*{\LWR@frowkeyplacement}{}%
```

```
94 \renewcommand*{\LWR@frowkeyname}{}%
```

```
95 \renewcommand*{\LWR@frowkeyfileext}{}%
```

```
96 \renewcommand*{\LWR@frowkeywithin}{}%
```

```
97 \renewcommand*{\LWR@frowkeycapstyle}{}%
```

Read new key values:

```

98 \LWR@traceinfo{about to setkeys frowkeys}%
99 \setkeys{frowkeys}{#2}%
100 \LWR@traceinfo{finished setkeys frowkeys}%

```

Create a new float with optional [within]:

```

101 \ifthenelse{\equal{\LWR@frowkeywithin}{}}{
102 {
103   \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\
104     \LWR@frowkeyfileext}%
105   \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}
106 }%
107 {%
108   \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\
109     \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
110   \newfloat{#1}{\LWR@frowkeyplacement}%
111   {\LWR@frowkeyfileext}[\LWR@frowkeywithin]%
112   \LWR@traceinfo{finished newfloat #1}
113 }%

```

Rename the float if a name was given:

```

114 \ifthenelse{\equal{\LWR@frowkeyname}{}}{
115 {}
116 {\floatname{#1}{\LWR@frowkeyname}}%
117 }

```

Not used:

```

118 \newcommand{\buildFBBBOX}[2]{}
119 \newcommand*{\CenterFloatBoxes}{}
120 \newcommand*{\TopFloatBoxes}{}
121 \newcommand*{\BottomFloatBoxes}{}
122 \newcommand*{\PlainFloatBoxes}{}
123
124 \newcommand{\capsubrowsettings}{}
125
126 \NewDocumentCommand{\RawFloats}{o o}{}

```

To be used inside a minipage or parbox.

```

127 \newcommand{\RawCaption}[1]{#1}

```

Places additional text inside a float, inside a CSS <div> of class floatfoot.

```

128 \NewDocumentCommand{\floatfoot}{s +m}{%
129 \begin{BlockClass}{floatfoot}
130 #2
131 \end{BlockClass}

```

```
132 }
```

Used to compute `\linewidth`.

```
133 \newbool{LWR@insubfloatrow}  
134 \boolfalse{LWR@insubfloatrow}
```

```
135 \newenvironment*{subfloatrow}[1][2]  
136 {
```

The row of floats is placed into a `<div>` of class `floatrow`:

```
137 \LWR@forcenewpage  
138 \BlockClass{floatrow}
```

While inside the `floatrow`, `LWR@insubfloatrow` is set true, which tells `\floatbox` to use `\subfigure` or `\subtable`.

```
139 \begingroup  
140 \booltrue{LWR@insubfloatrow}  
141 }  
142 {  
143 \endgroup  
144 \endBlockClass  
145 \boolfalse{LWR@insubfloatrow}  
146 }
```

## Package 36

# lwarp-flushend.sty

## 110 Flushend

Pkg flushend Emulated.

for HTML output: Discard all options for lwarp-flushend:

```
1 \LWR@ProvidesPackageDrop{flushend}
2 % \end{ma-crocode}
3 %
4 % \begin{macrocode}
5 \newcommand*{\flushend}{}
6 \newcommand*{\raggedend}{}
7 \newcommand*{\flushcolsend}{}
8 \newcommand*{\raggedcolsend}{}
9 \newcommand*{\atColsBreak}[1]{}
10 \newcommand*{\atColsEnd}[1]{}
11 \newcommand*{\showcolsendrule}{}

```



## Package 37

# lwarp-fontenc.sty

## 111 Fontenc

Pkg fontenc Error if fontenc is loaded after lwarp.

Discard all options for lwarp-fontenc:

for HTML output: 1 \LWR@ProvidesPackageDrop{fontenc}  
2 \LWR@loadbefore{fontenc}

## Package 38

# lwarp-fontspec.sty

## 112 Fontspec

Pkg fontspec Error if fontspec is loaded after lwarp.

Discard all options for lwarp-fontspec:

for HTML output: 1 \LWR@ProvidesPackageDrop{fontspec}  
2 \LWR@loadbefore{fontspec}

## Package 39

# lwarp-footmisc.sty

## 113 Footmisc

*(Based on original code by ROBIN FAIRBAIRNS.)*

Pkg `footmisc` `footmisc` is emulated during HTML output, and the `footmisc` package is ignored.

```
1 \LWR@ProvidesPackageDrop{footmisc}
```

Some nullified commands:

```
2 \newcommand{\footnotelayout}{}
3 \newcommand{\setfnsymbol}[1]{}
4 \NewDocumentCommand{\DefineFNSymbols}{s m o m}{}
5
6 \newdimen\footnotemargin
7 \footnotemargin1.8em\relax
8
9 \newcommand*\hangfootparskip{0.5\baselineskip}
10 \newcommand*\hangfootparindent{0em}%
11
12 \let\pagefootnoterule\footnoterule
13 \let\mpfootnoterule\footnoterule
14 \def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}
15
16 \providecommand*\multiplefootnotemarker{3sp}
17 \providecommand*\multfootsep{,}
```

Using `cleveref`:

```
18 \providecommand*\footref}[1]{\labelcref{#1}}
```

The following work as-is:

```
19 \newcommand\mpfootnotemark{%
20   \@ifnextchar[%
21     \@xmpfootnotemark
22     {%
23       \stepcounter\@mpfn
24       \protected@xdef\@thefnmark{\thempfn}%
25       \@footnotemark
26     }%
27 }
```

---

```
28 \def\@xmpfootnotemark[#1]{%
29   \begingroup
30     \csname c@\@mpfn\endcsname #1\relax
31     \unrestored@protected@xdef\@thefnmark{\thempfn}%
32   \endgroup
33   \@footnotemark
34 }
```

## Package 40

# lwarp-footnote.sty

## 114 Footnote

Pkg footnote footnote is used with minor patches.

for HTML output: 1 \LWR@ProvidesPackagePass{footnote}

Removed print-version formatting:

```

2 \def\fn@startnote{%
3 %   \@parboxrestore%
4 \protected@edef\@currentlabel{\csname p@\@mpfn\endcsname\@thefnmark}%
5 %   \color@begingroup% *** conflicts with lwarp
6 }
7
8 % \let\fn@endnote\color@endgroup% *** conflicts with lwarp
9 \newcommand*{\fn@endnote}{%
10 \LWR@htmltagc{/\LWR@tagregularparagraph}%
11 \LWR@orignewline%
12 }
```

Removed print-version formatting:

```

13 \def\fn@startfntext{%
14 \setbox\z@\vbox\bgroup%
15 \fn@startnote%
16 \fn@prefntext%
17 \ignorespaces%
18 }
```

Removed print-version formatting, added closing paragraph tag:

```

19 \def\fn@endfntext{%
20 \LWR@htmltagc{/\LWR@tagregularparagraph}%
21 \LWR@orignewline%
22 \fn@postfntext%
23 \egroup%
24 \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\endfootnotetext\endfootnote
```

## Package 41

# lwarp-footnotehyper.sty

## 115 Footnotehyper

Pkg footnotehyper footnotehyper is a hyperref-safe version of footnote. For lwarp, footnotehyper is emulated.

for HTML output: Discard all options for lwarp-footnotehyper:

```
1 \RequirePackage{footnote}
2 \LWR@ProvidesPackageDrop{footnotehyper}
```

## Package 42

# lwarp-framed.sty

## 116 Framed

*(Based on original code by DONALD ARSENEAU.)*

Pkg `framed` `framed` is supported and patched by `lwarp`.

for HTML output: Accept all options for `lwarp-framed`:

```

1 \LWR@ProvidesPackagePass{framed}
2 \RequirePackage{xcolor}% for \convertcolorspec

3
4 \renewenvironment{framed}{%
5 \LWR@forcenewpage
6 \BlockClass{framed}%
7 }
8 {\endBlockClass}
9
10 \renewenvironment{oframed}{%
11 \LWR@forcenewpage
12 \BlockClass{framed}%
13 }
14 {\endBlockClass}
15
16
17 \renewenvironment{shaded}{%
18 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
19 \LWR@forcenewpage
20 \BlockClass[background: \#\LWR@tempcolor]{framed}%
21 }
22 {\endBlockClass}
23
24 \renewenvironment{shaded*}{%
25 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
26 \LWR@forcenewpage
27 \BlockClass[background: \#\LWR@tempcolor]{framed}%
28 }
29 {\endBlockClass}
30
31
32 \renewenvironment{leftbar}{%
33 \LWR@forcenewpage
34 \BlockClass{framedleftbar}

```

```

35 \def\FrameCommand{ }%
36 \MakeFramed {}
37 }%
38 {\endMakeFramed\endBlockClass}
39
40
41 \renewenvironment{snugshade}{%
42 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
43 \LWR@forcenewpage
44 \BlockClass[background: \#\LWR@tempcolor]{snugframed}%
45 }
46 {\endBlockClass}
47
48 \renewenvironment{snugshade*}{%
49 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
50 \LWR@forcenewpage
51 \BlockClass[background: \#\LWR@tempcolor]{snugframed}%
52 }
53 {\endBlockClass}
54
55 \let\oframed\framed
56 \let\endoframed\endframed
57
58
59 \RenewEnviron{titled-frame}[1]{%
60 \CustomFBox{#1}{ }{0pt}{0pt}{0pt}{0pt}{\BODY}
61 }

\CustomFBox {<toptitle>} {<bottitle>} {<thicknesstop>} {<bottom>} {<left>} {<right>}
{<text contents>}

62 \renewcommand{\CustomFBox}[7]{%
63 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
64 \LWR@forcenewpage
65 \begin{BlockClass}[border: 3px solid \#\LWR@tempcolor]{framed}%
66 \ifthenelse{\isempty{#1}}{ }{% not empty
67 \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
68 \textcolor{TFTitleColor}{\textbf{#1}}%
69 \end{BlockClass}
70 }% not empty
71
72 #7
73
74 \ifthenelse{\isempty{#2}}{ }{% not empty
75 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
76 \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
77 \textcolor{TFTitleColor}{\textbf{#2}}%
78 \end{BlockClass}
79 }% not empty

```

---

```

80 \end{BlockClass}
81 }

\TitleBarFrame [\marker] {\i\title} {\i\contents}

82 \renewcommand\TitleBarFrame[3][]{
83 \CustomFBox
84   {\i2}{\i}%
85   \fboxrule\fboxrule\fboxrule\fboxrule
86   {\i3}%
87 }

88 \renewcommand{\TF@Title}[1]{\i1}

MakeFramed {\i\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 {\i\frame cmd no split} {\i\frame cmd split}

95 \renewcommand*\fb@put@frame[2]{%
96 \relax%
97 \@tempboxa%
98 }

```



## Package 43

# lwarp-ftnright.sty

## 117 Ftnright

Pkg `ftnright` ftnright is ignored.

for **HTML output**: Discard all options for lwarp-ftnright:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

## Package 44

# lwarp-fullpage.sty

## 118 Fullpage

Pkg `fullpage` Emulated.

for **HTML output**: Discard all options for lwarp-fullpage:

```
1 \LWR@ProvidesPackageDrop{fullpage}
```

## Package 45

# lwarp-geometry.sty

## 119 Geometry

Pkg `geometry` `geometry` is preloaded by `lwarp`, but must be nullified as seen by the user's source code.

for HTML output: Discard all options for `lwarp-geometry`:

```
1 \LWR@ProvidesPackageDrop{geometry}

2 \renewcommand*\geometry}[1]{}
3 \renewcommand*\newgeometry}[1]{}
4 \renewcommand*\restoregeometry[1]{}
5 \renewcommand*\savegeometry}[1]{}
6 \renewcommand*\loadgeometry}[1]{}

```

## Package 46

# lwarp-glossaries.sty

## 120 Glossaries

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

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

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

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

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

for HTML output:

```

1 \PassOptionsToPackage{xindy}{glossaries}
2 \LWR@ProvidesPackagePass{glossaries}
3 \setupglossaries{nonumberlist}
4 \setglossarystyle{index}
```

## Package 47

# lwarp-graphics.sty

## 121 Graphics

Pkg graphics graphics is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphics}

## Package 48

# lwarp-graphicx.sty

## 122 Graphicx

Pkg graphicx graphicx is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphicx}

## Package 49

# lwarp-hyperref.sty

## 123 Hyperref

Pkg hyperref hyperref is emulated during HTML output, and the hyperref package is ignored.

for HTML output:

```

1 % \LWR@ProvidesPackageDrop{hyperref}
2 \typeout{Using the lwarp html version of package 'hyperref' -- discarding options.}
3 \typeout{    Are not using ProvidesPackage, so that other packages}
4 \typeout{    do not attempt to patch lwarp's version of 'hyperref'.}
5 % \ProvidesPackage{lwarp-#1-#2}
6 \DeclareOption*{}
7 \ProcessOptions\relax

8 \newcommand*{\hypersetup}[1]{}
9 \newcommand*{\hyperbaseurl}[1]{}

```

Insert an image with alt text:

```

10 \NewDocumentCommand{\LWR@hyperimageb}{m +m}{%
11 \LWR@htmltag{img src="#1" alt="#2" class="hyperimage"{} }%
12 \endgroup%
13 \LWR@ensuredoingapar%
14 }
15
16 \newcommand{\hyperimage}{%
17 \LWR@ensuredoingapar%
18 \begingroup\catcode'\_ =12
19 \LWR@hyperimageb%
20 }

```

Creates an HTML anchor to category.name with the given text.

```

21 \NewDocumentCommand{\hyperdef}{m m +m}{%
22 \LWR@ensuredoingapar%
23 \LWR@subsublabel{#1.#2}%
24 #3%
25 }

```

Creates an HTML link to URL#category.name with the given text.

```

26 \NewDocumentCommand{\LWR@hyperrefb}{m m m +m}{%
27 \LWR@htmltag{a href="#1\LWR@hashmark#2.#3"%

```

```

28 #4%
29 \LWR@htmltag{/a}%
30 \endgroup%
31 }

```

Creates text as an HTML link to the L<sup>A</sup>T<sub>E</sub>X label.

```

32 \NewDocumentCommand{\LWR@hyperrefc}{O{label} +m}{
33 \LWR@startref{#1}%
34 #2%
35 \LWR@htmltag{/a}%
36 \endgroup%
37 }

```

```

38 \newcommand{\hyperref}{%
39 \LWR@ensuredoingapar%
40 \begingroup\catcode'\_ =12
41 \@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
42 }

```

Creates an anchor to name with the given text.

```

43 \NewDocumentCommand{\hypertarget}{m +m}{%
44 \label{#1}%
45 #2%
46 }

```

Creates a link to the anchor created by `hypertarget`, with the given link text.

```

47 \NewDocumentCommand{\hyperlink}{m +m}{%
48 \hyperref{#1}{#2}%
49 }

```

For HTML, `\cleveref` is used instead.

```

50 \NewDocumentCommand{\autoref}{s m}{%
51 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
52 }

```

For HTML, `\cleveref` is used instead.

```

53 \NewDocumentCommand{\autopageref}{s m}{%
54 \IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
55 }

```

```

56 \newcommand{\pdfstringdef}[2]{}

```

```

57 \newcommand{\pdfbookmark}[3][]{ }

```

```
58 \newcommand{\currentpdfbookmark}[2]{}{}
```

```
59 \newcommand{\subpdfbookmark}[2]{}{}
```

```
60 \newcommand{\belowpdfbookmark}[2]{}{}
```

```
61 \newcommand{\texorpdfstring}[2]{#2}
```

From hyperref.

```
62 \def\hypercalcbp#1{%  
63 \strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax  
64 }%
```

```
65 \newcommand{\Acrobatmenu}[2]{}{}
```

```
66 \newcommand*{\TextField}[2][{}]{}
```

```
67 \newcommand*{\CheckBox}[2][{}]{}
```

```
68 \newcommand{\ChoiceMenu}[3][{}]{}
```

```
69 \newcommand*{\PushButton}[2][{}]{}
```

```
70 \newcommand*{\Submit}[2][{}]{}
```

```
71 \newcommand*{\Reset}[2][{}]{}
```

```
72 \newcommand*{\LayoutTextField}[2]{}{}
```

```
73 \newcommand*{\LayoutChoiceField}[2]{}{}
```

```
74 \newcommand*{\LayoutCheckField}[2]{}{}
```

```
75 \newcommand*{\MakeRadioField}[2]{}{}
```

```
76 \newcommand*{\MakeCheckField}[2]{}{}
```

```
77 \newcommand*{\MakeTextField}[2]{}{}
```

```
78 \newcommand*{\MakeChoiceField}[2]{}{}
```

```
79 \newcommand{\MakeFieldButton}[1]{}{}
```

## Package 50

# lwarp-hyperxmp.sty

## 124 Hyperxmp

Pkg hyperxmp Emulated.

for **HTML output:** Discard all options for lwarp-hyperxmp:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```



## Package 51

# lwarp-idxlayout.sty

## 125 Idxlayout

Pkg idxlayout Emulated.

for **HTML output**: Discard all options for lwarp-idxlayout:

```

1 \LWR@ProvidesPackageDrop{idxlayout}

2 \newcommand{\LWR@indexprenote}{}
3
4 \renewcommand*{\printindex}
5 {
6 \LWR@indexsection
7 \LWR@startpars
8
9 \LWR@indexprenote
10
11 \LWR@origprintindex
12 }
13
14 \newcommand{\setindexprenote}[1]{\renewcommand{\LWR@indexprenote}{#1}}
15 \newcommand*{\noindexprenote}{\renewcommand{\LWR@indexprenote}{} }
16
17 \newcommand{\idxlayout}[1]{}
18 \newcommand*{\indexfont}{}
19 \newcommand*{\indexjustific}{}
20 \newcommand*{\indexsubsdelim}{}
21 \newcommand*{\indexstheadcase}{}

```

## Package 52

# lwarp-indentfirst.sty

## 126 Indentfirst

Pkg indentfirst indentfirst is ignored.

Discard all options for lwarp-indentfirst:

for HTML output: 1 \LWR@ProvidesPackageDrop{indentfirst}

## Package 53

# lwarp-inputenc.sty

## 127 Inputenc

Pkg inputenc Error if inputenc is loaded after lwarp.

Discard all options for lwarp-inputenc:

for HTML output: 1 \LWR@ProvidesPackageDrop{inputenc}

2 \LWR@loadbefore{inputenc}

## Package 54

# lwarp-keyfloat.sty

## 128 Keyfloat

Pkg keyfloat keyfloat is supported with minor adjustments.

for HTML output: 1 \LWR@ProvidesPackagePass{keyfloat}

After keyfloat has loaded:

```

2 \AtBeginDocument{

3 \let\KFLT@boxinner\relax
4 \let\endKFLT@boxinner\relax
5
6 \NewEnviron{KFLT@boxinner}
7 {%
8 \LWR@traceinfo{kflt@boxinner}%
9 \LWR@stoppars%
10 \KFLT@frame{\BODY}%
11 \LWR@startpars%
12 \LWR@traceinfo{ended kflt@boxinner}%
13 }

14 \DeclareDocumentEnvironment{KFLT@marginfloat}{0{-1.2ex} m}
15 {% start
16 \LWR@maybeinthisfloat%
17 \LWR@forcenewpage
18 \LWR@stoppars%
19 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
20 \LWR@startpars%
21 \captionsetup{type=#2}%
22 }
23 {
24 \LWR@htmldivclassend{div}
25 }

26 \DeclareDocumentEnvironment{marginfigure}{o}
27 {\begin{KFLT@marginfloat}{figure}}
28 {\end{KFLT@marginfloat}}
29
30 \DeclareDocumentEnvironment{margintable}{o}
31 {\begin{KFLT@marginfloat}{table}}
32 {\end{KFLT@marginfloat}}
```

---

```
33 \DeclareDocumentEnvironment{keywrap}{m +m}
34 {%
35 \begin{BlockClass}{marginblock}
36 \setlength{\linewidth}{#1}
37 #2%
38 \end{BlockClass}
39 }
40 {%
41 }

42 }% AtBeginDocument
```

## Package 55

# lwarp-layout.sty

## 129 Layout

Pkg layout layout is ignored.

for HTML output: Discard all options for lwarp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}
2 \NewDocumentCommand{\layout}{s}{}
```

## Package 56

# lwarp-letterspace.sty

## 130 Letterspace

Pkg letterspace letterspace is a subset of microtype, which is pre-loaded by lwarp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for lwarp-letterspace:

```
1 \LWR@ProvidesPackageDrop{letterspace}
2 \newcommand*\lsstyle{}
3 \newcommand\textls[2] [] {}
4 \def\textls#1#{}
5 \newcommand*\lslig[1]{#1}
```

## Package 57

# lwarp-lettrine.sty

## 131 Lettrine

(Based on original code by DANIEL FLIPO.)

Pkg lettrine Emulated.

for HTML output: Discard all options for lwarp-lettrine:

```
1 \LWR@ProvidesPackageDrop{lettrine}
```

The initial letter is in a `<span>` of class `lettrine`, and the following text is in a `<span>` of class `lettrinetext`. `\lettrine [keys] {letter} {additional text}`

```
2 \DeclareDocumentCommand{\lettrine}{o m m}{%
3 \InlineClass{lettrine}{#2}\InlineClass{lettrinetext}{#3} %
4 }
5
6 \newcounter{DefaultLines}
7 \setcounter{DefaultLines}{2}
8 \newcounter{DefaultDepth}
9 \newcommand*{\DefaultOptionsFile}{\relax}
10 \newcommand*{\DefaultLoversize}{0}
11 \newcommand*{\DefaultLraise}{0}
12 \newcommand*{\DefaultLhang}{0}
13 \newdimen\DefaultFindent
14 \setlength{\DefaultFindent}{\z@}
15 \newdimen\DefaultNindent
16 \setlength{\DefaultNindent}{0.5em}
17 \newdimen\DefaultSlope
18 \setlength{\DefaultSlope}{\z@}
19 \newdimen\DiscardVskip
20 \setlength{\DiscardVskip}{0.2\p@}
21 \newif\ifLettrineImage
22 \newif\ifLettrineOnGrid
23 \newif\ifLettrineRealHeight
24
25 \newcommand*{\LettrineTextFont}{\scshape}
26
27 \newcommand*{\LettrineFontHook}{\relax}
28
29 \newcommand*{\LettrineFont}[1]{\InlineClass{lettrine}{#1}}
30 \newcommand*{\LettrineFontEPS}[1]{\includegraphics[height=1.5ex]{#1}}
```

## Package 58

# lwarp-lips.sty

## 132 Lips

Pkg lips lips is emulated during HTML output, and the lips package is ignored.

```
1 % \LWR@ProvidesPackageDrop{lips}
2 \PackageInfo{lwarp}{Using the lwarp version of package 'lips'.}%
3 \ProvidesPackage{lwarp-lips}
4
5 \NewDocumentCommand{\Lips}{-}{\textellipsis}
6
7 \NewDocumentCommand{\BracketedLips}{-}{[\textellipsis]}
8
9 \let\lips\Lips
10 \let\olips\lips
11
12 \DeclareOption*{}
13 \DeclareOption{mla}{
14 \let\lips\BracketedLips
15 }
16 \ProcessOptions\relax
17
18 \newcommand \LPNobreakList {}
```

## Package 59

# lwarp-listings.sty

## 133 Listings

(Based on original code by CARSTEN HEINZ, BROOKS MOSES, JOBST HOFFMANN.)

`\pkg{listings}` listings is supported with some limitations. Text formatting is not yet supported.

for HTML output: `1 \begin{warpHTML}`

`2 \LWR@ProvidesPackagePass{listings}`

Patches to embed listings inside `pre` tags:

`3 \let\LWR@origlst@Init\lst@Init`

`4 \let\LWR@origlst@DeInit\lst@DeInit`

`5`

`6 \let\LWR@origlsthkEveryPar\lsthk@EveryPar`

`7`

`8 \renewcommand{\l@lstlisting}[2]{\hypertocfloat{1}{\lstlisting}{1ol}{#1}{#2}}`

Done at the start of a listing.

`9 \renewcommand{\lst@Init}[1]{%`

First, perform the listings initialization:

`10 \LWR@traceinfo{\lst@Init}%`

`11 \renewcommand*\@capttype{\lstlisting}%`

`12 \LWR@origlst@Init{#1}%`

`13 \LWR@traceinfo{finished origlst@Init}%`

`14 \lst@ifdisplaystyle%`

Creating a display.

Disable line numbers, produce the `<pre>`, then reenable line numbers.

`15 \LWR@traceinfo{About to create verbatim.}%`

`16 \let\lsthk@EveryPar\relax%`

`17 \LWR@forcenewpage`

`18 \LWR@atbeginverbatim{programlisting}%`

`19`

`20 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%`

`21 \else%`



Inline, so open a `<span>`:

```
22 \ifbool{LWR@verbtags}{\LWR@htmltag{span class="inlineprogramlisting"}}{}%
23 \fi%
24 }

25 \renewcommand*{\lst@DeInit}{%
26 \lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the `</pre>`, then reenable line numbers:

```
27 \let\lsthk@EveryPar\relax%
28
29 \LWR@afterendverbatim%
30 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
31 \else%
```

Inline, so create the closing `</span>`:

```
32 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
33 \fi%
```

Final listings deinit:

```
34 \LWR@origlst@DeInit%
35 }
```

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```
36 \def\lst@MakeCaption#1{%
37 \LWR@traceinfo{MAKING CAPTION at #1}%
38 \lst@ifdisplaystyle
39 \LWR@traceinfo{making a listings display caption}%
40 \ifx #1%
41 \ifx\lst@@caption\@empty\expandafter\lst@HRefStepCounter \else
42 \expandafter\refstepcounter
43 \fi {lstlisting}%
44 \LWR@traceinfo{About to assign label: !\lst@label!}%
45 % \ifx\lst@label\@empty\else
46 % \label{\lst@label}\fi
47 \LWR@traceinfo{Finished assigning the label.}%
48 \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
49 \global\let\lst@name\lst@arg \global\let\lstname\lst@name
50 \lst@ifnolol\else
51 \ifx\lst@@caption\@empty
```

```

52             \ifx\lst@caption\@empty
53             \ifx\lst@intname\@empty \else \def\lst@temp{ }%
54             \ifx\lst@intname\lst@temp \else

```

This code places a contents entry for a non-float. This would have to be modified for lwarp:

```

55 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
56 %             \addcontentsline{lol}{lstlisting}{\lst@name}
57             \fi\fi
58             \fi
59             \else

```

This would have to be modified for lwarp:

```

60 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
61             \addcontentsline{lol}{lstlisting}%
62 {\protect\numberline{\thelstlisting}%
63 {\protect\ignorespaces \lst@@caption \protect\relax}}%
64             \fi
65             \fi
66             \fi
67             \ifx\lst@caption\@empty\else
68 \LWR@traceinfo{lst@caption not empty-}%
69             \lst@ifsubstring #1\lst@captionpos
70             {\begingroup
71 \LWR@traceinfo{at the selected position}%

```

These space and box commands are not needed for HTML output:

```

72 %             \let\@vskip\vskip
73 %             \def\vskip{\afterassignment\lst@vskip \@tempkipa}%
74 %             \def\lst@vskip{\nobreak\@vskip\@tempkipa\nobreak}%
75 %             \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
76 %             \ifx #1t\allowbreak \fi
77             \ifx\lst@title\@empty

```

New lwarp code to create a caption:

```

78             \lst@makecaption\fnnum\lstlisting{\ignorespaces \lst@caption}
79             \else

```

New lwarp code to create a title:

```

80 %             \lst@maketitle\lst@title % (AS)
81 \LWR@traceinfo{Making title: \lst@title}%
82 \begin{BlockClass}{lstlistingtitle}% lwarp
83 \lst@maketitle\lst@title% lwarp
84 \end{BlockClass}% lwarp

```

```

85             \fi
86 \LWR@traceinfo{About to assign label: !\lst@label!}%
87             \ifx\lst@label\@empty\else
88 \leavevmode% gets rid of bad space factor error
89 \GetTitleStringExpand{\lst@caption}%
90 \edef\LWR@lntemp{\GetTitleStringResult}%
91 \edef\@currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
92 \label{\lst@label}\fi
93 \LWR@traceinfo{Finished assigning the label.}%

```

Not needed for lwarp:

```

94 %             \ifx #1b\allowbreak \fi
95             \endgroup}{}%
96     \fi
97 \LWR@traceinfo{end of making a listings display caption}%
98     \else
99 \LWR@traceinfo{INLINE}%
100    \fi
101 \LWR@traceinfo{DONE WITH CAPTION at #1}%
102 }

```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```

103 \lst@Key{numbers}{none}{%
104     \let\lst@PlaceNumber\@empty
105     \lstKV@SwitchCases{#1}%
106     {none&\\%
107         left&\def\lst@PlaceNumber{%
108 % \llap{
109 \LWR@orignormalfont%
110 \lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
111 % }
112 }
113 \\%
114     right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont
115         \kern\VerbatimHTMLWidth \kern\lst@numbersep
116         \lst@numberstyle{\thelstnumber}}}%
117     }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}
118 \end{warpHTML}

```

## Package 60

# lwarp-longtable.sty

## 134 Longtable

Pkg `longtable` `longtable` is emulated during HTML output, and the `longtable` package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{longtable}`



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

`\warpprintonly{row contents}`

instead of

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

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

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



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

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

or place it inside `\warpingprintonly`.

See:

<http://tex.stackexchange.com/questions/43006/why-is-input-not-expandable>

Env `longtable` `* [⟨horizontal⟩] {⟨colspec⟩}` Emulates the `longtable` environment.

Per the `caption` package, the starred version steps the counter per caption. The unstarred version steps the counter once at the beginning, but not at each caption.

Options `[c]`, `[l]`, and `[r]` are thrown away.

```

2 \newenvironment{longtable*}[2][{}]{%
3 \LWR@floatbegin{table}%
4 \setcaptiontype{LTcaption}%
5 \caption@setoptions{longtable}%
6 \caption@setoptions{@longtable}%
7 \caption@LT@setup%
8 \booltrue{LWR@starredlongtable}%
9 \let\captionlistentry\LWR@LTcaptionlistentry%
10 \LWR@tabular{#2}
11 }
```

```

12 {\endLWR@tabular\LWR@floatend}
13
14 \newenvironment{longtable}[2][\%
15 \LWR@floatbegin{table}%
16 \setcaptiontype{\LTcapttype}%
17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{\LTcapttype}%
21 \let\captionlistentry\LWR@LTcaptionlistentry%
22 \LWR@tabular{#2}
23 }
24 {\endLWR@tabular\LWR@floatend}
25

```

Provided for compatibility, but ignored:

```

26 \newcounter{LTchunksz}
27 \def\endhead{\LWR@tabularendofline}% throws away options //[dim] and /*
28 \def\endfirsthead{\LWR@tabularendofline}
29 \def\endfoot{\LWR@tabularendofline}
30 \def\endlastfoot{\LWR@tabularendofline}
31 \newcommand\tabularnewline{\LWR@tabularendofline}
32 \newcommand{\setlongtables}{}% Obsolete command, does nothing.
33 \newlength{\LTleft}
34 \newlength{\LTRight}
35 \newlength{\LTpre}
36 \newlength{\LTpost}
37 \newlength{\LTcapwidth}

38 \renewcommand*{\kill}{\LWR@tabularendofline}

```

## Package 61

# lwarp-lscape.sty

## 135 Lscape

Pkg lscape lscape is nullified.  
for HTML output: Discard all options for lwarp-lscape.

```
1 \LWR@ProvidesPackageDrop{lscape}
2 \newenvironment*{landscape}{}{}
```

## Package 62

# lwarp-ltcaption.sty

## 136 Ltcaption

Pkg ltcaption ltcaption is emulated during HTML output, and the ltcaption package is ignored.  
for HTML output: 1 \LWR@ProvidesPackageDrop{ltcaption}

\LTcaptype is already defined by lwarp.

longtable\* is already defined by lwarp-longtable.

```
2 \newlength{\LTcapskip}
3 \newlength{\LTcapleft}
4 \newlength{\LTcapright}
5 \newcommand*{\LTcapmarginfalse}{}

```

## Package 63

# lwarp-marginfit.sty

## 137 Marginfit

Pkg marginfit Emulated.

for HTML output: Discard all options for lwarp-marginfit:

```
1 \LWR@ProvidesPackageDrop{marginfit}
```

## Package 64

# lwarp-marginfix.sty

## 138 Marginfix

Pkg marginfix Emulated.

for HTML output: Discard all options for lwarp-marginfix:

```
1 \LWR@ProvidesPackageDrop{marginfix}

2 \newcommand*{\marginskip}[1]{}
3 \newcommand*{\clearmargin}{}
4 \newcommand*{\softclearmargin}{}
5 \newcommand*{\extendmargin}[1]{}
6 \newcommand*{\mparshift}[1]{}
7 \newdimen\marginheightadjustment
8 \newdimen\marginposadjustment
9 \newcommand*{\blockmargin}[1][1]{}
10 \newcommand*{\unblockmargin}[1][1]{}
11 \newcommand*{\marginphantom}[2][1]{}

```

## Package 65

# lwarp-marginnote.sty

## 139 Marginnote

Pkg marginnote Emulated.

for HTML output: Discard all options for lwarp-marginnote:

```

1 \LWR@ProvidesPackageDrop{marginnote}

2 \NewDocumentCommand{\marginnote}{o +m o}{\marginpar{#2}}
3 \newcommand*{\marginnoteleftadjust}{}
4 \newcommand*{\marginnoterightadjust}{}
5 \newcommand*{\marginnotetextwidth}{}
6 \let\marginnotetextwidth\textwidth
7 \newcommand*{\marginnotevadjust}{}
8 \newcommand*{\marginfont}{}
9 \newcommand*{\raggedleftmarginnote}{}
10 \newcommand*{\raggedrightmarginnote}{}

```

## Package 66

# lwarp-mcaption.sty

## 140 Mcaption

Pkg mcaption mcaption is nullified.

for HTML output: Discard all options for lwarp-mcaption:

```

1 \LWR@ProvidesPackageDrop{mcaption}

2 \newenvironment{margincap}{}{}
3 \newcommand*{\margincapalign}{}
4 \newlength{\margincapsep}

```



## Package 67

# lwarp-mdframed.sty

## 141 Mdfamed

**Pkg** **mdframed** **mdframed** is loaded with options forced to **framemethod=none**.

**for HTML output:** `1 \RequirePackage{xcolor}% for \convertcolorspec`  
`2 \LWR@ProvidesPackageDrop{mdframed}`

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

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

**font** For title font, use

`frametitlefont=\textbf,`

instead of

`frametitlefont=\bfseries,`

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

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

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

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

**CSS classes** Environments created or encapsulated by **mdframed** are enclosed in a `<div>` of class **md<environmentname>**, or **mdframed** otherwise.

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

Pre-existing hooks are used to patch extra functions before and after the frames.

**amsthm** must be loaded before **mdframed**

`3 \LWR@origRequirePackage{amsthm}`

Do not require *Tikz* or *pstricks*:

```
4 \LWR@origRequirePackage[framemethod=none]{mdframed}
```

To handle CSS and paragraphs, patch code at start and end of environment and contents. `\LWR@origraggedright` helps avoid hyphenation.

```
5 \mdfsetup{
6 startcode={\LWR@mdframedstart\LWR@origraggedright},
7 endcode={\LWR@mdframedend},
8 startinnercode={\LWR@startpars\LWR@origraggedright},
9 endinnercode={\LWR@stoppars},
10 }
```

Given the `mdframed` key, print the color.

```
11 \newcommand*{\LWR@mdfprintcolor}[1]{%
12 \convertcolorspec{named}{\csuse{mdf@#1}}{HTML}\LWR@tempcolor%
13 \#\LWR@tempcolor
14 }
```

Given the `mdframed` key, print the length.

```
15 \newcommand*{\LWR@mdfprintlength}[1]{%
16 \rndprintlength{\csuse{mdf@#1@length}}
17 }
```

Actions before an `mdframe` starts.

Encapsulate a frame inside a `<div>` of the desired `class`.

```
18 \newcommand*{\LWR@mdframedstart}{%
```

Turn off paragraph handling during the generation of the encapsulating tags:

```
19 \LWR@stoppars%
```

Below, print HTML pt units:

```
20 \uselengthunit{PT}%
```

Open a `<div>` and with custom `class` and custom `style`:

```
21 \LWR@htmltagc{div class="\LWR@mdthisenv" \LWR@orignewline
22 style=" \LWR@orignewline
```

Convert and print the background color:

```
23 background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```
24 border: \LWR@mdfprintlength{linewidth} solid
25 \LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
26 border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
27 \ifbool{mdf@shadow}{%
28   box-shadow:
29     \LWR@mdfprintlength{shadowsize}
30     \LWR@mdfprintlength{shadowsize}
31     \LWR@mdfprintlength{shadowsize}
32     \LWR@mdfprintcolor{shadowcolor} ;
33 }
34 {box-shadow: none ;}
35 \LWR@orignewline

36 "}
37 % \LWR@htmldivclass{\LWR@mdthisenv}
```

**mdframed** environment may not work with the HTML versions of the following, so restore them to their originals while inside **mdframed**:

```
38 \LetLtxMacro{\hspace}{\LWR@orighspace}%
39 \LetLtxMacro\rule\LWR@origrule%
40 \LetLtxMacro\makebox\LWR@origmakebox%
41 }
```

Actions after an **mdframe** ends.

After closing the `<div>`, globally restore to the default environment type:

```
42 \newcommand*{\LWR@mdframedend}{
```

Close the custom `<div>`:

```
43 \LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
44 \gdef\LWR@mdthisenv{mdframed}
```

Resume paragraph handling:

```

45 \LWR@startpars%
46 }

```

Encapsulation of the original which places the title inside a `<span>` of class `mdframedtitle`:

```

47 \LetLtxMacro\LWR@origmdfframedtitleenv\mdfframedtitleenv
48
49 \newlength{\LWR@titleroundcorner}
50
51 \renewrobustcmd\mdfframedtitleenv[1]{%
52 \LWR@origmdfframedtitleenv{%

```

Below, print HTML pt lengths:

```

53 \uselengthunit{PT}%

```

Open a `<span>` with a custom class and custom style:

```

54 \LWR@htmltagc{span class="mdframedtitle" \LWR@orignewline
55 style=" \LWR@orignewline

```

Convert and print the title background color:

```

56 background:
57 \LWR@mdfprintcolor{frametitlebackgroundcolor}
58 ; \LWR@orignewline

```

Convert and print the title rule:

```

59 \ifbool{mdf@frametitlerule}{%
60   border-bottom:
61   \LWR@mdfprintlength{frametitlerulewidth}
62   solid
63   \LWR@mdfprintcolor{frametitlerulecolor}
64   ; \LWR@orignewline
65 }{}%

```

The title's top border radius is adjusted for the line width:

```

66 border-radius:
67 \setlength{\LWR@titleroundcorner}
68   {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{0pt}}
69   \rndprintlength{\LWR@titleroundcorner}
70   \rndprintlength{\LWR@titleroundcorner}
71   Opt Opt
72   \LWR@orignewline

```

Finish the custom style and the opening span tag:

```
73 " \LWR@orignewline
74 }% span
```

Restrict paragraph tags inside a span:

```
75 \begin{LWR@nestspan}%
```

Print the title inside the span:

```
76 #1%
```

Close the span and unnest the paragraph tag restriction:

```
77 \LWR@htmltagc{/span}%
78 \end{LWR@nestspan}%
79 }
80 }
```

Common code for `\LWR@mdfsubtitle` and `\LWR@mdfsubsubtitle`.

Encapsulate the subtitle inside a `<span>` of class `mdframedsubtitle`:

```
81 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
82 {% the following empty line is required
83
```

Special handling for `mdframed`: Subtitles have `\pars` around them, so temporarily disable them here.

```
84 \let\par\LWR@origpar%
```

Open a `<span>` with a custom class and custom style:

```
85 \LWR@htmltagc{span class="mdframed#1title"
86 style=" \LWR@orignewline
```

Convert and print the background color:

```
87 background:
88 \LWR@mdfprintcolor{#1titlebackgroundcolor}
89 ; \LWR@orignewline
```

Convert and print the above line:

```
90 \ifbool{mdf@#1titleaboveline}{%
91   border-top:
```

```

92   \LWR@mdfprintlength{#1titleabovelinewidth}
93   solid
94   \LWR@mdfprintcolor{#1titleabovelinecolor}
95   ; \LWR@orignewline
96 }{}%

```

Convert and print the below line:

```

97 \ifbool{mdf@#1titlebelowline}{%
98   border-bottom:
99   \LWR@mdfprintlength{#1titlebelowlinewidth}
100   solid
101   \LWR@mdfprintcolor{#1titlebelowlinecolor}
102   ; \LWR@orignewline
103 }{}%

```

Finish the custom style and the opening span tag:

```

104 "% span

```

Restrict paragraph tags inside a span:

```

105 \begin{LWR@nestspan}%

```

Perform the original subtitle action:

```

106 \IfNoValueTF{#2}
107 {\csuse{LWR@origmdf#1title}{#3}}%
108 {\csuse{LWR@origmdf#1title}[#2]{#3}}%

```

Close the span and unnest the paragraph tag restriction:

```

109 \LWR@htmltagc{/span}% the following empty line is required
110 \end{LWR@nestspan}% must follow the /span or an extra <p> appears
111
112 }

```

```

113 \let\LWR@origmdfsubtitle\mdfsubtitle
114
115 \newcommand*{\LWR@mdfsubtitle}{%
116 \LWR@mdfsubtitlecommon{sub}%
117 }
118 \let\mdfsubtitle\LWR@mdfsubtitle

```

```

119 \let\LWR@origmdfsubsubtitle\mdfsubsubtitle
120
121 \newcommand*{\LWR@mdfsubsubtitle}{%
122 \LWR@mdfsubsubtitlecommon{subsub}%
123 }

```

```
124 \let\mdfsubsubtitle\LWR@mdfsubsubtitle
```

Stores the environment of the frame about to be created:

```
125 \newcommand*{\LWR@mdthisenv}{mdframed}
```

Modified from the original to remember the environment.

```
126 \renewrobustcmd*\newmdenv[2] [] {%
127 \newenvironment{#2}%
128 {%
129 \mdfsetup{#1}%
130 \renewcommand*{\LWR@mdthisenv}{md#2}%
131 \begin{mdframed}%
132 }
133 {\end{mdframed}}%
134 }
```

Modified from the original to remember the environment.

```
135 \renewrobustcmd*{\surroundwithmdframed}[2] [] {%
136 \BeforeBeginEnvironment{#2}{%
137 \renewcommand*{\LWR@mdthisenv}{md#2}%
138 \begin{mdframed}[#1]}%
139 \AfterEndEnvironment{#2}{\end{mdframed}}%
140 }
```

```
[<numberedlike>] [<caption>] [<within>]
```

Modified from the original to remember the environment.

```
141 \DeclareDocumentCommand{\mdtheorem}{ O{} m o m o }%
142 {\ifcsdef{#2}%
143   {\mdf@PackageWarning{Environment #2 already exists\MessageBreak}}%
144   {%
145     \IfNoValueTF {#3}%
146     {%#3 not given -- number relationship
147       \IfNoValueTF {#5}%
148       {%#3+#5 not given
149         \@definecounter{#2}%
150         \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
151         \newenvironment{#2}[1] [] {%
152           \refstepcounter{#2}%
153           \ifstrempy{##1}%
154           {\let\@temptitle\relax}%
155           {%
156             \def\@temptitle{\mdf@theoremseparator%
157               \mdf@theoremspace%
158               \mdf@theoremtitlefont%
```

```

159             ##1}%
160         \mdf@thm@caption{#2}{#{#4}{\csname the#2\endcsname}{##1}}%
161     }%
162     \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
163                                     \@temptitle}]]%
164     {\end{mdframed}}%
165     \newenvironment{#2*}[1][]{%
166         \ifstrepty{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
167         \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]%
168         {\end{mdframed}}%
169     }%
170     {%#5 given -- reset counter
171     \@definecounter{#2}\@newctr{#2}[#5]%
172     \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
173     \expandafter\xdef\csname the#2\endcsname{%
174         \expandafter\noexpand\csname the#5\endcsname \@thmcountersep%
175         \@thmcounter{#2}}%
176     \newenvironment{#2}[1][]{%
177         \refstepcounter{#2}%
178         \ifstrepty{##1}%
179         {\let\@temptitle\relax}%
180         {%
181             \def\@temptitle{\mdf@theoremseparator%
182                             \mdf@theoremspace%
183                             \mdf@theoremtitlefont%
184                             ##1}%
185             \mdf@thm@caption{#2}{#{#4}{\csname the#2\endcsname}{##1}}%
186         }
187         \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
188                                     \@temptitle}]]%
189         {\end{mdframed}}%
190     \newenvironment{#2*}[1][]{%
191         \ifstrepty{##1}%
192         {\let\@temptitle\relax}%
193         {%
194             \def\@temptitle{\mdf@theoremseparator%
195                             \mdf@theoremspace%
196                             \mdf@theoremtitlefont%
197                             ##1}%
198             \mdf@thm@caption{#2}{#{#4}{\csname the#2\endcsname}{##1}}%
199         }%
200         \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]%
201         {\end{mdframed}}%
202     }%
203 }%
204 {%#3 given -- number relationship
205     \global\@namedef{the#2}{\@nameuse{the#3}}%
206     \newenvironment{#2}[1][]{%
207         \refstepcounter{#3}%
208         \ifstrepty{##1}%

```



```

209 \let\@temptitle\relax}%
210 {%
211 \def\@temptitle{\mdf@theoremseparator%
212 \mdf@theoremspace%
213 \mdf@theoremtitlefont%
214 ##1}%
215 \mdf@thm@caption{#2}{#{4}{\csname the#2\endcsname}{##1}}%
216 }
217 \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
218 \@temptitle}]]%
219 {\end{mdframed}}%
220 \newenvironment{#2*}[1][{}]{%
221 \ifstrempy{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
222 \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]%
223 {\end{mdframed}}%
224 }%
225 \BeforeBeginEnvironment{#2}{\renewcommand*{\LWR@mdthisenv}{md#2}}% new
226 \BeforeBeginEnvironment{#2*}{\renewcommand*{\LWR@mdthisenv}{md#2}}% new
227 }%
228 }

```

$$[\langle \textit{numberedlike} \rangle] \{ \langle \textit{caption} \rangle \} [\langle \textit{within} \rangle]$$

Modified from the original to remember the environment.

```

229 \DeclareDocumentCommand\newmdtheoremenv{0}{ m o m o }{%
230   \ifbool{expr{ test {\IfNoValueTF {#3}} and test {\IfNoValueTF {#5}} }}{%
231     {\newtheorem{#2}{#4}}{%
232       \IfValueTF{#3}{\newtheorem{#2}[#3]{#4}}{%
233         \IfValueTF{#5}{\newtheorem{#2}{#4}[#5]}{%
234           }%
235 \BeforeBeginEnvironment{#2}{%
236 \renewcommand*{\LWR@mdthisenv}{\md#2}%
237 \begin{mdframed}[#1]}%
238 \AfterEndEnvironment{#2}{%
239 \end{mdframed}}}%
240 }

```

## Package 68

# lwarp-microtype.sty

## 142 Microtype

Pkg microtype microtype is pre-loaded by lwarp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for lwarp-microtype:

```

1 \LWR@ProvidesPackageDrop{microtype}

2 \DeclareDocumentCommand{\DeclareMicrotypeSet}{o m m}{}
3 \DeclareDocumentCommand{\UseMicrotypeSet}{o m}{}
4 \DeclareDocumentCommand{\DeclareMicrotypeSetDefault}{o m}{}
5 \DeclareDocumentCommand{\SetProtrusion}{o m m}{}
6 \DeclareDocumentCommand{\SetExpansion}{o m m}{}
7 \DeclareDocumentCommand{\SetTracking}{o m m}{}
8 \DeclareDocumentCommand{\SetExtraKerning}{o m m}{}
9 \DeclareDocumentCommand{\SetExtraSpacing}{o m m}{}
10 \DeclareDocumentCommand{\DisableLigatures}{o m}{}
11 \DeclareDocumentCommand{\DeclareCharacterInheritance}{o m m}{}
12 \DeclareDocumentCommand{\DeclareMicrotypeVariants}{m}{}
13 \DeclareDocumentCommand{\DeclareMicrotypeAlias}{m m}{}
14 \DeclareDocumentCommand{\LoadMicrotypeFile}{m}{}
15 \DeclareDocumentCommand{\DeclareMicrotypeBabelHook}{m m}{}
16 \DeclareDocumentCommand{\microtypesetup}{m}{}
17 \DeclareDocumentCommand{\microtypecontext}{m}{}
18 \DeclareDocumentCommand{\textmicrotypecontext}{m m}{#2}
19 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
20 \DeclareDocumentCommand{\lsstyle}{}{}
21 \DeclareDocumentCommand{\textls}{o +m}{}
22 \DeclareDocumentCommand{\slig}{m}{#1}
23 }
24 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
25 \def\DeclareMicrotypeVariants#1#{\@gobble}
26 \@onlypreamble\DeclareMicrotypeSet
27 \@onlypreamble\UseMicrotypeSet
28 \@onlypreamble\DeclareMicrotypeSetDefault
29 \@onlypreamble\DisableLigatures
30 \@onlypreamble\DeclareMicrotypeVariants
31 \@onlypreamble\DeclareMicrotypeBabelHook

```

## Package 69

# lwarp-moreverb.sty

## 143 Moreverb

Pkg moreverb moreverb is supported with some patches.

```
for HTML output: 1 \begin{warpHTML}

2 \LWR@ProvidesPackagePass{moreverb}

3 \BeforeBeginEnvironment{verbatim}{%
4 \LWR@forcenewpage
5 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
6 }
7 \AfterEndEnvironment{verbatim}{%
8 \LWR@afterendverbatim%
9 }
10
11
12 \LetLtxMacro\LWRMV@orig@verbatiminput\@verbatiminput
13
14 \renewcommand{\@verbatiminput}[2][]{%
15 \LWR@forcenewpage
16 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
17 \LWRMV@orig@verbatiminput[#1]{#2}%
18 \LWR@afterendverbatim%
19 }
20
21 \BeforeBeginEnvironment{listing}{%
22 \LWR@forcenewpage
23 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
24 }
25
26 \AfterEndEnvironment{listing}{%
27 % \unskip\LWR@origvspace*{-\baselineskip}%
28 \LWR@afterendverbatim%
29 }
30
31 \BeforeBeginEnvironment{listingcont}{%
32 \LWR@forcenewpage
33 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
34 }
35
36 \AfterEndEnvironment{listingcont}{%
```

---

```

37 % \unskip\LWR@origvspace*{-\baselineskip}%
38 \LWR@afterendverbatim%
39 }
40
41 \renewcommand{\listinglabel}[1]{\InlineClass{listinglabel}{\the#1}}
42
43 \def\thelisting@line{%
44 \@tempcnta=\listing@line%
45 \divide\@tempcnta\listing@step \multiply\@tempcnta\listing@step%
46 \ifnum\listing@line=\@ne%
47 \listinglabel\listing@line%
48 \else%
49 \ifnum\@tempcnta=\listing@line%
50 \listinglabel\listing@line%
51 \else%
52 \InlineClass{listinglabel}{}%
53 \fi%
54 \fi}
55
56 \LetLtxMacro\LWRMV@@listinginput\@listinginput
57
58 \renewcommand{\@listinginput}[3][ ]{
59 \LWR@forcenewpage
60 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
61 \LWRMV@@listinginput[#1]{#2}{#3}
62 \LWR@afterendverbatim%
63 }
64
65
66 \renewenvironment*{boxedverbatim}
67 {
68 \LWR@forcenewpage
69 \LWR@atbeginverbatim{boxedverbatim}\unskip\LWR@origvspace*{-\baselineskip}%
70 \verbatim%
71 }
72 {
73 \endverbatim%
74 \LWR@afterendverbatim%
75 }
76
77
78 \end{warpHTML}

```

## Package 70

# lwarp-mparhack.sty

## 144 Mparhack

Pkg mparhack Emulated.

for **HTML output:** Discard all options for lwarp-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```

## Package 71

# lwarp-multicol.sty

## 145 Multicol

Pkg multicol multicol is emulated during HTML output, and the multicol package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{multicol}[2015/09/13]

Multicols are converted into a 1–3 column display, browser-supported.

The optional multicols heading is placed inside a <div> of class multicolsheading.

The content is placed inside a <div> of class multicols.

```
2 \begin{warpHTML}
```

```
3 \NewDocumentEnvironment{multicols}{s m o}
```

HTML div class to contain everything:

```
4 {
5 \LWR@forcenewpage
6 \BlockClass{multicols}
```

Optional HTML div class for the heading:

```
7 \IfValueTF{#3}{\begin{BlockClass}{multicolsheading}#3\end{BlockClass}}{}}
```

When done with the environment, close the div:

```
8 {\endBlockClass}
```

Emulated null functions which are not used in HTML:

```
9 \newcommand*{\columnbreak}{}
10 \newcommand*{\RLmulticolcolumns}{}
11 \newcommand*{\LRmulticolcolumns}{}
12
13 \newlength{\premulticols}
14 \newlength{\postmulticols}
15 \newlength{\multicolsep}
16 \newlength{\multicolbaselineskip}
17 \newlength{\multicoltolerance}
```

---

```
18 \newlength{\multicolpretolerance}
19 \newcommand*{\columnseprulecolor}{\normalcolor}
20 \newcounter{columnbadness}
21 \newcounter{finalcolumnbadness}
22 \newcounter{collectmore}
23 \newcounter{unbalance}
24 \newlength{\multicolovershoot}
25 \newlength{\multicolundershoot}

26 \end{warpHTML}
```

## Package 72

# lwarp-multirow.sty

## 146 Multirow

Pkg multirow multirow is emulated during HTML output, and the multirow package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{multirow}

## Package 73

# lwarp-nameref.sty

## 147 Nameref

Pkg nameref nameref is emulated by lwarp.

for HTML output: Discard all options for lwarp-nameref:

```
1 \typeout{Using the lwarp html version of package 'nameref' -- discarding options.}
2 \typeout{   Are not using ProvidesPackage, so that other packages}
3 \typeout{   do not attempt to patch lwarp's version of 'nameref'.}
4 \DeclareOption*{}
5 \ProcessOptions\relax
```



## Package 74

# lwarp-needspace.sty

## 148 Needspace

Pkg `needspace` `needspace` is not used during HTML conversion.

for HTML output: Discard all options for `lwarp-needspace`:

```
1 \LWR@ProvidesPackageDrop{needspace}
2
3 \newcommand*{\needspace}[1]{ }
4 \DeclareDocumentCommand{\Needspace}{s m}{ }
```

## Package 75

# lwarp-newclude.sty

## 149 Newclude

Pkg `newclude` Error if `newclude` is loaded after `lwarp`.

Discard all options for `lwarp-newclude`:

for HTML output: 

```
1 \LWR@ProvidesPackageDrop{newclude}

2 \LWR@loadbefore{newclude}
```

## Package 76

# lwarp-newunicodechar.sty

## 150 Newunicodechar

Pkg newunicodechar Error if newunicodechar is loaded after lwarp.  
Discard all options for lwarp-newunicodechar:

for HTML output: 1 \LWR@ProvidesPackageDrop{newunicodechar}

2 \LWR@loadbefore{newunicodechar}

## Package 77

# lwarp-nextpage.sty

## 151 Nextpage

Pkg nextpage nextpage is nullified.

for HTML output: Discard all options for lwarp-nextpage.

1 \LWR@ProvidesPackageDrop{nextpage}

2 \newcommand{\cleartoevenpage}[1] [] {}

3 \newcommand{\movetoevenpage}[1] [] {}

4 \newcommand{\cleartooddpage}[1] [] {}

5 \newcommand{\movetooddpage}[1] [] {}

## Package 78

# lwarp-nowidow.sty

## 152 Nowidow

Pkg `nowidow` `nowidow` is not used during HTML conversion.

Discard all options for `lwarp-nowidow`:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{nowidow}

2 \newcommand*{\nowidow}[1] [] {}
3 \newcommand*{\setnowidow}[1] [] {}

4 \newcommand*{\noclub}[1] [] {}
5 \newcommand*{\setnoclub}[1] [] {}
```

## Package 79

# lwarp-ntheorem.sty

## 153 Ntheorem

(Based on original code by WOLFGANG MAY, ANDREAS SCHEDLER.)

Pkg `ntheorem` `ntheorem` is patched for use by `lwarp`.

### CSS styling of theorems and proofs:

**Theorem:** `<div>` of class `theorembody<theoremstyle>`

**Theorem Header:** `<span>` of class `theoremheader<style>`

where `<theoremstyle>` is `plain`, `break`, etc.

⚠ **Font control** This conversion is not total. Font control is via CSS, and the custom L<sup>A</sup>T<sub>E</sub>X font settings are ignored.

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

Options `amsthm` or `standard` choose which set of theorems and proofs to initialize.

⚠ **Disabled options** The options `thmmarks` and `amsmath` are disabled, since they heavily modify the underlying math code. Theorem marks are emulated. The AMS-math modifications are not done.

Option `thref` is disabled because `cleveref` functions are used instead. `\thref` is emulated.

Option `hyperref` is disabled because `lwarp` emulated `hyperref`.

for HTML output: Some disabled options:

```
1 \DeclareOption{thref}{}
2
3
4 \newbool{LWR@ntheoremmarks}
5 \boolfalse{LWR@ntheoremmarks}
6
7 \DeclareOption{thmmarks}{
```

```

8 \booltrue{LWR@theoremmarks}
9 \newif\ifsetendmark\setendmarktrue
10 }
11
12
13 \newbool{LWR@theoremamsthm}
14 \boolfalse{LWR@theoremamsthm}
15
16 \DeclareOption{amsthm}{\booltrue{LWR@theoremamsthm}}
17
18
19 \DeclareOption{amsmath}{}
20 \DeclareOption{hyperref}{}
21
22
23 \LWR@ProvidesPackagePass{theorem}

```

Storage for the style being used for new theorems.

```
24 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```

25 \gdef\theoremstyle#1{%
26   \ifundefined{th@#1}{\@warning
27     {Unknown theoremstyle ‘#1’. Using ‘plain’}%
28     \theorem@style{plain}
29     \renewcommand{\LWR@newtheoremstyle}{plain}% new
30   }%
31   {
32     \theorem@style{#1}
33     \renewcommand{\LWR@newtheoremstyle}{#1}% new
34   }
35 }

```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

36
37 \gdef\xnthm#1#2[#3]{%
38   \ifthm@tempif
39     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
40     \expandafter\@ifundefined{c@#1}%
41       {\@definecounter{#1}}{}%
42     \@newctr{#1}[#3]%
43     \expandafter\xdef\csname the#1\endcsname{%
44       \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
45       {\noexpand\csname\the\theoremnumbering\endcsname{#1}}}%
46     \expandafter\gdef\csname mkheader@#1\endcsname

```

```

47     {\csname setparms@#1\endcsname
48       \@thm{#1}{#1}{#2}
49     }%
50   \global\@namedef{end#1}{\@endtheorem}
51   \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
52   \fi
53 }
54
55 \gdef\@ynthm#1#2{%
56   \ifthm@tempif
57     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
58     \expandafter\@ifundefined{c@#1}%
59       {\@definecounter{#1}}{}%
60     \expandafter\xdef\csname the#1\endcsname
61       {\noexpand\csname\the\theoremnumbering\endcsname{#1}}%
62     \expandafter\gdef\csname mkheader@#1\endcsname
63       {\csname setparms@#1\endcsname
64         \@thm{#1}{#1}{#2}
65       }%
66     \global\@namedef{end#1}{\@endtheorem}
67     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
68   \fi
69 }
70
71 \gdef\@othm#1[#2]#3{%
72   \@ifundefined{c@#2}{\@nocounterr{#2}}%
73   {\ifthm@tempif
74     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
75     \global\@namedef{the#1}{\@nameuse{the#2}}%
76     \expandafter\protected@xdef\csname num@addtheoremline#1\endcsname{%
77       \noexpand\@num@addtheoremline{#1}{#3}}%
78     \expandafter\protected@xdef\csname nonum@addtheoremline#1\endcsname{%
79       \noexpand\@nonum@addtheoremline{#1}{#3}}%
80     \theoremkeyword{#3}%
81     \expandafter\protected@xdef\csname #1Keyword\endcsname
82       {\the\theoremkeyword}%
83     \expandafter\gdef\csname mkheader@#1\endcsname
84       {\csname setparms@#1\endcsname
85         \@thm{#1}{#2}{#3}
86       }%
87     \global\@namedef{end#1}{\@endtheorem}
88     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
89   \fi}
90 }

```

Mimics a float by incrementing the float counter and generating an HTML anchor. These are used for list-of-theorem cross-references.

```

91 \newcommand{\LWR@inctheorem}{%

```

```

92 \addtocounter{LWR@thisfloat}{1}%
93 \LWR@stoppars%
94 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
95 \LWR@startpars%
96 }

```

The following are patched for CSS.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc. They are gathered together here.

Each theorem is encased in a `BlockClass` environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader<style>`.

```

97 \gdef\newtheoremstyle#1#2#3{%
98   \expandafter\@ifundefined{th@#1}%
99   {\expandafter\gdef\csname th@#1\endcsname{%
100     \def\@begintheorem####1####2{%
101 \LWR@forcenewpage% new
102 \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
103 \LWR@inctheorem% new
104 #2}%
105   \def\@opargbegintheorem####1####2####3{%
106 \LWR@forcenewpage% new
107 \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
108 \LWR@inctheorem% new
109 #3}%
110 }%
111 }%
112 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
113 }
114
115 \renewtheoremstyle{plain}%
116   {\item[\hskip\labelsep \theorem@headerfont
117     \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]]}%
118   {\item[\hskip\labelsep \theorem@headerfont
119     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]]}
120
121 \renewtheoremstyle{break}%
122   {\item[
123 % \rlap{\vbox{\hbox{
124   \hskip\labelsep \theorem@headerfont
125   \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
126 % } \hbox{\strut}}]
127   ]}%
128   {\item[
129 % \rlap{\vbox{\hbox{
130   \hskip\labelsep \theorem@headerfont
131   \InlineClass{theoremheaderbreak}{##1\ ##2\ (##3)\theorem@separator}\newline

```

```

132 % } \hbox{\strut}}
133   ]}
134
135 \renewtheoremstyle{change}%
136   {\item[\hskip\labelsep
137     \theorem@headerfont
138     \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]]}%
139   {\item[\hskip\labelsep
140     \theorem@headerfont
141     \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]]}
142
143 \renewtheoremstyle{changebreak}%
144   {\item[
145 %   \rlap{\vbox{\hbox{
146           \hskip\labelsep \theorem@headerfont
147           \InlineClass{theoremheaderchangebreak}{##2\ ##1\theorem@separator}\newline
148 %   } \hbox{\strut}}}}
149   ]}%
150   {\item[
151 %   \rlap{\vbox{\hbox{
152           \hskip\labelsep \theorem@headerfont
153           \InlineClass{theoremheaderchangebreak}{##2\ ##1\ (##3)\theorem@separator}\newline
154 %   } \hbox{\strut}}}}
155   ]}
156
157 \renewtheoremstyle{margin}%
158   {\item[\hskip\labelsep\theorem@headerfont
159     \InlineClass{theoremheadermargin}{##2 \quad ##1\theorem@separator}
160   ]}%
161   {\item[\hskip\labelsep\theorem@headerfont
162     \InlineClass{theoremheadermargin}{##2 \quad ##1\ (##3)\theorem@separator}
163   ]}
164
165 \renewtheoremstyle{marginbreak}%
166   {\item[\hskip\labelsep\theorem@headerfont
167     \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\theorem@separator}\newline
168   ]}%
169   {\item[\hskip\labelsep\theorem@headerfont
170     \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\ (##3)\theorem@separator}\newline
171   ]}
172
173 \renewtheoremstyle{nonumberplain}%
174   {\item[\theorem@headerfont\hskip\labelsep
175     \InlineClass{theoremheaderplain}{##1\theorem@separator}]]}%
176   {\item[\theorem@headerfont\hskip \labelsep
177     \InlineClass{theoremheaderplain}{##1\ (##3)\theorem@separator}]]}
178
179 \renewtheoremstyle{nonumberbreak}%
180   {\item[
181 %   \rlap{\vbox{\hbox{

```



```

182 \hskip\labelsep \theorem@headerfont
183 \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline
184 % } \hbox{\strut}}
185 ]}%
186 {\item[
187 % \rlap{\vbox{\hbox{
188 \hskip\labelsep \theorem@headerfont
189 \InlineClass{theoremheaderbreak}{##1\ (##3)\theorem@separator}\newline
190 % } \hbox{\strut}}
191 ]}
192
193 \renewtheoremstyle{empty}%
194 {\item[]}%
195 {\item[\theorem@headerfont \hskip\labelsep\relax
196 \InlineClass{theoremheaderplain}{##3}]}
197
198 \renewtheoremstyle{emptybreak}%
199 {\item[]}%
200 {\item[\theorem@headerfont \hskip\labelsep\relax
201 \InlineClass{theoremheaderplain}{##3}] \ \newline}

```

The following manually adjust the CSS for the standard configuration objects which are not a purely plain style:

```

202 \ifbool{LWR@theoremamsthm}{-}{%
203 % upright text via CSS
204 \newtheoremstyle{plainupright}%
205 {\item[\hskip\labelsep \theorem@headerfont
206 \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
207 {\item[\hskip\labelsep \theorem@headerfont
208 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
209
210 % upright text and small caps header via CSS
211 \newtheoremstyle{nonumberplainuprightsc}%
212 {\item[\theorem@headerfont\hskip\labelsep
213 \InlineClass{theoremheadersc}{##1\theorem@separator}]}%
214 {\item[\theorem@headerfont\hskip \labelsep
215 \InlineClass{theoremheadersc}{##1\ (##3)\theorem@separator}]}

```

The following standard configuration is renewed using the new CSS:

```

216 \theoremstyle{plainupright}
217 \theorembodyfont{\upshape}
218 \theoremsymbol{\HTMLUnicode{25A1}}% UTF-8 white box
219 \renewtheorem{Example}{Example}
220 \renewtheorem{example}{Example}
221 \renewtheorem{Beispiel}{Beispiel}
222 \renewtheorem{beispiel}{Beispiel}
223 \renewtheorem{Bemerkung}{Bemerkung}

```

```

224 \renewtheorem{bemerkung}{Bemerkung}
225 \renewtheorem{Anmerkung}{Anmerkung}
226 \renewtheorem{anmerkung}{Anmerkung}
227 \renewtheorem{Remark}{Remark}
228 \renewtheorem{remark}{Remark}
229 \renewtheorem{Definition}{Definition}
230 \renewtheorem{definition}{Definition}
231
232 \theoremstyle{nonumberplainuprightsc}
233 \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
234 \renewtheorem{Proof}{Proof}
235 \renewtheorem{proof}{Proof}
236 \renewtheorem{Beweis}{Beweis}
237 \renewtheorem{beweis}{Beweis}
238 \qedsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
239
240 \theoremsymbol{}
241 }% not amsthm

```

Only if the `amsthm` option was given:

```

242 \ifbool{LWR@theoremamsthm}{
243
244 \gdef\th@plain{%
245 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
246 \def\@begintheorem##1##2{%
247 \LWR@forcenewpage% new
248 \BlockClass{theorembodyplain}% new
249 \LWR@intheorem% new
250 \item[\hskip\labelsep
251 % \theorem@headerfont
252 \InlineClass{theoremheaderplain}{##1\ ##2.}
253 ]}%
254 \def\@opargbegintheorem##1##2##3{%
255 \LWR@forcenewpage% new
256 \BlockClass{theorembodyplain}% new
257 \LWR@intheorem% new
258 \item[\hskip\labelsep
259 % \theorem@headerfont
260 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
261 ]}}
262
263 \gdef\th@nonumberplain{%
264 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
265 \def\@begintheorem##1##2{%
266 \LWR@forcenewpage% new
267 \BlockClass{theorembodyplain}% new
268 \LWR@intheorem% new
269 \item[\hskip\labelsep
270 % \theorem@headerfont

```

```

271 \InlineClass{theoremheaderplain}{##1.}
272 }}%
273 \def\@opargbegintheorem##1##2##3{%
274 \LWR@forcenewpage% new
275 \BlockClass{theorembodyplain}% new
276 \LWR@inctheorem% new
277 \item[\hskip\labelsep
278 % \theorem@headerfont
279 \InlineClass{theoremheaderplain}{##1\ (##3).}
280 ]}}
281
282 \gdef\th@definition{%
283 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
284 \def\@begintheorem##1##2{%
285 \LWR@forcenewpage% new
286 \BlockClass{theorembodydefinition}% new
287 \LWR@inctheorem% new
288 \item[\hskip\labelsep
289 % \theorem@headerfont
290 \InlineClass{theoremheaderdefinition}{##1\ ##2.}
291 ]}%
292 \def\@opargbegintheorem##1##2##3{%
293 \LWR@forcenewpage% new
294 \BlockClass{theorembodydefinition}% new
295 \LWR@inctheorem% new
296 \item[\hskip\labelsep
297 % \theorem@headerfont
298 \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
299 ]}}
300
301 \gdef\th@nonumberdefinition{%
302 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
303 \def\@begintheorem##1##2{%
304 \LWR@forcenewpage% new
305 \BlockClass{theorembodydefinition}% new
306 \LWR@inctheorem% new
307 \item[\hskip\labelsep
308 % \theorem@headerfont
309 \InlineClass{theoremheaderdefinition}{##1.}
310 ]}%
311 \def\@opargbegintheorem##1##2##3{%
312 \LWR@forcenewpage% new
313 \BlockClass{theorembodydefinition}% new
314 \LWR@inctheorem% new
315 \item[\hskip\labelsep
316 % \theorem@headerfont
317 \InlineClass{theoremheaderdefinition}{##1\ (##3).}
318 ]}}
319
320 \gdef\th@remark{%

```

```

321 \def\theorem@headerfont{\itshape}\normalfont%
322 \def\@begintheorem##1##2{%
323 \LWR@forcenewpage% new
324 \BlockClass{theorembodyremark}% new
325 \LWR@inctheorem% new
326 \item[\hskip\labelsep
327 % \theorem@headerfont
328 \InlineClass{theoremheaderremark}{##1\ ##2.}
329 ]}%
330 \def\@opargbegintheorem##1##2##3{%
331 \LWR@forcenewpage% new
332 \BlockClass{theorembodyremark}% new
333 \LWR@inctheorem% new
334 \item[\hskip\labelsep
335 % \theorem@headerfont
336 \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
337 ]}}
338
339 \gdef\th@nonumberremark{%
340 \def\theorem@headerfont{\itshape}\normalfont%
341 \def\@begintheorem##1##2{%
342 \LWR@forcenewpage% new
343 \BlockClass{theorembodyremark}% new
344 \LWR@inctheorem% new
345 \item[\hskip\labelsep
346 % \theorem@headerfont
347 \InlineClass{theoremheaderremark}{##1.}
348 ]}%
349 \def\@opargbegintheorem##1##2##3{%
350 \LWR@forcenewpage% new
351 \BlockClass{theorembodyremark}% new
352 \LWR@inctheorem% new
353 \item[\hskip\labelsep
354 % \theorem@headerfont
355 \InlineClass{theoremheaderremark}{##1\ (##3).}
356 ]}}
357
358 \gdef\th@proof{%
359 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
360 \def\@begintheorem##1##2{%
361 \LWR@forcenewpage% new
362 \BlockClass{theorembodyproof}% new
363 \LWR@inctheorem% new
364 \item[\hskip\labelsep
365 % \theorem@headerfont
366 \InlineClass{theoremheaderproof}{##1.}
367 ]}%
368 \def\@opargbegintheorem##1##2##3{%
369 \LWR@forcenewpage% new
370 \BlockClass{theorembodyproof}% new

```

---

```

371      \LWR@inctheorem% new
372      \item[\hskip\labelsep
373 % \theorem@headerfont
374 \InlineClass{theoremheaderproof}{##1\ (##3).}
375 ]}}
376
377
378
379 \newcounter{proof}%
380 \if@thmmarks
381     \newcounter{currproofctr}%
382     \newcounter{endproofctr}%
383 \fi
384
385 \gdef\proofSymbol{\openbox}
386
387 \newcommand{\proofname}{Proof}
388
389 \newenvironment{proof}[1][\proofname]{
390     \th@proof
391     \def\theorem@headerfont{\itshape}%
392     \normalfont
393     \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
394     \@thm{proof}{proof}{#1}
395 }%
396 {\@endtheorem}
397
398 }{}% amsthm option

```

Patched for CSS:

```

399 \let\LWR@origendtheorem\@endtheorem
400 \renewcommand{\@endtheorem}{%
401 \ifbool{LWR@theoremmarks}{%
402     \ifsetendmark%
403     \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
404     \setendmarkfalse%
405     \fi%
406 }{}%
407 \LWR@origendtheorem%
408 \ifbool{LWR@theoremmarks}{\global\setendmarktrue}{}%
409 \endBlockClass%
410 }

411 \gdef\NoEndMark{\global\setendmarkfalse}

```

Redefined to reuse the float mechanism to add list-of-theorem links:

`\thm@thmline {⟨1: printed type⟩} {⟨2: #⟩} {⟨3: optional⟩} {⟨4: page⟩}`

```
412 \renewcommand{\thm@@thmline@noname}[4]{%
413 \hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
414 }
415
416 \renewcommand{\thm@@thmline@name}[4]{%
417 \hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
418 }
```

This was redefined by `ntheorem` when loaded, so it is now redefined for `lwarp`:

```
419 \def\thm@@thmline{\thm@@thmline@name}
```

Patch for CSS:

```
420 \def\listtheorems#1{
421 \LWR@html@elementclass{nav}{lothm}%
422 \begingroup
423 \c@tocdepth=-2%
424 \def\thm@list{#1}\thm@processlist
425 \endgroup
426 \LWR@html@elementclassend{nav}{lothm}%
427 }
```

Proof QED symbol:

```
428
429 \newcommand{\qed}{\qquad\the\qedsymbol}
430
431 \AtBeginDocument{
432 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
433 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
434 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
435 }
```

`\thref {⟨label⟩}`

```
436 \newcommand*{\thref}[1]{\cref{#1}}
```

## Package 80

# lwarp-pagenote.sty

## 154 Pagenote

Pkg pagenote pagenote works as-is.

It is only included as an `lwarp-pagenote.sty` file because past versions of `lwarp` used `pagenote` to emulate footnotes, and so the file may exist on current installations, and should be over-written by this newer version.

for HTML output: 1 \LWR@ProvidesPackagePass{pagenote}

## Package 81

# lwarp-paralist.sty

## 155 Paralist

Pkg paralist paralist is supported with minor changes.

for HTML output: 1 \LWR@ProvidesPackagePass{paralist}

for HTML output: 2 \AtBeginEnvironment{compactitem}{\LWR@itemizestart}  
 3 \AtEndEnvironment{compactitem}{\LWR@itemizeend}  
 4 \AtBeginEnvironment{compactenum}{\LWR@enumeratestart}  
 5 \AtEndEnvironment{compactenum}{\LWR@enumerateend}  
 6 \AtBeginEnvironment{compactdesc}{\LWR@descriptionstart}  
 7 \AtEndEnvironment{compactdesc}{\LWR@descriptionend}  
 8 \def\paradescriptionlabel#1{\normalfont\textbf{#1}}

## Package 82

# lwarp-parskip.sty

## 156 Parskip

Pkg `parskip` `parskip` is ignored.

for **HTML output:** Discard all options for `lwarp-parskip`.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

## Package 83

# lwarp-pdfscape.sty

## 157 Pdfscape

Pkg `pdfscape` `Emulated`.

for **HTML output:** Discard all options for `lwarp-pdfscape`:

```
1 \LWR@ProvidesPackageDrop{pdfscape}
```



## Package 84

# lwarp-pdftsync.sty

## 158 Pdftsync

Pkg pdftsync Emulated.

for HTML output: Discard all options for lwarp-pdftsync:

```
1 \LWR@ProvidesPackageDrop{pdftsync}

2 \let\pdftsync\relax
3 \let\pdftsyncstart\relax
4 \let\pdftsyncstop\relax
```

## Package 85

# lwarp-placeins.sty

## 159 Placeins

Pkg placeins placeins is not used during HTML conversion.

Discard all options for lwarp-placeins:

for HTML output: 

```
1 \LWR@ProvidesPackageDrop{placeins}

2 \newcommand*{\FloatBarrier}{}

```

## Package 86

# lwarp-prelim2e.sty

## 160 Prelim2e

Pkg `prelim2e` Emulated.

for **HTML output**: Discard all options for `lwarp-prelim2e`:

```
1 \LWR@ProvidesPackageDrop{prelim2e}

2 \newcommand{\PrelimText}{}
3 \newcommand{\PrelimTextStyle}{}
4 \newcommand{\PrelimWords}{}

```

## Package 87

# lwarp-ragged2e.sty

## 161 Ragged2e

Pkg ragged2e ragged2e is not used during HTML conversion.

Discard all options for lwarp-ragged2e:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{ragged2e}

2 \newcommand*{\Centering}{\centering}
3 \newcommand*{\RaggedLeft}{\raggedleft}
4 \newcommand*{\RaggedRight}{\raggedright}
5 \newcommand*{\justifying}{\}
6 \newlength{\CenteringLeftskip}
7 \newlength{\RaggedLeftLeftskip}
8 \newlength{\RaggedRightLeftskip}
9 \newlength{\CenteringRightskip}
10 \newlength{\RaggedLeftRightskip}
11 \newlength{\RaggedRightRightskip}
12 \newlength{\CenteringParfillskip}
13 \newlength{\RaggedLeftParfillskip}
14 \newlength{\RaggedRightParfillskip}
15 \newlength{\JustifyingParfillskip}
16 \newlength{\CenteringParindent}
17 \newlength{\RaggedLeftParindent}
18 \newlength{\RaggedRightParindent}
19 \newlength{\JustifyingParindent}
20 \newenvironment*{Center}{\center}{\endcenter}
21 \newenvironment*{FlushLeft}{\flushleft}{\endflushleft}
22 \newenvironment*{FlushRight}{\flushright}{\endflushright}
23 \newenvironment*{justify}{\justifying}{\endjustifying}

```

## Package 88

# lwarp-rotating.sty

## 162 Rotating

Pkg rotating rotating is emulated during HTML output, and the rotating package is ignored.

All rotations are ignored in HTML output.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{rotating}

2 \LetLtxMacro\sidewaystable\table
3 \let\endsidewaystable\endtable
4
5 \LetLtxMacro\sidewaysfigure\figure
6 \let\endsidewaysfigure\endfigure
7
8 \newenvironment*{sideways}{}{}
9 \newenvironment*{turn}[1]{}{}
10 \newenvironment*{rotate}[1]{}{}
11 \NewDocumentCommand{\turnbox}{m +m}{#2}
12 \let\rotcaption\caption
13 \let\@makerotcaption\@makecaption

```

## Package 89

# lwarp-rotfloat.sty

## 163 Rotfloat

Pkg rotfloat rotfloat is emulated during HTML output, and the rotfloat package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{rotfloat}
2
3 \RequirePackage{float}
```

`\newfloat`  $\{\langle 1: type \rangle\} \{\langle 2: placement \rangle\} \{\langle 3: ext \rangle\} [\langle 4: within \rangle]$

Emulates the `\newfloat` command from the float package. Sideways floats are `\let` to the same as regular floats.

“placement” is ignored.

```

4 \RenewDocumentCommand{\newfloat}{m m m o}{%
5 \IfValueTF{#4}
6 {
7   \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}
8 }
9 {
10  \DeclareFloatingEnvironment[fileext=#3]{#1}
11  \DeclareFloatingEnvironment[fileext=#3]{sideways#1}
12 }
13 \csletcs{sideways#1}{#1}
14 \csletcs{endsideways#1}{end#1}
```

`newfloat` package automatically creates the `\listof` command for new floats, but float does not, so remove `\listof` here in case it is manually created later.

```

15 \cslet{listof#1s}\relax
16 \cslet{listof#1es}\relax
17 }
```

## Package 90

# lwarp-savetrees.sty

## 164 Savetrees

Pkg **savetrees** Emulated.

for **HTML output**: Discard all options for lwarp-savetrees:

```
1 \LWR@ProvidesPackageDrop{savetrees}
```

## Package 91

# lwarp-setspace.sty

## 165 Setspace

Pkg    **setspace**    setspace is not used during HTML conversion.

Discard all options for lwarp-setspace:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{setspace}
2
3 \newcommand*{\setstretch}[1]{}
4 \newcommand*{\SetSinglespace}[1]{}
5 \newcommand*{\singlespacing}{}
6 \newcommand*{\onehalfspacing}{}
7 \newcommand*{\doublespacing}{}
8
9 \newenvironment*{singlespace}
10 {
11 \LWR@forcenewpage
12 \BlockClass{singlespace}
13 }
14 {\endBlockClass}
15
16 \newenvironment*{singlespace*}
17 {
18 \LWR@forcenewpage
19 \BlockClass{singlespace}
20 }
21 {\endBlockClass}
22
23 \newenvironment*{spacing}[1]{
24
25 }{
26
27 }
28
29 \newenvironment*{onehalfspace}
30 {
31 \LWR@forcenewpage
32 \BlockClass{onehalfspace}
33 }
34 {\endBlockClass}
35
36 \newenvironment*{doublespace}
37 {

```

```
38 \LWR@forcenewpage
39 \BlockClass{doublespace}
40 }
41 {\endBlockClass}
```

## Package 92

# lwarp-shadow.sty

## 166 Shadow

Pkg shadow shadow is emulated.

for HTML output: Discard all options for lwarp-shadow:

```
1 \LWR@ProvidesPackageDrop{shadow}

2 \newdimen\sboxsep
3 \newdimen\sboxrule
4 \newdimen\sdim
5
6 \newcommand{\shabox}[1]{%
7 \InlineClass{shabox}{#1}%
8 }
```



## Package 93

# lwarp-showidx.sty

## 167 Showidx

Pkg `showidx` `showidx` is ignored.

for **HTML output**: Discard all options for `lwarp-showidx`:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

## Package 94

# lwarp-showkeys.sty

## 168 Showkeys

Pkg `showkeys` `showkeys` is ignored.

for **HTML output**: Discard all options for `lwarp-showkeys`:

```
1 \LWR@ProvidesPackageDrop{showkeys}
```

```
2 \NewDocumentCommand{\showkeys}{s}{}
```

## Package 95

# lwarp-sidecap.sty

## 169 Sidecap

Pkg sidecap sidecap is nullified.

for HTML output: Discard all options for lwarp-sidecap.

```
1 \LWR@ProvidesPackageDrop{sidecap}
```

See:

<http://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment>  
regarding the creation of starred environments with xparse.

```
2 \NewDocumentEnvironment{SCtable}{soo}
3 {\IfValueTF{#3}{\table[#3]}{\table}}
4 {\endtable}
5
6 \ExplSyntaxOn
7 \cs_new:cpn {SCtable*} {\SCtable*}
8 \cs_new_eq:cN {endSCtable*} \endSCtable
9 \ExplSyntaxOff
10
11
12 \NewDocumentEnvironment{SCfigure}{soo}
13 {\IfValueTF{#3}{\figure[#3]}{\figure}}
14 {\endfigure}
15
16 \ExplSyntaxOn
17 \cs_new:cpn {SCfigure*} {\SCfigure*}
18 \cs_new_eq:cN {endSCfigure*} \endSCfigure
19 \ExplSyntaxOff
20
21
22 \newenvironment*{wide}{}{}
```

## Package 96

# lwarp-sidenotes.sty

## 170 Sidenotes

(Based on original code by ANDY THOMAS, OLIVER SCHEBAUM.)

Pkg sidenotes Patched for lwarp.

for HTML output: Load the original package:

```
1 \LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with lwarp:

Stop paragraph handling while creating the caption:

```
2 \RenewDocumentCommand \sidecaption {s o o m}
3 {
4   \LWR@stoppars
5   \captionsetup{style=sidecaption}
6   \IfBooleanTF{#1}
7   { % starred
8     \IfNoValueOrEmptyTF{#2}
9     {\marginnote{\caption*{#4}}}
10    {\marginnote{\caption*{#4}}[#2]}
11  }
12  { % unstarred
13    \IfNoValueOrEmptyTF{#2}
14    {\def\@sidenotes@sidecaption@tof{#4}}
15    {\def\@sidenotes@sidecaption@tof{#2}}
16    \IfNoValueOrEmptyTF{#3}
17    {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}}
18    {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}[#3]}
19  }
20  \LWR@startpars
21 }
```

Borrowed from the lwarp version of keyfloat:

```
22 \NewDocumentEnvironment{KFLT sidenotes @marginfloat}{0{-1.2ex} m}
23 {% start
24 \LWR@maybeinthisfloat%
25 \LWR@forcenewpage
26 \LWR@stoppars%
27 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}}{}
```

---

```

28 \LWR@startpars%
29 \captionsetup{type=#2}%
30 }
31 {
32 \LWR@htmldivclassend{div}
33 }
34
35 \RenewDocumentEnvironment{marginfigure}{o}
36   {\begin{KFLTsidenotes@marginfloat}{figure}}
37   {\end{KFLTsidenotes@marginfloat}}
38
39 \RenewDocumentEnvironment{margintable}{o}
40   {\begin{KFLTsidenotes@marginfloat}{table}}
41   {\end{KFLTsidenotes@marginfloat}}

```

The following were changed by sidenotes, and now are reset back to their lwarp-supported originals:

Restoring the definition from the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> `article.cls` source:

```

42 \renewenvironment{figure*}
43     {\@dblfloat{figure}}
44     {\end@dblfloat}
45
46 \renewenvironment{table*}
47     {\@dblfloat{table}}
48     {\end@dblfloat}

```

## Package 97

# lwarp-soul.sty

## 171 Soul

(Based on original code by MELCHIOR FRANZ.)

Pkg soul Emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{soul}
2 \RequirePackage{xcolor}% for \convertcolorspec
```

Storage for the colors to use:

```
3 \newcommand*\LWR@soululcolor{}
4
5 \newcommand*\LWR@soulstcolor{}
6
7 % \definecolor{LWR@soulhlcolordefault}{HTML}{F8E800}
8 % \newcommand*\LWR@soulhlcolor{LWR@soulhlcolordefault}
9 \newcommand*\LWR@soulhlcolor{}
```

Basic markup with CSS:

```
10 \newcommand{\so}[1]{\InlineClass{letterspacing}{#1}}
11 \newcommand{\caps}[1]{\InlineClass{capsspacing}{#1}}
```

Add colors if not empty:

```
12 \newcommand{\LWR@soulcolor}[4]{%
13 \ifcsempy{#2}%
14 {%
15 \InlineClass{#3}{#1}}%
16 {%
17 \convertcolorspec{named}{\csuse{#2}}{HTML}\LWR@tempcolor%
18 \InlineClass[#4: \#\LWR@tempcolor]{#3}{#1}%
19 }%
20 }
21
22 \newcommand{\ul}[1]{%
23 \LWR@soulcolor{#1}{LWR@soululcolor}{uline}{text-decoration-color}%
24 }
25
26 \newcommand{\st}[1]{
27 \LWR@soulcolor{#1}{LWR@soulstcolor}{sout}{text-decoration-color}%
```

```

28 }
29
30 \newcommand{\hl}[1]{
31 \LWR@soulcolor{#1}{\LWR@soulhlcolor}{highlight}{background-color}%
32 }

```

Nullified:

```

33 \newcommand*\soulaccent[1]{}
34 \newcommand*\soulregister[2]{}
35 \newcommand*\sloppyword[1]{#1}
36 \newcommand*\sodef[5]{\DeclareRobustCommand*#1[1]{\so{##1}}}
37 \newcommand*\resetso{}
38 \newcommand*\capsdef[5]{}
39 \newcommand*\capsreset{}
40 \newcommand*\capssave[1]{}
41 \newcommand*\capsselect[1]{}
42 \newcommand*\setul[2]{}
43 \newcommand*\resetul{}
44 \newcommand*\setuldepth[1]{}
45 \newcommand*\setuloverlap[1]{}

```

Set colors:

```

46 \newcommand*\setulcolor[1]{\renewcommand{\LWR@soululcolor}{#1}}
47 \newcommand*\setstcolor[1]{\renewcommand{\LWR@soulstcolor}{#1}}
48 \newcommand*\sethlcolor[1]{\renewcommand{\LWR@soulhlcolor}{#1}}

```

Long versions of the user-level macros:

```

49 \let\textso\so
50 \let\textul\ul
51 \let\texthl\hl
52 \let\textcaps\caps

```


## Package 98

# lwarp-subfig.sty

## 172 Subfig

(Based on original code by STEVEN DOUGLAS COCHRAN.)

Pkg subfig subfig is supported and patched by lwarp.

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

horizontal spacing In the document source, use \hfill and \hspace\* between subfigures to spread them apart horizontally. The use of other forms of whitespace may cause paragraph tags to be generated, resulting in subfigures appearing on the following lines instead of all on a single line.

for HTML output: Accept all options for lwarp-subfig:

```
1 \LWR@ProvidesPackagePass{subfig}
```

```
\sf@@@subfloat {<1 type>} [<2 lof entry>] [<3 caption>] {<4 contents>}
```

The outer minipage allows side-by-side subfloats with \hfill between.

```
2 \long\def\sf@@@subfloat#1[#2][#3]#4{%
3 \begin{minipage}{\linewidth}% new
4 \LWR@stoppars% new
5   \ifundefined{FBsc@max}{}%
6     {\FB@readaux{\let\FBsuboheight\relax}}%
7   \@tempcnta=\@ne
8   \if@minipage
9     \@tempcnta=\z@
10  \else\ifdim \lastskip=\z@ \else
11    \@tempcnta=\tw@
12  \fi\fi
13  \ifmaincaptiontop
14    \sf@top=\sf@nearskip
15    \sf@bottom=\sf@farskip
16  \else
17    \sf@top=\sf@farskip
18    \sf@bottom=\sf@nearskip
19  \fi
20  \leavevmode
21  \setbox\@tempboxa \hbox{#4}%
22  \@tempdima=\wd\@tempboxa
```

```

23   \@ifundefined{FBsc@max}{}%
24   {\global\advance\Xhsize-\wd\@tempboxa
25     \dimen@=\ht\@tempboxa
26     \advance\dimen@\dp\@tempboxa
27     \ifdim\dimen@>\FBso@max
28       \global\FBso@max\dimen@
29     \fi}%
30   \vtop\bgroup
31     \vbox\bgroup
32       \ifcase\@tempcnta
33       \@minipagefalse
34       \or
35       \vskip\sf@top
36       \or
37       \ifdim \lastskip=\z@ \else
38         \@tempskipb\sf@top\relax\@xaddvskip
39       \fi
40     \fi
41     \sf@ifpositiontop{%
42       \ifx \@empty#3\relax \else
43         \sf@subcaption{#1}{#2}{#3}%
44         \vskip\sf@capskip
45         \vskip\sf@captopadj
46       \fi\egroup
47       \hrule width0pt height0pt depth0pt
48       \LWR@startpars% new
49 %   \box\@tempboxa
50       #4
51       \LWR@stoppars% new
52     }{%
53       \LWR@startpars% new
54       \@ifundefined{FBsc@max}{%
55         {
56 %   \box\@tempboxa
57         #4
58         }%
59         {\ifx\FBsuboheight\relax
60 %       \box\@tempboxa
61         #4
62         \else
63 %       \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
64         #4
65         \fi}%
66       \LWR@stoppars% new
67       \egroup
68       \ifx \@empty#3\relax \else
69         \vskip\sf@capskip
70         \hrule width0pt height0pt depth0pt
71         \sf@subcaption{#1}{#2}{#3}%
72       \fi

```



```

73      }%
74      \vskip\sf@bottom
75      \egroup
76      \@ifundefined{FBsc@max}{}%
77      {\addtocounter{FRobj}{-1}%
78       \ifnum\c@FRobj=0\else
79       \subfloatrowsep
80       \fi}%
81      \ifmaincaptiontop\else
82      \global\advance\@nameuse{c@\@capttype}\m@ne
83      \fi
84 \end{minipage}% new
85 \LWR@startpars% new
86 \endgroup\ignorespaces%
87 }%

\sf@subcaption {\langle 1 type\rangle} {\langle 2 lof entry\rangle} {\langle 3 caption\rangle}

88 \long\def\sf@subcaption#1#2#3{%
89 \LWR@stoppars% new
90 \ifx \relax#2\relax \else
91   \bgroup
92   \let\label=\@gobble
93   \let\protect=\string
94   \def\@subcaplabel{%
95     \caption@lstfmt{\@nameuse{p#1}}{\@nameuse{the#1}}}%
96     \sf@updatecaptionlist{#1}{#2}{\the\value{\@capttype}}{\the\value{#1}}%
97   \egroup
98   \fi
99   \bgroup
100   \ifx \relax#3\relax
101     \let\captionlabelsep=\relax
102   \fi
103 %     \setbox0\vbox{%
104 %       \hb@xt@\the\@tempdima{%
105 %
106 % %           \hss
107 % %           \parbox[t]{\the\@tempdima}{%
108 % %             \caption@make
109 % %               {\@nameuse{sub\@capttype name}}}%
110 % %             {\@nameuse{thesub\@capttype}}}%
111 % %             {\#3}
112 % %   }%
113 % %           \hss
114 % %   }
115 % }%
116 \@ifundefined{FBsc@max}%
117 %   {\box0}%
118 %   {
119 % \parbox[t]{\the\@tempdima}{%

```

```

120 \LWR@traceinfo{sfsubcap B1}% new
121     \LWR@htmlblocktag{figcaption}% new
122     \caption@make
123         {\@nameuse{sub\@captype name}}}%
124         {\@nameuse{thesub\@captype}}}%
125         {#3}
126     \LWR@htmlblocktag{/figcaption}% new
127 \LWR@traceinfo{sfsubcap B2}% new
128 % }%
129     }%
130     {\dimen@\ht0%
131     \advance\dimen@\dp0%
132     \ifdim\dimen@>\FBsc@max
133     \global\FBsc@max\dimen@
134     \fi
135     \FB@readaux{\let\FBsubcheight\relax}%
136     \ifx\FBsubcheight\relax
137     \def\next{
138 % \parbox[t]{\the\@tempdima}
139     }%
140     \else
141     \def\next{
142 % \parbox[t][\FBsubcheight][t]{\the\@tempdima}
143     }%
144     \fi
145     \vbox{%
146 %         \hb@xt@\the\@tempdima{%
147
148 %             \hss
149 %             \next{%
150 \LWR@traceinfo{sfsubcap C1}% new
151     \caption@make
152         {\@nameuse{sub\@captype name}}}%
153         {\@nameuse{thesub\@captype}}}%
154         {#3}
155 \LWR@traceinfo{sfsubcap C1}% new
156 % }%
157 %         \hss
158
159 % }
160     }
161     }%
162 \egroup
163 \LWR@startpars% new
164 }

\caption@@@make {\langle caption label\rangle} {\langle caption text\rangle}

165 \renewcommand\caption@@@make[2]{%
166 \LWR@startpars% new

```

```

167 \sbox\@tempboxa{#1}%
168 \ifdim\wd\@tempboxa=\z@
169   \let\caption@lsep\relax
170 \fi
171 \caption@ifempty{#2}{%
172   \let\caption@lsep\@empty
173   \let\caption@tfmt\@firstofone
174 }%
175 %   \@setpar{\@par\caption@@par}\caption@@par
176 \renewcommand{\@par}{\LWR@closeparagraph\LWR@orig@@par}% new
177 \caption@applyfont
178 \caption@fmt
179   {\ifcaption@star\else
180     \begingroup
181       \captionlabelfont
182       #1%
183     \endgroup
184   \fi}%
185 {\ifcaption@star\else
186   \begingroup
187     \caption@iflf\captionlabelfont
188     \relax\caption@lsep
189   \endgroup
190 \fi}%
191 {{\captiontextfont
192   \caption@ifstrut
193     {\vrule\@height\ht\strutbox\@width\z@}%
194     {}}%
195   \nobreak\hskip\z@skip % enable hyphenation
196   \caption@tfmt{#2}
197   \LWR@ensuredoingapar}% new
198   \caption@ifstrut
199     {\ifhmode\@finalstrut\strutbox\fi}%
200     {}}%
201   \par}}
202 \LWR@stoppars}% new
203 }

```

Patches for \sf@sub@label:

```

204 \def\subfloat@label{%
205 \LWR@ensuredoingapar}% new
206 \@ifnextchar(%  %){ match left parenthesis
207   {\sf@sub@label}
208   {\sf@sub@label(Sub\@capttype\space
209     \@ifundefined{thechapter}{\@nameuse{thechapter}\space}%
210     \@nameuse{p@sub\@capttype}%
211     \@nameuse{thesub\@capttype}.)}}

```

Patches for \subref.

The unstarred version uses a `\ref` link whose printed text comes from the `sub@<label>`:

```
212 \renewcommand{\sf@subref}[1]{%
213 \LWR@subnewref{#1}{sub@#1}%
214 }
```

The starred version uses the printed `sub@<label>` which is stored as if it were a page number:

```
215 \renewcommand{\sf@@subref}[1]{\LWR@origpageref{sub@#1}}
```

Defining new subfloats. The `l@sub<type>` for each is redefined.

```
216 \LetLtxMacro\LWR@orig@newsubfloat\@newsubfloat
217
218 \def\@newsubfloat[#1]#2{%
219 \LWR@orig@newsubfloat[#1]{#2}%
220 \renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}
221 }
```

Pre-defined for figures and tables:

```
222 \renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2}}
223 \renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}
224 % \def\subfigure{\subfloat}
225 % \def\subtable{\subfloat}
```

## Package 99

# lwarp-syntonly.sty

## 173 Syntonly

Pkg syntonly Emulated.

for **HTML output**: Discard all options for lwarp-syntonly:

```
1 \LWR@ProvidesPackageDrop{syntonly}

2 \newif\ifsyntax@
3 \syntax@false
4
5 \newcommand*{\syntaxonly}{}
6
7 \@onlypreamble\syntaxonly
```

## Package 100

# lwarp-tabularx.sty

## 174 Tabularx

Pkg `tabularx` `tabularx` is emulated by `lwarp`.

for **HTML output**: Discard all options for `lwarp-tabularx`:

```
1 \LWR@ProvidesPackageDrop{tabularx}

2 \NewDocumentEnvironment{tabularx}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabularx*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}
```

## Package 101

# lwarp-tabulary.sty

## 175 Tabulary

Pkg tabulary tabulary is emulated by lwarp.

for HTML output: Discard all options for lwarp-tabulary.

Column types L, C, R, and J are emulated by lwarp core code.

```
1 \LWR@ProvidesPackageDrop{tabulary}

2 \NewDocumentEnvironment{tabulary}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabulary*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}
9
10 \newdimen\tymin
11 \newdimen\tymax
12 \def\tyformat{}
```

## Package 102

# lwarp-textpos.sty

## 176 Textpos

Pkg textpos textpos is emulated during HTML output, and the textpos package is ignored.

```
for HTML output: 1 \LWR@ProvidesPackageDrop{textpos}

2 \NewDocumentEnvironment{textblock}{m r()}{\}{}
3 \NewDocumentEnvironment{textblock*}{m o r()}{\}{}
4 \newcommand*{\TPGrid}[3][\]{}
5 \NewDocumentCommand{\TPMargin}{s o}{}
6 \newcommand*{\textblockcolour}[1]{}
7 \newcommand*{\textblockrulecolour}[1]{}
8 \newcommand*{\textblockcolor}[1]{}
9 \newcommand*{\textblockrulecolor}[1]{}
10 \newcommand*{\tekstblokkulur}[1]{}
11 \newcommand*{\tekstblokrulekulur}[1]{}
12 \newlength{\TPHorizModule}
13 \newlength{\TPVertModule}
14 \newlength{\TPboxrulesize}
15 \newcommand{\textblocklabel}[1]{}
16 \newcommand*{\showtextsize}{}
17 \newcommand{\textblockorigin}[2]{}

```



## Package 103

# lwarp-theorem.sty

## 177 Theorem

(Based on original code by FRANK MITTELBAACH.)

Pkg theorem theorem is patched for use by lwarp.

### CSS styling of theorems and proofs:

**Theorem:** <div> of class theorembody<theoremstyle>

**Theorem Header:** <span> of class theoremheader

where <theoremstyle> is plain, break, etc.

for HTML output: 1 \LWR@ProvidesPackagePass{theorem}

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
3 \gdef\theoremstyle#1{%
4   \@ifundefined{th#1}{\@warning
5     {Unknown theoremstyle ‘#1’. Using ‘plain’}%
6     \theorem@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% new
8   }%
9   {%
10    \theorem@style{#1}%
11    \renewcommand{\LWR@newtheoremstyle}{#1}% new
12  }%
13  \begingroup
14    \csname th@the\theorem@style \endcsname
15  \endgroup}
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```
16 \gdef\xnthm#1#2[#3]{%
17   \expandafter\@ifdefinable\csname #1\endcsname
```

```

18  {%
19    \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
20    \@definecounter{#1}\@newctr{#1}[#3]%
21    \expandafter\xdef\csname the#1\endcsname
22      {\expandafter \noexpand \csname the#3\endcsname
23        \@thmcountersep \@thmcounter{#1}}%
24    \def\@tempa{\global\@namedef{#1}}%
25    \expandafter \@tempa \expandafter{%
26      \csname th@the \theorem@style
27        \expandafter \endcsname \the \theorem@bodyfont
28        \@thm{#1}{#2}}%
29    \global \expandafter \let \csname end#1\endcsname \@endtheorem
30    \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
31  }}
32
33 \gdef\@ynthm#1#2{%
34   \expandafter\@ifdefinable\csname #1\endcsname
35   {
36     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
37     \@definecounter{#1}%
38     \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
39     \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
40     \expandafter{\csname th@the \theorem@style \expandafter
41       \endcsname \the\theorem@bodyfont \@thm{#1}{#2}}%
42     \global \expandafter \let \csname end#1\endcsname \@endtheorem
43     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
44   }}
45
46 \gdef\@othm#1[#2]#3{%
47   \expandafter\ifx\csname c@#2\endcsname\relax
48     \@nocounterr{#2}%
49   \else
50     \expandafter\@ifdefinable\csname #1\endcsname
51     {
52       \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
53       \expandafter \xdef \csname the#1\endcsname
54         {\expandafter \noexpand \csname the#2\endcsname}%
55       \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
56       \expandafter{\csname th@the \theorem@style \expandafter
57         \endcsname \the\theorem@bodyfont \@thm{#2}{#3}}%
58       \global \expandafter \let \csname end#1\endcsname \@endtheorem
59       \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
60     }%
61   \fi}

```

The following are patched for CSS.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc. They are gathered together here.

Each theorem is encased in a BlockClass environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader`.

```

62 \gdef\th@plain{%\normalfont\itshape
63   \def\@begintheorem##1##2{%
64     \LWR@forcenewpage% new
65     \BlockClass{theorembody\LWR@thisthmstyle}% new
66     \item[\hskip\labelsep
67       \InlineClass{theoremheader}{##1\ ##2}
68     ]}%
69 \def\@opargbegintheorem##1##2##3{%
70   \LWR@forcenewpage% new
71   \BlockClass{theorembody\LWR@thisthmstyle}% new
72   \item[\hskip\labelsep
73     \InlineClass{theoremheader}{##1\ ##2\ (##3)}
74   ]}
75 }
76
77 \gdef\th@break{%\normalfont\slshape
78   \def\@begintheorem##1##2{%
79     \LWR@forcenewpage% new
80     \BlockClass{theorembody\LWR@thisthmstyle}% new
81     \item[\hskip\labelsep
82       \InlineClass{theoremheader}{##1\ ##2\ \newline%
83     ]}%
84 \def\@opargbegintheorem##1##2##3{%
85   \LWR@forcenewpage% new
86   \BlockClass{theorembody\LWR@thisthmstyle}% new
87   \item[\hskip\labelsep
88     \InlineClass{theoremheader}{##1\ ##2\ (##3)}\newline
89   ]}
90 }
91
92 \gdef\th@marginbreak{%\normalfont\slshape
93   \def\@begintheorem##1##2{
94     \LWR@forcenewpage% new
95     \BlockClass{theorembody\LWR@thisthmstyle}% new
96     \item[\hskip\labelsep %
97       \InlineClass{theoremheader}{##2 \quad ##1}\newline
98     ]}%
99 \def\@opargbegintheorem##1##2##3{%
100   \LWR@forcenewpage% new
101   \BlockClass{theorembody\LWR@thisthmstyle}% new
102   \item[\hskip\labelsep %
103     \InlineClass{theoremheader}{##2 \quad ##1\ %
104     (##3)}\newline
105   ]}
106 }
107
```

```

108 \gdef\th@changebreak{%\normalfont\slshape
109 \def\@begintheorem##1##2{
110 \LWR@forcenewpage% new
111 \BlockClass{theorembody\LWR@thisthmstyle}% new
112 \item[\hskip\labelsep
113 \InlineClass{theoremheader}{##2\ ##1}\newline
114 ]}%
115 \def\@opargbegintheorem##1##2##3{%
116 \LWR@forcenewpage% new
117 \BlockClass{theorembody\LWR@thisthmstyle}% new
118 \item[\hskip\labelsep
119 \InlineClass{theoremheader}{ ##2\ ##1\ %
120 (##3)}\newline
121 ]}
122 }
123
124 \gdef\th@change{%\normalfont\slshape
125 \def\@begintheorem##1##2{
126 \LWR@forcenewpage% new
127 \BlockClass{theorembody\LWR@thisthmstyle}% new
128 \item[\hskip\labelsep
129 \InlineClass{theoremheader}{##2\ ##1}
130 ]}%
131 \def\@opargbegintheorem##1##2##3{%
132 \LWR@forcenewpage% new
133 \BlockClass{theorembody\LWR@thisthmstyle}% new
134 \item[\hskip\labelsep
135 \InlineClass{theoremheader}{##2\ ##1\ (##3)}
136 ]}
137 }
138
139 \gdef\th@margin{%\normalfont\slshape
140 \def\@begintheorem##1##2{
141 \LWR@forcenewpage% new
142 \BlockClass{theorembody\LWR@thisthmstyle}% new
143 \item[\hskip\labelsep
144 \InlineClass{theoremheader}{##2 \qquad ##1}
145 ]}%
146 \def\@opargbegintheorem##1##2##3{%
147 \LWR@forcenewpage% new
148 \BlockClass{theorembody\LWR@thisthmstyle}% new
149 \item[\hskip\labelsep
150 \InlineClass{theoremheader}{##2 \qquad ##1\ (##3)}
151 ]}
152 }

```

Patched for CSS:

```

153 \gdef\@endtheorem{\endBlockClass\endtrivlist}

```

## Package 104

# lwarp-threeparttable.sty

## 178 Threeparttable

Pkg **threeparttable** **threeparttable** is emulated during HTML output, and the **threeparttable** package is ignored.

Table note are contained inside a CSS <div> of class **tnotes**. If **enumitem** is used, the note item labels are also individually highlighted with an additional CSS <span> of class **tnoteitemheader**, otherwise they are plain text.

for HTML output: 1 \LWR@ProvidesPackageDrop{threeparttable}

Prints the table note item header inside a CSS class of **tnoteitemheader**.

```
2 \newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}
```

To emulate **threeparttable**:

```
3 \newenvironment*{threeparttable}[1][b]{}{}
```

```
4 \newenvironment*{tablenotes}[1] []
5 {%
6 \LWR@forcenewpage
7 \BlockClass{tnotes}%
8 \ltx@ifpackageloaded{enumitem}{%
9 \setlist[description]{format=\LWR@printtablenote}%
10 }{}%
11 \description%
12 }
13 {%
14 \enddescription%
15 \endBlockClass%
16 }
```

```
17 \newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}
```

## Package 105

# lwarp-tikz.sty

## 179 Tikz

Pkg `tikz` `tikz` is supported.

Accept all options for `lwarp-tikz`:

```
1 \LWR@ProvidesPackagePass{tikz}
```

**catcodes** `lwarp` changes the catcode of `$` for its own use. The `Tikz` `babel` library temporarily changes catcodes back to normal for `Tikz`'s use. `tikz` v3.0.0 introduced the `babel` library which handles catcode changes. For older versions, `lwarp` must change `$`'s catcode itself.

Also see:

<https://tex.stackexchange.com/questions/16199/test-if-a-package-or-package-option-is-loaded>

for HTML output: `2 \begin{warpHTML}`

```
3 \newboolean{LWR@tikzbabel}
4
5 \@ifpackagelater{tikz}{2013/12/20}% Test for Tikz version v3.0.0
6 {\usetikzlibrary{babel}\booltrue{LWR@tikzbabel}}
7 {\boolfalse{LWR@tikzbabel}}
```

Env `tikzpicture` `tikzpicture` environment is enclosed inside a `\lateximage`. May be used as-is, and its contents will be converted to an image.

```
8 \BeforeBeginEnvironment{tikzpicture}{%
9 \lateximage%
10 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
11 {}%
12 {\catcode'\$=3} % dollar sign is math shift
13 }
14
15 \AfterEndEnvironment{tikzpicture}{%
16 \endlateximage%
17 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
18 {}%
19 {\catcode'\$=\active}%
20 }
```

---

21 \end{warpHTML}

## Package 106

# lwarp-titles.sty

## 180 Titles

Pkg titles titles is loaded and used by lwarp during HTML output. All user options and macros are ignored and disabled.

Discard all options for lwarp-titles:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{titles}
```

\pagestyle and \thispagestyle are already disabled in the lwarp code.

```
2 \NewDocumentCommand{\newpagestyle}{m o m}{}
3 \NewDocumentCommand{\renewpagestyle}{m o m}{}

```

```
4 \NewDocumentCommand{\sethead}{o o o m m m}{}
5 \NewDocumentCommand{\setfoot}{o o o m m m}{}

```

```
6 \NewDocumentCommand{\settitledmarks}{s m}{}

```

```
7 \newcommand*{\headrule}{}
8 \newcommand*{\footrule}{}

```

```
9 \newcommand*{\setheadrule}[1]{}
10 \newcommand*{\setfootrule}[1]{}

```

```
11 \newcommand*{\makeheadrule}{}
12 \newcommand*{\makefootrule}{}

```

```
13 \newcommand{\setmarkboth}[1]{}

```

```
14 \NewDocumentCommand{\widenhead}{s o o m m}{}

```

```
15 \newcommand*{\bottitledmarks}{}
16 \newcommand*{\toptitledmarks}{}
17 \newcommand*{\firsttitledmarks}{}
18 \newcommand*{\nexttoptitledmarks}{}
19 \newcommand*{\outertitledmarks}{}
20 \newcommand*{\innertitledmarks}{}

```

```
21 \NewDocumentCommand{\newtitledmark}{s m}{}

```



---

```
22 \NewDocumentCommand{\pretitlemark}{s m}{-}{}

23 \newcommand{\ifsamemark}[4]{}

24 \NewDocumentCommand{\setfloathead}{s o o m m m m}{-}{}
25 \NewDocumentCommand{\setfloatfoot}{s o o m m m m}{-}{}

26 \NewDocumentCommand{\nextfloathead}{s o o m m m m}{-}{}
27 \NewDocumentCommand{\nextfloatfoot}{s o o m m m m}{-}{}

28 \newcommand{\newmarkset}[1]{}

29 \NewDocumentCommand{\newextramarkset}{s m}{-}{}

30 \newcommand{\botextramarks}[1]{}
31 \newcommand{\topextramarks}[1]{}
32 \newcommand{\firstextramarks}[1]{}
33 \newcommand{\nexttoextramarks}[1]{}
34 \newcommand{\outerextramarks}[1]{}
35 \newcommand{\innerextramarks}[1]{}

```

## Package 107

# lwarp-titleref.sty

## 181 Titleref

Pkg titleref titleref is superceded by hyperref and nameref.

for HTML output: 1 \LWR@loadnever{titleref}{hyperref and nameref}

## Package 108

# lwarp-titlesec.sty

## 182 Titlesec

Pkg titlesec titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titlesec:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{titlesec}

2 \newcommand*{\titlelabel}[1]{}

3 \newcommand\titleformat{%
4   \@ifstar{\ttl@format@s}%
5     {\ttl@format@i}}
6 \newcommand{\ttl@format@s}[1]{}
7 \NewDocumentCommand{\ttl@format@i}{m o m m m o}{}

8 \@ifundefined{chapapp}{\let\@chapapp\chaptername}{}
9 \newcommand\chaptertitlename{\@chapapp}

10 \NewDocumentCommand{\titlespacing}{s m m m m o}{}

11 \newcommand*{\filright}{}
12 \newcommand*{\filcenter}{}
13 \newcommand*{\filleft}{}
14 \newcommand*{\fillast}{}
15 \newcommand*{\filinner}{}
16 \newcommand*{\filouter}{}

17 \newcommand\wordsep{\fontdimen\tw@\font \@plus
18   \fontdimen\thr@\font \@minus \fontdimen4\font}

19 \NewDocumentCommand{\titleline}{s o m}{}

20 \providecommand*{\titlerule}{\@ifstar{\ttl@row}{\ttl@rule}}
21 \newcommand*{\ttl@rule}[1] [] {}
22 \newcommand*{\ttl@row}[2] [] {}

23 \newcommand{\iftitlemeasuring}[2] {#2}

24 \newcommand{\assignpagestyle}[2] {#2}

25 \NewDocumentCommand{\titleclass}{m o m o}
```

## Package 109

# lwarp-titletoc.sty

## 183 Titletoc

Pkg titletoc titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titletoc:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{titletoc}

2 \NewDocumentCommand{\dottedcontents}{m o m m m}{\}

3 \newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
4 \NewDocumentCommand{\ttl@tcstar}{m o m m m o o}{\}
5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m m o}{\}

6 \newcommand{\contentsmargin}[2] [] {}

7 \newcommand*{\thecontentslabel}{\thecontentslabel}
8 \newcommand*{\thecontentspage}{\thecontentspage}

9 \newcommand{\contentslabel}[2] [] {\thecontentslabel}
10 \newcommand{\contentspage}[1] [] {\thecontentspage}

11 \newcommand{\contentspush}[1] {}

12 \newcommand{\contentsuse}[2] {}

13 \newcommand*{\startcontents}[1] [] {}
14 \newcommand*{\stopcontents}[1] [] {}
15 \newcommand*{\resumecontents}[1] [] {}

16 \newcommand{\printcontents}[4] [] {}

17 \newcommand{\startlist}[2] [] {}
18 \newcommand{\stoplist}[2] [] {}
19 \newcommand{\resumelist}[2] [] {}

20 \newcommand{\printlist}[4] [] {}

```

## Package 110

# lwarp-titling.sty

## 184 Titling

**Pkg titling** titling is used by lwarp. The following patches are not needed by lwarp, but are required if the user requests titling.

lwarp uses page notes for footnotes, so the various titling footnote restyling commands have no effect.

Pass all options to lwarp-titling:

**for HTML output:** `1 \LWR@ProvidesPackagePass{titling}`

Patch `\@bsmtitleempty`:

```
2 \let\LWR@orig@bsmtitleempty\@bsmtitleempty
3 \renewcommand*{\@bsmtitleempty}{%
4 \LWR@orig@bsmtitleempty%
5 \global\let\published\relax%
6 \global\let\subtitle\relax%
7 }
```

Patch `\keepthetitle`:

```
8 \let\LWR@origkeepthetitle\keepthetitle
9 \renewcommand*{\keepthetitle}{%
10 \LWR@orig@keepthetitle%
11 \global\let\@published\@empty%
12 \global\let\@subtitle\@empty%
13 }
```

Patch `\killtitle`:

```
14 \let\LWR@origkilltitle\killtitle
15 \renewcommand*{\killtitle}{%
16 \LWR@orig@killtitle%
17 \global\let\thepublished\relax%
18 \global\let\thesubtitle\relax%
19 }
```

## Package 111

# lwarp-tocloft.sty

## 185 Tocloft

Pkg `tocloft` `tocloft` is emulated. Most user options and macros are ignored and disabled. `\newlistof` and `\cftchapterprecis` are supported.

Discard all options for `lwarp-tocloft`:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{tocloft}

2 \newcommand{\tocloftpagestyle}[1]{}

3 \newcommand*{\cftmarktoc}{}
4 \newcommand*{\cfttoctitlefont}{}
5 \newcommand*{\cftaftertoctitle}{}

6 \newlength{\cftbeforetoctitleskip}
7 \newlength{\cftaftertoctitleskip}

8 \newcommand*{\cftmarklof}{}
9 \newcommand*{\cftloftitlefont}{}
10 \newcommand*{\cftafterloftitle}{}

11 \newlength{\cftbeforeloftitleskip}
12 \newlength{\cftafterloftitleskip}

13 \newcommand*{\cftmarklot}{}
14 \newcommand*{\cftlottitlefont}{}
15 \newcommand*{\cftafterlottitle}{}

16 \newlength{\cftbeforelottitleskip}
17 \newlength{\cftafterlottitleskip}

18 \newcommand*{\cftdot}{.}
19 \providecommand*{\cftdotsep}{1}
20 \newcommand*{\cftnodots}{5000}
21
22 \providecommand{\cftdotfill}[1]{}

23 \newcommand*{\cftsetpnumwidth}[1]{}
24 \newcommand*{\cftsetrmarg}[1]{}

```

```
25 \newcommand*{\cftpnumalign}[1]{}

26 \newlength{\cftparskip}

27 \newlength{\cftbeforepartskip}
28 \newlength{\cftpartincent}
29 \newlength{\cftpnumwidth}
30 \newcommand*{\cftpfont}{}
31 \newcommand*{\cftpapresnum}{}
32 \newcommand*{\cftpataftersnum}{}
33 \newcommand*{\cftpataftersnumb}{}
34 \newcommand*{\cftpleader}{}
35 \newcommand*{\cftpdotsep}{1}
36 \newcommand*{\cftpfont}{}
37 \newcommand*{\cftpartafterpnum}{}

38 \newlength{\cftbeforechapskip}
39 \newlength{\cftchapindent}
40 \newlength{\cftchapnumwidth}
41 \newcommand*{\cftchapfont}{}
42 \newcommand*{\cftchappresnum}{}
43 \newcommand*{\cftchapaftersnum}{}
44 \newcommand*{\cftchapaftersnumb}{}
45 \newcommand*{\cftchapleader}{}
46 \newcommand*{\cftchapdotsep}{1}
47 \newcommand*{\cftchapfont}{}
48 \newcommand*{\cftchapafterpnum}{}

49 \newlength{\cftbeforesecskip}
50 \newlength{\cftsecindent}
51 \newlength{\cftsecnumwidth}
52 \newcommand*{\cftsecfont}{}
53 \newcommand*{\cftsecpresnum}{}
54 \newcommand*{\cftsecaftersnum}{}
55 \newcommand*{\cftsecaftersnumb}{}
56 \newcommand*{\cftsecleader}{}
57 \newcommand*{\cftsecdotsep}{1}
58 \newcommand*{\cftsecfont}{}
59 \newcommand*{\cftsecafterpnum}{}

60 \newlength{\cftbeforesubsecskip}
61 \newlength{\cftsubsecindent}
62 \newlength{\cftsubsecnumwidth}
63 \newcommand*{\cftsubsecfont}{}
64 \newcommand*{\cftsubsecpresnum}{}
65 \newcommand*{\cftsubsecaftersnum}{}
66 \newcommand*{\cftsubsecaftersnumb}{}
67 \newcommand*{\cftsubsecleader}{}
68 \newcommand*{\cftsubsecdotsep}{1}
```

---

```

69 \newcommand*{\cftsubsecpagefont}{}
70 \newcommand*{\cftsubsecafterpnum}{}

71 \newlength{\cftbeforesubsubsecskip}
72 \newlength{\cftsubsubsecindent}
73 \newlength{\cftsubsubsecnumwidth}
74 \newcommand*{\cftsubsubsecfont}{}
75 \newcommand*{\cftsubsubsecpresnum}{}
76 \newcommand*{\cftsubsubsecaftersnum}{}
77 \newcommand*{\cftsubsubsecaftersnumb}{}
78 \newcommand*{\cftsubsubsecleader}{}
79 \newcommand*{\cftsubsubsecdotsep}{1}
80 \newcommand*{\cftsubsubsecpagefont}{}
81 \newcommand*{\cftsubsubsecafterpnum}{}

82 \newlength{\cftbeforeparaskip}
83 \newlength{\cftparaindent}
84 \newlength{\cftparanumwidth}
85 \newcommand*{\cftparafont}{}
86 \newcommand*{\cftparapresnum}{}
87 \newcommand*{\cftparaftersnum}{}
88 \newcommand*{\cftparaftersnumb}{}
89 \newcommand*{\cftparaleader}{}
90 \newcommand*{\cftparadotsep}{1}
91 \newcommand*{\cftparapagefont}{}
92 \newcommand*{\cftparaafterpnum}{}

93 \newlength{\cftbeforesubparaskip}
94 \newlength{\cftsubparaindent}
95 \newlength{\cftsubparanumwidth}
96 \newcommand*{\cftsubparafont}{}
97 \newcommand*{\cftsubparapresnum}{}
98 \newcommand*{\cftsubparaftersnum}{}
99 \newcommand*{\cftsubparaftersnumb}{}
100 \newcommand*{\cftsubparaleader}{}
101 \newcommand*{\cftsubparadotsep}{1}
102 \newcommand*{\cftsubparapagefont}{}
103 \newcommand*{\cftsubparaafterpnum}{}

104 \newlength{\cftbeforefigskip}
105 \newlength{\cftfigindent}
106 \newlength{\cftfignumwidth}
107 \newcommand*{\cftfigfont}{}
108 \newcommand*{\cftfigpresnum}{}
109 \newcommand*{\cftfigaftersnum}{}
110 \newcommand*{\cftfigaftersnumb}{}
111 \newcommand*{\cftfigleader}{}
112 \newcommand*{\cftfigdotsep}{1}
113 \newcommand*{\cftfigpagefont}{}

```



```

114 \newcommand*\cftfigafterpnum){}

115 \newlength{\cftbeforesubfigskip}
116 \newlength{\cftsubfigindent}
117 \newlength{\cftsubfignumwidth}
118 \newcommand*\cftsubfigfont){}
119 \newcommand*\cftsubfigpresnum){}
120 \newcommand*\cftsubfigaftersnum){}
121 \newcommand*\cftsubfigaftersnumb){}
122 \newcommand*\cftsubfigleader){}
123 \newcommand*\cftsubfigdotsep}{1}
124 \newcommand*\cftsubfigpagefont){}
125 \newcommand*\cftsubfigafterpnum){}

126 \newlength{\cftbeforetabskip}
127 \newlength{\cfttabindent}
128 \newlength{\cfttabnumwidth}
129 \newcommand*\cfttabfont){}
130 \newcommand*\cfttabpresnum){}
131 \newcommand*\cfttabaftersnum){}
132 \newcommand*\cfttabaftersnumb){}
133 \newcommand*\cfttableader){}
134 \newcommand*\cfttabdotsep}{1}
135 \newcommand*\cfttabpagefont){}
136 \newcommand*\cfttabafterpnum){}

137 \newlength{\cftbeforesubtabskip}
138 \newlength{\cftsubtabindent}
139 \newlength{\cftsubtabnumwidth}
140 \newcommand*\cftsubtabfont){}
141 \newcommand*\cftsubtabpresnum){}
142 \newcommand*\cftsubtabaftersnum){}
143 \newcommand*\cftsubtabaftersnumb){}
144 \newcommand*\cftsubtableader){}
145 \newcommand*\cftsubtabdotsep}{1}
146 \newcommand*\cftsubtabpagefont){}
147 \newcommand*\cftsubtabafterpnum){}

148 \newcommand{\cftsetindents}[3]{}

149 \newcommand{\pagenumbersoff}[1]{}
150 \newcommand{\pagenumberson}[1]{}

```

Emulated through the \newfloat mechanism.

```

151 \NewDocumentCommand{\newlistof}{o m m}
152 {%
153 \IfValueTF{#1}

```

```

154 {\newfloat{#2}{tbp}{#3}[#1]}
155 {\newfloat{#2}{tbp}{#3}}
156 \@namedef{listof#2}{\listof{#2}{#4}}
157 \@namedef{#2depth}{1}
158 \expandafter\newlength\csuse{cftbefore#2skip}
159 \expandafter\newlength\csuse{cft#2indent}
160 \expandafter\newlength\csuse{cft#2numwidth}
161 \@namedef{cft#2font}{}
162 \@namedef{cft#2presnum}{}
163 \@namedef{cft#2aftersnum}{}
164 \@namedef{cft#2aftersnumb}{}
165 \@namedef{cft#2leader}{}
166 \@namedef{cft#2dotsep}{1}
167 \@namedef{cft#2pagefont}{}
168 \@namedef{cft#2afterpnum}{}
169 }

```

\cftchapterprecis from tocloft:

```

170 \newcommand{\cftchapterprecis}[1]{%
171   \cftchapterprecishere{#1}
172   \cftchapterprecistoc{#1}}
173 \newcommand{\cftchapterprecishere}[1]{%
174   \begin{quote}\textit{#1}\end{quote}}
175 \newcommand{\cftchapterprecistoc}[1]{
176   \addtocontents{toc}{%
177     {
178       \protect\begin{quote}#1\protect\end{quote}}
179   }
180 }

```


## Package 112

# lwarp-transparent.sty

## 186 Transparent

*(Based on original code by HEIKO OBERDIEK.)*

Pkg transparent Emulated. `\texttransparent` works for inline objects. `\transparent` only works for `\includegraphics`.

 Not X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X! Note that transparent does not work with X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X.

for HTML output: Discard all options for lwarp-transparent:

```
1 \LWR@ProvidesPackageDrop{transparent}

2 \newcommand*{\transparent}[1]{\edef\LWR@opacity{#1}}
3
4 \newcommand*{\texttransparent}[2]{%
5 \begingroup%
6 \transparent{#1}%
7 \InlineClass[opacity: #1]{transparent}{#2}%
8 \endgroup%
9 }
```

## Package 113

# lwarp-trivfloat.sty

## 187 Trivfloat

**Pkg** `trivfloat` `trivfloat` is forced to use the built-in `lwarp` emulation for floats.

Discard all options for `lwarp-trivfloat`. This tells `trivfloat` not to use `floatrow` or `memoir`.

**for HTML output:** `1 \LWR@ProvidesPackageDrop{trivfloat}`  
`2 \LWR@origRequirePackage{trivfloat}`

**for HTML & PRINT:** `3 \begin{warpall}`

To create a new float type and change its name:

---

```
\trivfloat{example}
\renewcommand{\exemplename}{Example Name}
\crefname{example}{example}{examples}
\Crefname{example}{Example}{Examples}
```

---

`4 \end{warpall}`

**\tfl@chapter@fix** Nullified at the beginning of the document. Is used by `trivfloat` to correct float chapter numbers, but is not needed for `lwarp`.

**for HTML output:** `5 \begin{warpHTML}`  
`6 \AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}`  
`7 \end{warpHTML}`

### 187.1 Combining `\newfloat`, `\trivfloat`, and `algorithmicx`

**for HTML & PRINT:** `8 \begin{warpall}`

For both print and HTML output:



When using `float`, `trivfloat`, or `algorithmicx` at the same time, be aware of conflicting file usage. `algorithmicx` uses `.loa`. `trivfloat` by default starts with `.loa` and goes up

for additional floats, skipping .lof and .lot.



When using `\newfloat`, be sure to manually assign higher letters to the `\newfloat` files to avoid .loa used by `algorithmicx`, and any files used by `trivfloat`. Also avoid using .lof and .lot.



When using `\trivfloat`, you may force it to avoid conflicting with `algorithmicx` by starting `trivfloat`'s file extensions with .lob:

---

```
\makeatletter
\setcounter{tfl@float@cnt}{1}} % start trivfloats with .lob
\makeatletter
```

---

```
9 \end{warpall}
```

## Package 114

# lwarp-ulem.sty

## 188 Ulem

*(Based on original code by DONALD ARSENEAU.)*

Pkg ulem Emulated.

for HTML output: Original lwarp definitions:

```
1 \LetLtxMacro\LWR@ulemorigemph\emph
2 \LetLtxMacro\LWR@ulemorigtextbf\textbf
```

Basic markup commands, using CSS:

```
3 \NewDocumentCommand{\uline}{+m}{%
4 \InlineClass{uline}{#1}%
5 }
6
7 \NewDocumentCommand{\uuline}{+m}{%
8 \InlineClass{uuline}{#1}%
9 }
10
11 \NewDocumentCommand{\uwave}{+m}{%
12 \InlineClass{uwave}{#1}%
13 }
14
15 \NewDocumentCommand{\sout}{+m}{%
16 \InlineClass{sout}{#1}%
17 }
18
19 \NewDocumentCommand{\xout}{+m}{%
20 \InlineClass{xout}{#1}%
21 }
22
23 \NewDocumentCommand{\dashuline}{+m}{%
24 \InlineClass{dashuline}{#1}%
25 }
26
27 \NewDocumentCommand{\dotuline}{+m}{%
28 \InlineClass{dotuline}{#1}%
29 }
```

Nullified parameters:

```
30 \NewDocumentCommand{\ULthickness}{-}{-}
31 \newlength{\ULdepth}
```

Nullified/emulated macros:

```
32 \NewDocumentCommand{\markoverwith}{m}{-}
33 \NewDocumentCommand{\ULon}{+m}{\uline{#1}\egroup}
```

`\useunder` only works with `\textbf`, etc, but not `\bfseries`, etc.

```
34 \NewDocumentCommand{\useunder}{m m m}{%
35 \relax%
36 \ifx\relax#3\relax\else % argumentative command
37   \def#3{#1}\MakeRobust{#3}\fi
38 }
```

Triggered by package options, also available for the users:

```
39 \newcommand*{\normalem}{\LetLtxMacro\emph\LWR@ulemorigemph}
40 \newcommand*{\ULforem}{\LetLtxMacro\emph\uline}
41 \ULforem% default
```

Package options:

```
42 \DeclareOption{normalem}{\normalem}
43 \DeclareOption{ULforem}{\ULforem}
44 \DeclareOption{normalbf}{-}
45 \DeclareOption{UWforbf}{\useunder{\uwave}{\bf}\textbf}}
```

Emulate the original package:

```
46 \LWR@ProvidesPackageDrop{ulem}
```

## Package 115

# lwarp-upref.sty

## 189 Upref

Pkg upref Ignored.

for **HTML output:** Discard all options for lwarp-upref:

```
1 \LWR@ProvidesPackageDrop{upref}
```



## Package 116

# lwarp-verse.sty

## 190 Verse

(Based on original code by PETER WILSON.)

**Pkg verse** verse is supported and patched by lwarp.

**for HTML output:** Pass all options for lwarp-verse:

```
1 \LWR@ProvidesPackagePass{verse}
```

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

---

```
\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}
```

---

**Len \leftskip** These lengths are used by `verse` and `memoir` to control the left margin, and they may already be set by the user for print output. New lengths `\HTMLvleftskip` and `\HTMLleftmargini` are provided to control the margins in HTML output. These new lengths may be set by the user before any `verse` environment, and persist until they are manually changed again. One reason to change `\HTMLleftmargini` is if there is a wide `\flagverse` in use, such as the word “Chorus”, in which case the value of `\HTMLleftmargini` should be set to a wide enough length to contain “Chorus”. The default is wide enough for a stanza number.

**Len \leftmargini**

**Len \TMLvleftskip**

**Len \TMLleftmargini**

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

**Env verse** The `verse` environment will be placed inside a HTML `pre`.

```
2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching verse.}
```

At the beginning of the `verse` environment:

```

4 \AtBeginEnvironment{verse}
5 {%

Pkg  verse  The verse or memoir packages can place stanza numbers to the left with their
Pkg  memoir \flagverse command. Do not allow them to go into the left margin, which would
\flagverse cause pdfcrop to crop the entire page further to the left:

Len  \leftskip
6 \ifdef{\vleftskip}{%
7 \setlength{\vleftskip}{\HTMLvleftskip}
8 \setlength{\leftmargini}{\HTMLleftmargini}
9 }{}
10 \LWR@forcenewpage
11 \LWR@atbeginverbatim{verse}
12 \unskip\LWR@origvspace{-\baselineskip}
13 }
```

After the end of the `verse` environment, which places the `pre` tag at the regular left margin:

```

14 \AfterEndEnvironment{verse}{
15 \unskip\LWR@origvspace{-\baselineskip}
16 \LWR@afterendverbatim
17 }
```

Patch to place poemtitle inside an HTML span of class `poemtitle`:

```

18 \ifdef{\poemtitle}{
19 \DeclareDocumentCommand{\@vstypeptitle}{m}{%
20   \vspace{\beforepoemtitleskip}%
21   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
22   \vspace{\afterpoemtitleskip}%
23 }
24 }{}
25
26 }% AfterEndPreamble
```

## Package 117

# lwarp-wallpaper.sty

## 191 Wallpaper

Pkg wallpaper wallpaper is emulated during HTML output, and the wallpaper package is ignored.

```
for HTML output: 1 \LWR@ProvidesPackageDrop{wallpaper}

2 \newcommand*{\CenterWallPaper}[2]{}
3 \newcommand*{\ThisCenterWallPaper}[2]{}
4 \newcommand*{\TileWallPaper}[3]{}
5 \newcommand*{\ThisTileWallPaper}[3]{}
6 \newcommand*{\TileSquareWallPaper}[2]{}
7 \newcommand*{\ThisTileSquareWallPaper}[2]{}
8 \newcommand*{\ULCornerWallPaper}[2]{}
9 \newcommand*{\ThisULCornerWallPaper}[2]{}
10 \newcommand*{\LLCornerWallPaper}[2]{}
11 \newcommand*{\ThisLLCornerWallPaper}[2]{}
12 \newcommand*{\URCornerWallPaper}[2]{}
13 \newcommand*{\ThisURCornerWallPaper}[2]{}
14 \newcommand*{\LRCornerWallPaper}[2]{}
15 \newcommand*{\ThisLRCornerWallPaper}[2]{}
16 \newcommand*{\ClearWallPaper}{}
17 \newlength{\wpXoffset}
18 \newlength{\wpYoffset}
```

## Package 118

# lwarp-wrapfig.sty

## 192 Wrapfig

Pkg wrapfig wrapfig is emulated during HTML output, and the wrapfig package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{wrapfig}

Computed width of a wrapped object. Used to print the HTML style.

```

2 \newlength{\LWR@wrapwidth}

3 \newcommand*{\LWR@wrapposition}{%
4
5 \newcommand*{\LWR@subwrapfigure}[2]{%
6 \LWR@maybeinthisfloat%
7 \renewcommand*{\LWR@wrapposition}{}%
8 \ifthenelse{%
9   \equal{#1}{r}\OR\equal{#1}{R}\OR%
10  \equal{#1}{o}\OR\equal{#1}{O}%
11 }{%
12 {\renewcommand*{\LWR@wrapposition}{float:right}}%
13 {\renewcommand*{\LWR@wrapposition}{float:left}}%
14 \setlength{\LWR@wrapwidth}{#2}%
15 \addtolength{\LWR@wrapwidth}{4em}%
16 \uselengthunit{PT}%
17 \LWR@forcenewpage
18 \LWR@stoppars%
19 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{\LWR@thisfloat}"
20 style="width:\rndprintlength{\LWR@wrapwidth} ; %
21 \LWR@wrapposition"%
22 }%
23 \LWR@startpars
24 }
25
26
27 \NewDocumentEnvironment{wrapfigure}{o m o m}
28 {%
29 \LWR@subwrapfigure{#2}{#4}%
30 \captionsetup{type=figure}%
31 }
32 {%
33 \LWR@htmldivclassend{div}
34 }
```

```
35
36
37 \NewDocumentEnvironment{wraptable}{o m o m}
38 {%
39 \LWR@subwrapfigure{#2}{#4}%
40 \captionsetup{type=table}%
41 }
42 {%
43 \LWR@htmldivclassend{div}
44 }
45
46
47 \NewDocumentEnvironment{wrapfloat}{m o m o m}
48 {%
49 \LWR@subwrapfigure{#3}{#5}%
50 \captionsetup{type=#1}%
51 }
52 {%
53 \LWR@htmldivclassend{div}
54 }
55
56 \newlength{\wrapoverhang}
```

## Package 119

# lwarp-xcolor.sty

## 193 Xcolor

Pkg	<code>xcolor</code>	<code>xcolor</code> is supported by <code>lwarp</code> .
<code>\colorboxBlock</code> and <code>\fcolorboxBlock</code>		<code>\colorboxBlock</code> and <code>\fcolorboxBlock</code> are provided for increased HTML compatibility, and they are identical to <code>\colorbox</code> and <code>\fcolorbox</code> in print mode. In HTML mode they place their contents into a <code>&lt;div&gt;</code> instead of a <code>&lt;span&gt;</code> . These <code>&lt;div&gt;</code> s are set to <code>display: inline-block</code> so adjacent <code>\colorboxBlocks</code> appear side-by-side in HTML, although text is placed before or after each.
		Print-mode definitions for <code>\colorboxBlock</code> and <code>\fcolorboxBlock</code> are created by <code>lwarp</code> 's core if <code>xcolor</code> is loaded.
background:	<code>none</code>	<code>\fcolorbox</code> and <code>\fcolorboxBlock</code> allow a background color of <code>none</code> , in which case only the frame is drawn, which can be useful for HTML.
color support		Color definitions, models, and mixing are fully supported without any changes required.
tables		Colored tables are ignored so far. Use CSS to style tables.
colored text and boxes		<code>\textcolor</code> , <code>\colorbox</code> , and <code>\fcolorbox</code> are supported.
<code>\color</code> and <code>\pagecolor</code>		<code>\color</code> and <code>\pagecolor</code> are ignored. Use CSS or <code>\textcolor</code> where possible.

### 193.1 Xcolor definitions: location and timing

The `lwarp` core and its `lwarp-xcolor` package are tightly integrated to allow comparable results for print, HTML and print inside an HTML `lateximage`. This requires a number of definitions and redefinitions depending on whether each of `xcolor` and `lateximage` is being used, and whether print or HTML is being generated. Some of these actions are one-time when `xcolor` is loaded, and others are temporary as `lateximage` is used.

**When `lwarp` is loaded in print or HTML mode:** `\LWR@restoreorigxcolor` and `\LWR@restoremoreorigxcolor` are defined as null. Section 66.1

**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@restoreorigxcolor` is defined 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 66.2. 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@restoremorigxcolor` is defined to temporarily set these functions to their print-mode versions.

**In a lateximage in HTML mode:** `\LWR@restoreorigxcolor` and `\LWR@restoremorigxcolor` are used to temporarily restore the print-mode definitions of xcolor's functions. See `\LWR@restoreorigformatting` on page 307.

**\color:**

**Print:** Used as-is.

**HTML:** Ignored by `pdftotext`, and will not appear.

**HTML lateximage:** Colors will appear in a `lateximage`.

**\textcolor:**

**Print:** Used as-is.

**HTML:** Redefined by lwarp-xcolor, page 538.

**HTML lateximage:** Remembers and reuses the print version.

**\pagecolor:**

**Print:** Used as-is.

**HTML:** Ignored.

**HTML lateximage:** Colors will be picked up in a `lateximage`.

**\nopagecolor:**

**Print:** Used as-is.

**HTML:** Ignored.

**HTML lateximage:** Colors will be picked up in a `lateximage`.

**\colorbox:**

**Print:** Used as-is.

**HTML:** Redefined by lwarp-xcolor, page 538.

**HTML lateximage:** Remembers and reuses the print version.

**\colorboxBlock:**

**Print:** Becomes \colorbox.

**HTML:** Newly defined by lwarp-xcolor to use a <div>, page 539.

**HTML lateximage:** Remembers and reuses the print version \colorbox.

**\fcolorbox:**

**Print:** Modified to allow a background of none.

\LWRprint@fcolorbox at section 66.2

**HTML:** Redefined by lwarp-xcolor, page 539.

**HTML lateximage:** Remembers and reuses the print version.

**\fcolorboxBlock:**

**Print:** Becomes \fcolorbox. Section 66.2

**HTML:** Newly defined by lwarp-xcolor to use a <div>, page 540.

**HTML lateximage:** Remembers and reuses the print version \fcolorbox.

**fcolorminipage:**

**Print:** Newly defined in the lwarp core.

\LWRprint@fcolorminipage at section 66.2

**HTML:** Newly defined by lwarp-xcolor, page 540.

**HTML lateximage:** Uses the print version.

**\boxframe:**

**Print:** Used as-is.

**HTML:** Redefined by lwarp-xcolor, page 541.

**HTML lateximage:** Remembers and reuses the print version.

## 193.2 Code

for HTML output: 1 \LWR@ProvidesPackagePass{xcolor}

2 \begin{warpHTML}



Remember the following print-mode actions to be restored when inside a `lateximage` environment:

```

3 \LetLtxMacro\LWRprint@textcolor\textcolor
4 \LetLtxMacro\LWRprint@pagecolor\pagecolor
5 \LetLtxMacro\LWRprint@nopagecolor\nopagecolor
6 \LetLtxMacro\LWRprint@colorbox\colorbox
7 \LetLtxMacro\LWRprint@colorboxBlock\colorbox
8 \LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox
9 \LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox
10 \LetLtxMacro\LWRorigprint@boxframe\boxframe

```

`\LWR@restoreorigxcolor` Inside a `lateximage` the following gets restored to their print-mode actions:

```

11 \renewcommand*{\LWR@restoreorigxcolor}{%
12 \LWR@traceinfo{\LWR@restoreorigxcolorformatting}%
13 \LetLtxMacro\textcolor\LWRprint@textcolor%
14 \LetLtxMacro\pagecolor\LWRprint@pagecolor%
15 \LetLtxMacro\nopagecolor\LWRprint@nopagecolor%
16 \LetLtxMacro\colorbox\LWRprint@colorbox%
17 \LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
18 \LetLtxMacro\boxframe\LWRorigprint@boxframe%
19 }

```

`\LWR@tempcolor` The color converted to HTML colorspace.

```

20 \newcommand*{\LWR@tempcolor}{}
21 \newcommand*{\LWR@tempcolortwo}{}

```

`\LWR@colorstyle`  $\{\langle 1: styletext \rangle\} \{\langle 2: model \rangle\} \{\langle 3: color \rangle\}$

For a color style, prints the color converted to HTML colors.

```

22 \NewDocumentCommand{\LWR@colorstyle}{m m}{%
23 \begingroup%
24 \LWR@FBcancel%

```

Use the `xcolor` package to convert to an HTML color space:

```

25 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%

```

Print the converted color:

```

26 \#\LWR@tempcolor%
27 \endgroup%
28 }

```

`\LWR@borderpadding`  $\{\langle colorstyle \rangle\} \{\langle color \rangle\}$  Prints the HTML attributes for a black border and padding.

`\LWR@forceminwidth` must be used first in order to set the border width.

```
29 \newcommand*\LWR@borderpadding}[2]{%
30 \uselengthunit{PT}%
31 border:\rndprintlength{\LWR@atleastonept} solid \LWR@colorstyle{#1}{#2} ; %
32 padding:\rndprintlength{\fboxsep}%
33 }
```

`\color` `\color` appears in the L<sup>A</sup>T<sub>E</sub>X PDF output, but is ignored by `pdftotext` and thus is ignored in the HTML file. Text styling by local group is not yet supported.

Each of the following macros is given a temporary name, and is `\let` to the final name once the HTML conversion starts.

`\textcolor` [*model*] {*color*} {*text*}

Converted into an HTML hex color span.

```
34 \RenewDocumentCommand{\textcolor}{0{named} m m}{%
35 \begingroup%
36 \LWR@FBcancel%
37 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
38 \InlineClass[color:\LWR@colorstyle{#1}{#2}]{textcolor}{%
39 \renewcommand*\LWR@currenttextcolor}{\#\LWR@tempcolor}%
40 #3%
41 }%
42 \endgroup%
43 }
```

`\pagecolor` [*model*] {*color*}

Ignored. Use `\CSSFilename` instead.

```
44 \renewcommand*\pagecolor}[2][named]{}
```

`\nopagecolor` Ignored.

```
45 \let\nopagecolor\relax
```

`\colorbox` [*model*] {*color*} {*text*}

Converted into an HTML hex background color `<span>`.

```
46 \RenewDocumentCommand{\colorbox}{0{named} m +m}{%
47 \begingroup%
48 \LWR@FBcancel%
```

```

49 \uselengthunit{PT}%
50 \InlineClass[%
51 background:\LWR@colorstyle{#1}{#2} ; %
52 padding:\rndprintlength{\fboxsep}%
53 ]{colorbox}{#3}%
54 \endgroup%
55 }

```

`\colorboxBlock` [*model*] {*color*} {*text*}

Converted into an HTML hex background color <div>.

```

56 \NewDocumentCommand{\colorboxBlock}{0{named} m +m}{%
57 \begingroup%
58 \LWR@FBcancel%
59 \uselengthunit{PT}%
60 \begin{BlockClass}%
61 background:\LWR@colorstyle{#1}{#2} ; %
62 padding:\rndprintlength{\fboxsep}%
63 ]{colorboxBlock}
64 #3
65 \end{BlockClass}%
66 \endgroup%
67 }

```

`\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.

```

68 \RenewDocumentCommand{\fcolorbox}{0{named} m 0{named} m +m}{%
69 \LWR@traceinfo{HTML fcolorbox #2 #4}%
70 \begingroup%
71 \LWR@FBcancel%
72 \uselengthunit{PT}%
73 \LWR@forceminwidth{\fboxrule}%
74 \ifthenelse{\equal{#4}{none}}{%
75 {% no background color
76   \InlineClass[%
77   \LWR@borderpadding{#1}{#2}%
78   ]{fcolorbox}{#5}%
79 }%
80 {% yes background color
81   \InlineClass[%
82   \LWR@borderpadding{#1}{#2} ; %
83   background:\LWR@colorstyle{#3}{#4}%

```

```

84    ]{\fcolorbox}{#5}%
85 }%
86 \endgroup%
87 }

```

`\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.

```

88 \NewDocumentCommand{\fcolorboxBlock}{0{named} m 0{named} m +m}{%
89 \LWR@traceinfo{HTML fcolorboxBlock #2 #4}%
90 \begingroup%
91 \LWR@FBcancel%
92 \uselengthunit{PT}%
93 \LWR@forceminwidth{\fboxrule}%
94 \ifthenelse{\equal{#4}{none}}{%
95 {% no background color
96   \begin{BlockClass}[%
97     \LWR@borderpadding{#1}{#2}%
98   ]{\fcolorboxBlock}
99   #5
100   \end{BlockClass}%
101 }%
102 {% yes background color
103   \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
104   \begin{BlockClass}[%
105     background:\#\LWR@tempcolortwo; %
106     \LWR@borderpadding{#1}{#2}%
107   ]{\fcolorboxBlock}
108   #5
109   \end{BlockClass}%
110 }%
111 \endgroup%
112 \LWR@traceinfo{HTML fcolorboxBlock done}%
113 }

```

Env `fcolorminipage` [*1:framemodel*] {*2:framecolor*} [*3:boxmodel*] {*4:boxcolor*} [*5:align*]  
[*6:height*] [*7:inner-align*] {*8:width*}

Creates a framed HTML <div> around its contents.

A print-output version is defined in the lwarp core: section [66.2](#)

`\LWR@subfcolorminipage` {*framemodel*} {*framecolor*} {*background tag*} {*height*}

```

114 \NewDocumentCommand{\LWR@subfcolorminipage}{m m m m}{%
115 \begin{BlockClass}{%
116 #3%
117 \LWR@borderpadding{#1}{#2} ; %
118 \IfValueTF{#4}{height:\rndprintlength{\LWR@tempheight} ; }{}%
119 width:\rndprintlength{\LWR@tempwidth}%
120 }{fcolorminipage}%
121 }

```

**fcolorminipage** [*1:framestyle*] [*2:framecolor*] [*3:bkgcolorstyle*] [*4:bkgcolor*] [*5:align*]  
 [*6:height*] [*7:inneralign*] [*8:width*]

```

122 \NewDocumentEnvironment{fcolorminipage}{O{named} m O{named} m O{c} o o m}{
123 {%
124 \LWR@FBcancel%
125 \setlength{\LWR@tempwidth}{#8}%
126 \IfValueTF{#6}{\setlength{\LWR@tempheight}{#6}}{}%
127 \uselengthunit{PT}%
128 \LWR@forceminwidth{\fboxrule}%
129 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
130 \ifthenelse{\equal{#4}{none}}{}%
131 {\LWR@subfcolorminipage{#1}{#2}{#6}}%
132 {%
133 \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
134 \LWR@subfcolorminipage{#1}{#2}{background:\#\LWR@tempcolortwo\ ; }{#6}%
135 }%
136 }
137 {\end{BlockClass}}

```

**\boxframe** [*width*] [*height*] [*depth*]

The depth is added to the height, but the box is not decended below by the depth.  
**\textcolor** is honored.

```

138 \renewcommand*\boxframe}[3]{%
139 {%
140 \setlength{\LWR@tempwidth}{#1}%
141 \setlength{\LWR@tempheight}{#2}%
142 \addtolength{\LWR@tempheight}{#3}%
143 \uselengthunit{PT}%
144 \LWR@forceminwidth{\fboxrule}%
145 \InlineClass[%
146 display:inline-block ; %
147 border:\rndprintlength{\LWR@atleastonept} solid \LWR@currenttextcolor{ } ; %
148 width:\rndprintlength{\LWR@tempwidth} ; %
149 height:\rndprintlength{\LWR@tempheight}%
150 ]{boxframe}{}%
151 }%

```

---

152 }

153 \end{warpHTML}

## Package 120

# lwarp-xfrac.sty

## 194 Xfrac

Pkg **xfrac** Supported by adding xfrac instances.

for HTML output: 1 \LWR@ProvidesPackagePass{xfrac}

⚠ font size

In the user's document preamble, **lwarp** should be loaded after font-related setup. During HTML conversion, this font is used by **lwarp** to generate its initial PDF output containing HTML tags, later to be converted by **pdftotext** to a plain text file. While the text may be in any font which **pdftotext** can read, the math is directly converted into SVG images using this same user-selected font. **xfrac** below is set for the Latin Modern (lmr) font. If another font is used, it may be desirable to redefine `\xfracHTMLfontsize` with a different em size.

`\sfrac` [*instance*] {*num*} [*sep*] {*denom*}

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. `\scalebox` is made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a `lateximage`, no adjustments are necessary.

for HTML & PRINT: 2 \begin{warpall}

`\xfracHTMLfontsize` User-redefinable macro which controls the font size of the fraction.

3 \newcommand\*{\xfracHTMLfontsize}{.6em}

4 \end{warpall}

for HTML output: 5 \begin{warpHTML}

font size A span for a small font, used in the numerator and denominator:

```
6 \newcommand*{\LWR@htmlsmallfontstart}{%
7 \LWR@htmltagc{span style="font-size:\xfracHTMLfontsize"}{}}%
8 \LWR@nestspan%
9 %
10 }
```

```

11
12 \newcommand*{\LWR@htmlsmallfontend}{%
13 \LWR@htmltagc{/span}%
14 \endLWR@nestspan%
15 }

```

`\scalebox` A nullified `\scalebox` command, to avoid introducing HTML scaling tags:

```

16 \NewDocumentCommand{\LWR@noscalebox}{m o m}{#3}

```

instances Instances of `xfrac` for various font choices:

Produce HTML tags for a small superscript numerator and a small (non-subscript) denominator.

Scaling is turned off so that `pdftotext` correctly reads the result.

```

17 \DeclareInstance{xfrac}{default}{text}{
18 numerator-format = {%
19 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
20 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
21 denominator-format = {%
22 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
23 \LWR@htmlsmallfontstart{\,\,#1\LWR@htmlsmallfontend},

```

For `pdftotext`, do not scale the text:

```

24 scaling = false
25 }
26 \DeclareInstance{xfrac}{lmr}{text}{
27 numerator-format = {%
28 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
29 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
30 denominator-format = {%
31 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
32 \LWR@htmlsmallfontstart{\,\,#1\LWR@htmlsmallfontend},

```

For `pdftotext`, do not scale the text:

```

33 scaling = false
34 }
35 \DeclareInstance{xfrac}{lmss}{text}{
36 numerator-format = {%
37 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
38 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
39 denominator-format = {%
40 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
41 \LWR@htmlsmallfontstart{\,\,#1\LWR@htmlsmallfontend},

```



For `pdftotext`, do not scale the text:

```
42 scaling = false
43 }
44 \DeclareInstance{xfrac}{lmtt}{text}{
45 numerator-format = {%
46 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
47 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
48 denominator-format = {%
49 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
50 \LWR@htmlsmallfontstart{ }\, #1\LWR@htmlsmallfontend},
```

For `pdftotext`, do not scale the text:

```
51 scaling = false
52 }
```

```
53 \end{warpHTML}
```

## Package 121

# lwarp-xmpincl.sty

## 195 Xmpincl

Pkg xmpincl Emulated.

for **HTML output**: Discard all options for lwarp-xmpincl:

```
1 \LWR@ProvidesPackageDrop{xmpincl}
```

```
2 \newcommand*{\includemp}[1]{}
```

# Change History and Index

## Change History

v0.10		Docs: Commands for a successful HTML conversion. . . . . 70
General: 2016/03/08 Initial version	1	
v0.11		Docs: Commands into a warpprint environment. . . . . 67
General: 2016/03/11 . . . . .	1	Docs: Newclude limitations. . . 84
Added section:		Docs: Table: Cross-referencing data structures. . . . . 277
Operating-System portability.	99	Docs: Table: Float data structures. . . . . 286
Added section: Selecting the operating system. . . . .	66	Docs: Trademarks section. . . 369
Test Suite: limages and index in README.txt . . . . .	1	Docs: Troubleshooting cross-references. . . . . 91
Test Suite: MS-Windows in README.txt . . . . .	1	Test Suite: Assigned cleveref name for Test Float. . . . . 1
v0.12		Test Suite: Floatrow . . . . . 1
\LWR@newhtmlfile: Bugfix: TOC with numbered files. . . . .	204	v0.15
General: 2016/03/14 . . . . .	1	\printpublished: No div if empty. . . . . 227
Global: Uses \p@(type) in float captions. . . . .	1	\printsubtitle: No div if empty. 227
Test Suite: Sub-figures . . . . .	1	General: 2016/04/06 . . . . . 1
v0.13		Added . . . . . 415
\CaptionSeparator: Fix for newer babel package. . . . .	288	Ampersand (&): Fixed handling when passed as an argument. 248
\LWR@LwarpStart: \up and \fup .	217	Docs: Added warning icons for items needing special attention. 96
\published: Default to empty published. . . . .	223	Docs: Clarify print/HTML output. . . . . 66
\subtitle: Default to empty subtitle. . . . .	224	Docs: Moved the supported functions table to the introduction. . . . . 29
General: 2016/03/24 . . . . .	1	Files: lwarp_formal.css added. . 1
Removed package: subfig . . . . .	1	Fix: steps counter . . . . . 414
Test Suite: Ordinals, Subcaption	1	Fixed & handling. . . . . 412
tikzpicture: Fix dollar-redefined bug for newer package. . . . .	510	Test Suite: test_suite_formal.css file added. . . . . 1
v0.14		v0.16
\LWR@htmlsectionfilename: Fix: Links to home page. . . . .	175	General: 2016/04/11 . . . . . 1
General: 2016/03/31 . . . . .	1	
floatrow: Added. . . . .	410	

<code>\tittingpage</code> : Improved print-output spacing. . . . .	228	<code>\LWR@subhyperrefclass</code> : Improved HTML output linebreaks. . . .	284
Added XeLaTeX, LuaLaTeX support. . . . .	107	<code>\LWR@subinlineimage</code> : Surpress extra space. . . . .	285
Docs: Font and UTF-8 support. . . .	63	General: 2016/05/19 . . . . .	1
Docs: Moved location of <code>\usepackage{lwarp}</code> . . . . .	64	File: lwarp.css: Improved TOC outline display. . . . .	1
Docs: Text not converting. . . . .	91	Files: lwarp.css and lwarp_formal.css: Improved responsive design. . . . .	1
Fix: amsmath options clash . . . . .	111	Microtype disabled during HTML generation . . . . .	107
Fix: newtxmath compatibility. . . . .	111	PDF Unicode input characters. . . .	97
Lwarp no longer selects fonts. . . . .	63, 107	Test Suite: Verse package . . . . .	1
Removed package: suffix . . . . .	1	<code>lateximage</code> : pdfcrop: --hires added. . . . .	321
Test Suite: Improved tittingpage. . . . .	228	Reorganize <code>\HomeHTMLFilename</code> logic. . . . .	321
Test Suite: Lwarp no longer selects fonts. . . . .	1	Surpress extra space. . . . .	321
Test Suite: Supports XeLaTeX, LuaLaTeX. . . . .	1	<code>verse</code> : Supports verse, memoir packages. . . . .	529
<code>xfrac</code> : Adjusted for the use of any font: . . . . .	544	<code>minipage</code> : Fix: <code>\linewidth</code> , <code>\textwidth</code> , <code>\textheight</code> inside a minipage. . . . .	346
v0.17 <code>\LWR@htmlsectionfilename</code> : Fix: Links when entire doc is one HTML page. . . . .	175	v0.19 <code>\HTMLFilename</code> : Docs: Escape filename underscores. . . . .	174
General: 2016/04/14 . . . . .	1	<code>\HomeHTMLFilename</code> : Docs: Escape filename underscores. . . . .	174
<code>mdframed</code> : Added. . . . .	449	<code>\LWR@LwarpStart</code> : Enabled <code>\\</code> equal to <code>\newline</code> . . . . .	216
Test Suite: Fix: Print-version front-matter page numbers. . . . .	1	<code>\LWR@doubledollar</code> : MathJax support. . . . .	304
Test Suite: Mdfamed . . . . .	1	<code>\LWR@filestart</code> : lwarp_mathjax.txt loaded. . . .	214
v0.18 <code>\LWR@hspace</code> : <code>\hspace</code> supported. . . .	361	<code>\LWR@hspace</code> : Fix: <code>\hspace</code> length computations. . . . .	361
<code>\LWR@includegraphicsb</code> : Add: svgz file extension. . . . .	331	<code>\LWR@minipagestartpars</code> : Surpresses paragraph tags between minipages. . . . .	360
em, ex, %, px dimensions preserved. . . . .	331	<code>\LWR@singledollar</code> : MathJax support. . . . .	304
Fix: <code>\linewidth</code> , <code>\textwidth</code> , <code>\textheight</code> inside a minipage. . . . .	331	<code>\LWR@tabledatamultirowtag</code> : Added optional args. . . . .	271
Improved HTML output linebreaks. . . . .	331	<code>\LateximageFontSizeName</code> : Add: User-adjustable math/lateximage font size. . . .	320
<code>\LWR@myshorttoc</code> : Reorganize <code>\HomeHTMLFilename</code> logic. . . .	292	<code>\minipagefullwidth</code> : Added: No width tag for the next minipage in HTML. . . . .	346
<code>\LWR@newhtmlfile</code> : sideTOC after title, improving responsive design. . . . .	203		
<code>\LWR@requesttoc</code> : Reorganize <code>\HomeHTMLFilename</code> logic. . . .	219		
<code>\LWR@subhyperref</code> : Improved HTML output linebreaks. . . .	284		

<code>\rule</code> : Added	365	<code>\LWR@hspace</code> : Add: Supports HTML thin breakable space.	361
<code>\warpHTMLonly</code> : Added.	100	<code>\LWR@htmldivclass</code> : Added optional style.	185
<code>\warpprintonly</code> : Replaces <code>\rowprintedonly</code> .	100	<code>\LWR@htmlclass</code> : Added optional style.	184
<code>\xfracHTMLfontsize</code> : Added.	543	<code>\LWR@htmlsectionfilename</code> : HTMLFilename: removed additional trailing '-', and may be empty.	175
General: 2016/06/08	1	Sections called "Index" or "index" have an underscore prepended to their filenames if no prefix.	175
Avoids MathJax.	303	<code>\LWR@includegraphicsb</code> : Fix: <code>\linewidth</code> in a floatrow.	331
<code>cleveref</code> : Loaded		Fix: Expands filename.	331
<code>\AtEndPreamble</code> .	342	<code>\LWR@longtabledatacaptiontag</code> : Fix: Pars in captions.	266
CSS for table note item.	509	<code>\LWR@section</code> : Combined higher-level sections together into files.	208
Docs: <code>multirow</code> browser bug workaround.	81, 244	<code>\LWR@setOSWindows</code> : Auto-detects operating system.	99
Docs: Math options.	64	<code>\LWR@subhtmlclass</code> : Factored code.	184
Docs: Table: Cross-referencing data structures, updated.	277	<code>\SetHTMLFileNumber</code> : Add: Control file numbers.	174
File: <code>lwarp_mathjax.txt</code> added.	1	<code>\cpagerefFor</code> : User-redefinable word for page references.	343
File: <code>lwarp.css</code> : <code>tnoteitemheader</code> added	1	<code>\dotfill</code> : Inserts an ellipsis.	358
Introduction: MathJax support mentioned.	28	<code>\hfill</code> : Inserts a <code>\qqquad</code> .	358
MathJax support added.	306, 311, 312	<code>\hrulefill</code> : Inserts a short rule.	358
Options: <code>mathsvg</code> and <code>mathjax</code>	101	<code>\hyperindexref</code> : Print mode provided in case <code>hyperref</code> not used.	300
Supports colored <code>\rule</code> .	537	<code>\pageref</code> : Added.	283
titles: null <code>\pagestyle</code> and <code>\thispagestyle</code> for HTML.	512	<code>\tracinglwarp</code> : Added.	115
<code>tikzpicture</code> : Adapts to <code>tikz</code> version.	510	General: 2017/02/09	1
<code>equation</code> : MathJax support.	309	<code>afterpage</code> : Added.	372
v0.20		<code>alltt</code> : Added.	374
<code>\BlockClassSingle</code> : Renamed from "LWR@htmldivclassline".	186	<code>bookmark</code> : Added.	379
<code>\HTMLDescription</code> : Added <code>\NewHTMLdescription</code> . (Renamed in v0.30.)	196	caption and subcaption supported.	1
<code>\HTMLFilename</code> : No longer escape underscores.	174	<code>cleveref</code> and referencing patches: Applied <code>\AfterEndPreamble</code> .	342
<code>\HomeHTMLFilename</code> : No longer escape underscores.	174	<code>draftwatermark</code> : Added.	385
<code>\InlineClass</code> : Renamed from "inlineclass".	187	<code>eso-pic</code> : Added.	391
<code>\LWR@LwarpStart</code> : Fix: math cross references.	218	<code>everypage</code> : Added.	392
<code>\LWR@closeparagraph</code> : <code>\unskip</code> extra spaces.	190	<code>extramarks</code> : Added.	393
No break tags in the start/end of a tabular.	190	<code>fancyhdr</code> : Added.	397
<code>\LWR@endoffline</code> : Fix: <code>\\</code>	359		
<code>\LWR@filestart</code> : Adds meta description.	214		

hyperref: Additional user macros. ....	429	Test Suite: HTML meta descriptions. ....	1
keyfloat: Added. ....	435	Verbatim: Added. ....	399
letterspace: User-interface emulated. ....	437	verbatim: Added. ....	239
listings: Added. ....	440	BlockClass: Added optional style. ....	186
lrcaption: Added. ....	446	Renamed from "blockclass". ..	186
lwarp-newproject: Added. ....	119	LWR@nestspan: Fix: Minipages inside a span. ....	181
microtype: User-interface emulated. ....	458	v0.21	
needspace: Added. ....	465	\LWR@LwarpStart: Changed lateximages to a .txt file. .	216
nowidow: Added. ....	467	\LWR@filestart: Skip title if not given. ....	214
placeins: Added. ....	481	\LWR@newhtmlfile: Skip title if not given. ....	203
ragged2e: Added. ....	483	\marginpar: Fixed source listing. ....	200
setspace: Improved support. ..	487	General: 2017/02/23 .....	1
textpos: Added. ....	504	fontenc: Added. ....	417
titles: Added. ....	512	fontspec: Added. ....	417
titlesec: Added. ....	515	inputenc: Added. ....	434
titletoc: Added. ....	516	newclude: Added. ....	465
titling: Improved compatibility. ....	517	newunicodechar: Added. ....	466
tocloft: Added. ....	518	lwarpmk: Fix: lwarpmk again for Windows. ....	158
wallpaper: Added. ....	531	lwarpmk: Fix: lwarpmk limages for Windows. ....	158
wrapfig: Added. ....	532	lwarpmk: Fix: lwarpmk uses lateximages text file instead of shell script. ....	158
Added @, <, > columns. ....	245	Add: Errors for misplaced packages. ....	103
Added single-expansion data arrays. ....	173	Docs: Added <b>internet</b> class. .	34
Code factored into independent lwarp_html files. ....	369	Docs: Added TeX2page, GladTeX. ....	34
Docs: Examples for generating HTML file names. ....	56	Docs: Installing on Windows. .	40
Docs: Improved index. ....	1	File lwarp_tutorial.txt added. ....	45
Enhanced titling support. ....	225	v0.22	
File: lwarp.css: Minor fixes for validation. ....	1	\LWR@parseDcolumn: Added tabular D column. ....	255
File: lwarpmk used to compile print, HTML, indexes, and lateximages. ....	1	\LWR@parsebangcolumn: Added tabular ! column. ....	252
Fix: \linewidth in a floatrow. ....	413	\LWR@parsetablecols: Unknown table column types become l. Added tabular D, !, X columns. ....	255
Improved float caption type handling. ....	406	\LWR@printmccoldata: Added tabular D, !, and X columns. ....	264
Moved sidebar and example code to test suite. ....	1	General: 2017/03/02 .....	1
Page geometry set to 6in wide with large margins. ....	108	abstract: Added. ....	370
Parallel versions of aux files for print/HTML. ....	1	changepage: Added. ....	381
Removed reliance on make, grep, gawk. ....	1	dcolumn: Added. ....	384
Tabular: \unskip extra spaces. ....	245		

ftnright: Added. . . . .	425	emptypage: Added. . . . .	386
geometry: Nullified commands. . . . .	426	framed: Added. . . . .	422
indentfirst: Added. . . . .	434	lips: Added. . . . .	439
layout: Added. . . . .	437	mdframed: Help avoid hyphenation. . . . .	450
lscap: Added. . . . .	446	ntheorem: Added. . . . .	468
mcaption: Added. . . . .	448	showidx: Added. . . . .	489
nameref: Added. . . . .	464	theorem: Added. . . . .	505
nextpage: Added. . . . .	466	Basic L <sup>A</sup> T <sub>E</sub> X theorems: improved CSS. . . . .	239
parskip: Added. . . . .	480	Docs: Adds credits for patched code. . . . .	1
showkeys: Added. . . . .	489	Docs: Testing <code>lwarp</code> . . . . .	90
sidecap: Added. . . . .	490	Fix: Allows XeL <sup>A</sup> T <sub>E</sub> X and LuaL <sup>A</sup> T <sub>E</sub> X to preload graphics and graphicx. . . . .	104
tabularx: Added. . . . .	502		
varioref: Supported. . . . .	76		
verse: Added. . . . .	529		
v0.23		v0.26	
\LWR@parsetablecols: Fix for vert bar column type. . . . .	255	General: 2017/03/31 . . . . .	1
\LWR@printmccoldata: Fix for vert bar column type. . . . .	264	cutwin: Added. . . . .	383
General: 2017/03/02 . . . . .	1	endnotes: Added. . . . .	387
v0.24		floatflt: Added. . . . .	408
\LWR@hspace: Add: <code>\hspace \fill</code> converts to 2em . . . . .	361	footmisc: Added. . . . .	418
\LWR@htmlfileref: Fix: Index links while <code>\tracinglwarp</code> . . . . .	279	footnotehyper: Added. . . . .	421
\hypertocfloat: List of floats responds to <code>lofdepth</code> , <code>lotdepth</code> . . . . .	298	footnote: Added. . . . .	420
General: 2017/03/15 . . . . .	1	marginfix: Added. . . . .	447
floatrow: Support for <code>subfig</code> . . . . .	410	marginnote: Added. . . . .	448
subfig: Added. . . . .	495	mparhack: Added. . . . .	461
tikz: For tikz v3.0.0 or later, auto-loads tikz babel library if necessary. . . . .	510	pagenote: Supported as-is. . . . .	479
Docs: Filename underscore. . . . .	54, 71	sidenotes: Added. . . . .	491
No longer preloads <code>subcaption</code> ; conflicted with <code>subfig</code> . . . . .	110	<code>lwarp.css</code> : Improved responsive <code>marginpar</code> and <code>marginblock</code> . . . . .	121
picture: Fix for inline images. . . . .	343	Docs: Improved MiK <sub>T</sub> E <sub>X</sub> install instructions. . . . .	37, 40
tikzpicture: Fix for inline images. . . . .	510	Dollar span avoided in a <code>lateximage</code> . . . . .	303
v0.25		Footnotes now are L <sup>A</sup> T <sub>E</sub> X boxes instead of pagenotes. . . . .	196
\LWR@loadnever: Added the ability to prevent conflicting packages. . . . .	104	lateximage: Labels track page numbers of <code>lateximages</code> . . . . .	321
\addcontentsline: Handles theorems. . . . .	291	Print mode now uses a <code>minipage</code> of <code>\linewidth</code> . . . . .	321
General: 2016/03/22 . . . . .	1	picture: Fix for <code>\makebox</code> in picture. . . . .	343
amsthm: Added. . . . .	375	v0.27	
ccaption: Prevented. . . . .	381	\LWR@footnotetext: Fix for table footnote par tags. . . . .	198
ellipsis: Added. . . . .	385	General: 2017/04/04 . . . . .	1
		lettrine: Added. . . . .	438
		microtype: Fix with XeL <sup>A</sup> T <sub>E</sub> X, LuaL <sup>A</sup> T <sub>E</sub> X. . . . .	458

soul: Added. . . . .	493	Docs: Modfying lwarpmk and index processing. . . . .	90
ulem: Added. . . . .	526	File <code>lwarp_mathjax.txt</code> : Updated CDN repository. . . .	156
Docs: Installing utilities for MacOS. . . . .	42	Forced oneside to maintain large right margin. . . . .	108
Docs: Limitations of saveboxes. . . .	72		
Page geometry modified to reduce line overflow. . . . .	108		
v0.28		v0.29	
\@wrindex: Improved indexing. . . .	300	\LWR@includegraphicsb: Fix: Error when no optional arguments. . . . .	331
\HTMLauthor: Added \HTMLauthor. (Renamed in v0.30.) . . . . .	196	General: 2017/04/15 . . . . .	1
\LWR@LwarpEnd: If FormatEpub or FormatWordProcessor, no bottom nav. . . . .	219	lwarpmk: Add: language option for config files. . . . .	158
\LWR@LwarpStart: FormatWordProcessor forces single-file output. . . . .	216	Add: lwarpmklang option for lwarp. . . . .	101
\LWR@filestart: Adds HTML meta author. . . . .	214	Docs: Using a glossary . . . . .	61
\LWR@forcenewpage: Forces new PDF page before major environments. . . . .	178	File <code>*.lwarpmkconf</code> : Add: language option for config files. . . . .	120
\LWR@htmlcomment: Breaks ligatures in HTML comments. . . .	183	File <code>lwarpmk.conf</code> : Add: language option for config files. . . . .	119
\LWR@includegraphicsb: Adapts to graphics syntax. . . . .	331	v0.30	
\LWR@newhtmlfile: If FormatEpub or FormatWordProcessor: skips headers, footers, nav. . . . .	203	\CSSFilename: Renamed from \NewCSS. . . . .	195
\LWR@parsetablecols: Added L, C, R, J column types. . . . .	255	\HTMLauthor: Renamed from \HTMLauthor. . . . .	196
\LWR@startref: Removed space. . . .	281	\HTMLDescription: Renamed from \NewHTMLdescription. . . . .	196
\chapter: If EPUB, prints footnotes before each section. . . . .	212	\HTMLFirstPageTop: Renamed from \SetFirstPageTop. . . . .	194
\hyperindexref: Improved indexing. . . . .	300	\HTMLLanguage: Renamed from \MetaLanguage. . . . .	214
\textup: Fixed span class. . . . .	354	\HTMLPageBottom: Renamed from \SetPageBottom. . . . .	195
General: 2017/04/14 . . . . .	1	\HTMLPageTop: Renamed from \SetPageTop. . . . .	194
glossaries: Added. . . . .	427	General: 2017/04/29 . . . . .	1
graphics: Added. . . . .	428	lwarp-newproject removed, and combined with lwarp. . . . .	119
tabularx: Fix for optional pos. . . .	502	lwarpmk: Add: xdyfile configuration option. . . . .	158
tabulary: Added. . . . .	503	lwarpmk: Fix: xindy and texindy adjusted for pdflatex, xelatex and lualatex. . . . .	158
lwarpmk: Add: printglossary and htmlglossary commands. . . .	158	lwarpmk: Fix: xindy now used for print index generation with latexmk. . . . .	158
Added boolean FormatEpub. . . . .	115		
Added boolean FormatWordProcessor. . . . .	116		
Added boolean HTMLDebugComments. . . . .	115		
Added boolean HTMLMarkFloats . . .	116		



lwarpmk: language now used for both index and glossary generation. . . . .	158	\StartDefiningTabulars: Add: Avoids error: misplaced alignment tab character &. . . . .	248
File: lwarpmk_html.xdy renamed to lwarpmk.xdy. . . . .	156	General: 2017/07/10 . . . . .	1
Fix: *.css files only written in print mode. . . . .	121	amsmath: Removed fleqn option. . . . .	111
Fix: lwarpmk.xdy only written in print mode. . . . .	156	fancyhdr: Fix: Optional args for \lhead, etc. . . . .	397
Fix: lwarpmk_mathjax.txt only written in print mode. . . . .	156	Add: Tabular at and bang columns now have their own HTML columns. . . . .	245
Option OSWindows replaces macro \warpOSWindows. . . . .	102	cleveref: Fix: Loaded \AtEndPreamble. . . . .	342
Option latexmk replaces macro \UseLatexmk. . . . .	102	Fix: Incorrectly-inline math environments. . . . .	312
Option lwarpmklang changed to IndexLanguage. . . . .	101	New handling of & to localize catcode changes. . . . .	245
Option xdyFilename added. . . . .	102	v0.34	
Options HomeHTMLFilename and HTMLFilename replace macros \HomeHTMLFilename and \HTMLFilename. . . . .	102	\@fnsymbol: Text symbols instead of math. . . . .	232
v0.31		\InlineClass: Moved optional argument in front of mandatory. . . . .	187
General: 2017/05/15 . . . . .	1	\LWR@htmldivclass: Moved optional argument in front of mandatory. . . . .	185
keyfloat: Improved compatibility. . . . .	435	\LWR@htmlclass: Moved optional argument in front of mandatory. . . . .	184
v0.32		\LWR@htmlclassline: Moved optional argument in front of mandatory. . . . .	185
\RequirePackage: Fix: Ignores blanks in package list. . . . .	112	\LWR@htmlspanclass: Moved optional argument in front of mandatory. . . . .	182
General: 2016/06/09 . . . . .	1	\LWR@nullfont: Improved font control. . . . .	355
glossaries: Prevent error with \glo@name not defined. . . . .	301	\LWR@restoreorigformatting: booktabs: Works inside lateximage. . . . .	307
lwarpmk: Fix: io.lines() changed to file:lines() due to luatex changes. . . . .	158	Improved font control. . . . .	307
v0.33		\LWR@subhtmlclass: Moved optional argument in front of mandatory. . . . .	184
\HTMLAuthor: Fix: Provides empty default author if none given. . . . .	196	\LWR@tabledatacolumnntag: booktabs: Works inside lateximage. . . . .	271
\LWR@loadbefore: Fix: No \PackageError if already loaded. . . . .	104	\fboxBlock: Added. . . . .	351
\LWR@parseatcolumn: Fix: Column alignment with leftmost @. . . . .	251	\makebox: Fix: Handles paren arg. . . . .	349
\LWR@tabledatasinglecolumnntag: Fix: Macros in tabular could cause extra data cell. . . . .	260	General: 2017/08/08 . . . . .	1
\LWR@vspace: Add: \vspace nullified. . . . .	362		

babel-french: Adds fixed-width HTML spaces to punctuation. . . . .	179	Verbatim refactored to remove fancyvrb requirement. . . . .	237
balance: Added. . . . .	378	lateximage: Fix: lateximage with minipage, \parbox, \makebox, \fbox, \framebox, \raisebox, \scalebox, \reflectbox. . . . .	321
booktabs: Works inside lateximage. . . . .	273, 379	fcolorminipage: Added. . . . .	540
boxedminipage2e: Added. . . . .	380	BlockClass: Moved optional argument in front of mandatory. . . . .	186
boxedminipage: Prevented. . . . .	380	fminipage: Added. . . . .	351
crop: Added. . . . .	382	LWR@nestspan: Fix: Minipages, BlocksClass, and lists inside a span. . . . .	181
enumerate: Added. . . . .	388	LWR@tabular: booktabs: Works inside lateximage. . . . .	273
enumitem: Added, no longer required. . . . .	389	v0.35	
everyshi: Added. . . . .	393	General: 2017/08/08 . . . . .	1
fancybox: Added. . . . .	395	Fix: \textbf and related. . . . .	353
fancyvrb: Added, no longer required. . . . .	399	v0.36	
figcaps: Added. . . . .	405	\LWR@HTMLsanitize: Fix for babel-french. . . . .	320
filecontents: Required. Patched for morewrites. . . . .	109	\LWR@HTMLsanitizeexpand: Fix for babel-french. . . . .	321
floatpag: Added. . . . .	409	\LWR@closeparagraph: Extra HTML source space after paragraphs. . . . .	190
flushend: Added. . . . .	416	\LWR@currenttextcolor: Fix for \rule when xcolor not loaded. . . . .	363
fullpage: Added. . . . .	425	\LWR@footnotetext: Extra HTML source space after paragraphs. . . . .	198
hyperxmp: Added. . . . .	432	Force HTML superscripts. . . . .	198
idxlayout: Added. . . . .	433	\LWR@nullfonts: Fix: Filenames while using MathJax. . . . .	355
marginfit: Added. . . . .	447	\LWR@restoreorigformatting: siunitx: Improved super/subscripts in a lateximage. . . . .	307
mdframed: Improved mdtheorem patch. . . . .	455	\LWR@section: Improved spacing. . . . .	208
moreverb: Added. . . . .	459	\LWR@stoppars: Extra HTML source space after paragraphs. . . . .	192
paralist: Added. . . . .	479	\fbox: Fix: Uses \fboxrule and \fboxsep. . . . .	350
pdfscape: Added. . . . .	480	\framebox: Fix: Handles width and horiz position. . . . .	349
pdfsync: Added. . . . .	481	\makebox: Fix: Handles width and horiz position. . . . .	349
prelim2e: Added. . . . .	482	General: 2017/08/17 . . . . .	1
rotfloat: Added. . . . .	485	babel-french: Adjustements for French variants, load order, footnotes, ellipses. . . . .	179
savetrees: Added. . . . .	486		
shadow: Added. . . . .	488		
syntonly: Added. . . . .	501		
titles: No longer required. . . . .	512		
titleref: Prevented. . . . .	514		
xmpincl: Added. . . . .	546		
Docs: Horizontal space limitations. . . . .	1		
Docs: Misplaced alignment character. . . . .	91		
File: README.txt: updated. . . . .	1		
File: lwarp_mathjax.txt version change. . . . .	156		
Fix: Added the eqnarray environments. . . . .	312		
Improved font control. . . . .	353		
Lists refactored to remove enumitem requirement. . . . .	241		

footnote: Extra HTML source space after paragraphs. ....	420	xcolor: Uses <code>\fboxrule</code> and <code>\fboxsep</code> . ....	534
siunitx: Fix for babel-french. ..	326	xcolor: <code>\fcolorbox</code> etc. now work inside <code>lateximage</code> . ...	534
siunitx: Improved symbol support. ....	118	Docs: Reorganized: Special cases and limitations. ....	70
transparent: Added. ....	523	Source: Improved formatting. ..	1
upref: Added. ....	528	<code>lateximage</code> : Footnotes appear in regular text instead of the <code>lateximage minipage</code> . ....	321
xcolor: Added <code>\fcolorboxBlock</code> , <code>\colorboxBlock</code> . ....	534	<code>LWR@tabular</code> : Fix for babel-french.	273
xcolor: Fix: Background <code>none</code> in print mode. ....	534		
xcolor: Refactored <code>\LWR@colorstyle</code> . ....	537		

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols			
<code>\\$</code> .....	<i>303</i>	<code>\@subtitle</code> .....	<i>222</i>
<code>\&amp;</code> .....	<i>174</i> , <i>4742</i>	<code>\@title</code> .....	<i>222</i>
<code>\(</code> .....	<i>5865</i>	<code>\@wrglossary</code> .....	<i>5806</i>
<code>\)</code> .....	<i>5865</i>	<code>\@wrindex</code> .....	<i>5799</i>
<code>\,</code> .....	<i>71</i>	<code>\</code> .....	<i>359</i>
<code>\@@setcpageref</code> .....	<i>6826</i>	<code>\$</code> .....	<i>304</i>
<code>\@@setcref</code> .....	<i>6822</i>	<code>\$\$</code> .....	<i>304</i>
<code>\@@setcrefrange</code> .....	<i>6823</i>	<code>~</code> .....	<i>71</i>
<code>\@author</code> .....	<i>222</i>	<b>A</b>	
<code>\@begintheorem</code> .....	<i>4590</i>	<code>abstract</code> (environment) .....	<i>4498</i>
<code>\@capytype</code> .....	<i>5555</i>	<code>abstract</code> (package) .....	<i>370</i>
<code>\@currentlabelname</code> .....	<i>5335</i>	<code>\abstractname</code> .....	<i>56</i> , <i>4497</i>
<code>\@date</code> .....	<i>222</i>	accents .....	<i>206</i>
<code>\@dlbfloat</code> .....	<i>5533</i>	<code>\addcontentsline</code> .....	<i>5609</i>
<code>\@endtheorem</code> .....	<i>4600</i>	Adobe (program) .....	<i>36</i>
<code>\@ensuredmath</code> .....	<i>5872</i>	<code>\affiliation</code> .....	<i>4163</i>
<code>\@float</code> .....	<i>5533</i>	<code>afterpage</code> (package) .....	<i>372</i>
<code>\@fnsymbol</code> .....	<i>4457</i>	algorithmic	
<code>\@footnotetext</code> .....	<i>3631</i>	with newfloat, trivfloat .....	<i>524</i>
<code>\@makecaption</code> .....	<i>5557</i>	<code>algorithmicx</code> (package) .....	<i>373</i>
<code>\@makefnmark</code> .....	<i>3603</i>	<code>align</code> (environment) .....	<i>6189</i>
<code>\@makefntext</code> .....	<i>3602</i>	<code>align*</code> (environment) .....	<i>6212</i>
<code>\@maketitle</code> .....	<i>4469</i>	alignment tab character <code>&amp;</code> , misplaced	
<code>\@mpfootnotetext</code> .....	<i>3632</i>	.....	<i>81</i> , <i>244</i> , <i>248</i>
<code>\@opargbegintheorem</code> .....	<i>4595</i>	<code>alltt</code> (package) .....	<i>374</i>
<code>\@published</code> .....	<i>222</i>	<code>\AmS</code> .....	<i>7336</i>

AMSMath			
split miss-numbered	. 77, 79, 301, 468		
amsmath (package)	111		
amsthm (package)	375		
\and	222		
array (package)	276		
AsciiDoc (program)	35		
AsciiDoctor (program)	35		
Asciidoctor-LaTeX (program)	35		
\attrib	84, 236, 529		
\attribution	4508		
author			
HTML meta tag	69, 196		
\author	68, 222, 4170		
autosec	207		
<b>B</b>			
babel (package)	85		
\backmatter	3820		
balance (package)	378		
\BaseJobname	3059		
BaseJobname (option)	101		
baseline			
tabular	256		
\bfseries	7142		
\BibTeX	7331		
BlockClass (environment)	3377		
\BlockClassSingle	3389		
bookmark (package)	379		
booktabs (package)	379		
booleans:			
CombineHigherDepths	55, 201		
FileSectionNames	56, 175		
FormatEPUB	86, 115		
FormatWordProcessor	88, 116		
HTMLDebugComments	115		
HTMLMarkFloats	88, 116		
LWR@amsmultline	311		
LWR@doingapar	188		
LWR@doinghline	246		
LWR@doingstartpars	188		
LWR@doingtbrule	246		
LWR@freezethisfloat	288		
LWR@infloatrow	331		
LWR@intabularmetadata	246		
LWR@minipagefullwidth	346		
LWR@minipagethispar	346		
LWR@skipatbang	246		
LWR@skippingmrowcell	246		
LWR@starredlongtable	266		
LWR@startedrow	246		
LWR@tableparcell	246		
LWR@tracinglwarp	115		
LWR@validtablecol	255		
LWR@verbtags	237		
mathjax	100		
usingOSWindows	99		
warpingHTML	100		
warpingprint	100		
boxedminipage (package)	380		
boxedminipage2e (package)	380		
\boxframe	138		
bugs	91		
BVerbatim (environment)	181		
<b>C</b>			
calc (package)	110		
Calibre	86		
caption (package)	110		
\caption@begin	5588		
\caption@end	5588		
\captionlistentry	5592		
\captionof	5622		
\CaptionSeparator	5556		
ccaption (package)	381		
center (environment)	6425		
\centering	217		
changepage (package)	381		
\chapter	3948		
\citetitle	4515		
classes:			
internet	34		
cleveref (package)	342		
cmap (package)	64		
\color	538		
\colorbox	46		
\colorboxBlock	56, 6722		
CombineHigherDepths (boolean)	55, 201		
comment (package)	98		
Computer Modern	63		
\ConTeXt	7328		
counters:			
FileDepth	55, 201		
lofdepth	297		
lotdepth	297		
LWR@externalfilecnt	303		
LWR@htmlfilenumber	175		
LWR@latestautopage	289		
LWR@lateximagedepth	320		
LWR@lateximagenumber	319		





GELLMU (program) .....	34
generator .....	HTMLleftmargini (length)..... 84, 236, 237, 529
HTML meta tag .....	215
geometry (package) .....	108, 426
getttitlestring (package) .....	109
GladTeX (program) .....	35
glossaries (package) .....	85, 427
glossary .....	
language .....	61
processing .....	61
glyphtounicode (file) .....	63
graphics (package) .....	326, 428
\graphicspath .....	6454
graphicx (package) .....	327, 428
<b>H</b>	
Hevea (program) .....	34
\hfill .....	7156
\HomeHTMLFilename .....	3061
HomeHTMLFilename (option) ....	54, 102
horizontal space .....	71
between minipages .....	360
\href .....	5478
\hrulefill .....	7157
\hskip .....	71
\hspace .....	71, 360, 7252
HTML .....	
commands for a successful conver-	sion .....
sion .....	70
conversion settings .....	54
debug comments .....	115
EPUB .....	86, 115
word processor .....	88, 116
filename generation .....	56
headings .....	94
in place of SVG images .....	51, 92
meta tag .....	
author .....	69, 196
description .....	68, 196
generator .....	215
viewport .....	215
tabular column conversion .....	256
\HTMLAuthor .....	57, 69, 3587
HTMLDebugComments (boolean) ....	115
\HTMLDescription .....	57, 68, 3593
\HTMLEntity .....	3042
\HTMLFilename .....	3060
HTMLFilename (option) .....	54, 102
\HTMLFirstPageTop .....	57, 3550
\HTMLLanguage .....	57, 4006
<b>I</b>	
icon .....	
warning .....	96
idxlayout (package) .....	433
ifplatform (package) .....	98
images .....	
graphicx package .....	327
in strange places .....	92
showing as HTML .....	51, 92
\includegraphics .....	6646
indentfirst (package) .....	434
InDesign (program) .....	36
index .....	
language .....	61
processing .....	49, 50
UTF-8 .....	64
IndexLanguage (option) .....	61, 85, 101, 427
\InlineClass .....	3397
inputenc (package) .....	63, 434
internet (class) .....	34
item .....	
empty .....	71, 240
itemize (environment) .....	4612
\itshape .....	7147
<b>J</b>	
JavaScript .....	
MathJax .....	77, 301
<b>K</b>	
\kern .....	71
keyfloat (package) .....	435
kvoptions (package) .....	100
<b>L</b>	
label .....	
in HTML .....	218
math environment .....	311

language		\LuaTeX	7322
glossary	61	lwrap	
index	61	loading	64
language HTML metadata	214	options	64
\LaTeX	7314	lwrap (package)	64
LaTeX2HTML (program)	34	lwrap.css (file)	69, 121
\LaTeXe	7314	lwrap.xdy	
lateximage (environment)	6335	customizing	70
\LateximageFontSizeName	6293	lwrap.xdy (file)	70, 156
lateximages		lwrap_formal.css (file)	151
font size	77, 301, 319	lwrap_mathjax.txt (file)	156
latexmk (option)	54, 102	lwrap_sagebrush.css (file)	146
LaTeXML (program)	34	lwrap_tutorial.txt (file)	45
layout (package)	437	lwrapmk	
lengths:		customizing	90
HTMLleftmargini	84, 236, 237, 529	lwrapmk (option)	102
HTMLvleftskip	84, 236, 237, 529	lwrapmk (program)	90, 158
LWR@minipageheight	345	lwrapmk.conf (file)	119
LWR@minipagewidth	344	lwrapmk.lua (file)	90
LWR@tempheight	361	\LWR@addmathjax	5985
LWR@tempraise	361	\LWR@afterendverbatim	4564
LWR@tempwidth	361	\LWR@amsmultline (boolean)	311
\VerbatimHTMLWidth	237	\LWR@atbeginverbatim	4550
vleftmargini	84, 236, 529	\LWR@beginhideamsmath	6030
vleftskip	84, 236, 529, 530	\LWR@blackborderpadding	6990
letltxmacro (package)	98	\LWR@borderpadding	29
letterspace (package)	437	\LWR@botnavigation	3119
lettrine (package)	438	\LWR@caption@begin	5569
LibreOffice (program)	35	\LWR@caption@end	5579
ligatures	63, 64, 108	\LWR@clearmidrules	4996
line numbers	96	\LWR@closeparagraph	3465
\linebreak	7255	\LWR@closeprevious	3139
\LinkHome	3111	\LWR@closetabledatacell	4703
Linux (program)	66, 99	LWR@colafterspec	251
lips (package)	439	LWR@colatspec	251
list item, empty	71, 240	LWR@colbangspec	251
listings (package)	440	LWR@colbeforespec	251
\listof	5690	\LWR@copyfile	385
lmodern (package)	63	\LWR@createautosec	3826
lofdepth (counter)	297	\LWR@currentcss	3572
longtable (environment)	2	\LWR@currenttextcolor	7262
longtable (package)	444	\LWR@descitem	4640
lotdepth (counter)	297	\LWR@docmidrule	5015
lscape (package)	446	LWR@doingapar (boolean)	188
ltcaption (package)	446	LWR@doinghline (boolean)	246
LuaLaTeX		LWR@doingstartpars (boolean)	188
detection	107	LWR@doingtbrule (boolean)	246
file & section names	206	\LWR@domulticolumn	5089
\LuaLaTeX	7322	\LWR@doubledollar	5839
LuaLaTeX (program) [requirement]	38	\LWR@endhideamsmath	6038



<code>\LWR@endminipage</code>	6852	<code>LWR@latestautopage</code> (counter)	289
<code>\LWR@endoffline</code>	7205	<code>LWR@lateximagedepth</code> (counter)	320
<code>\LWR@endsubminipage</code>	6873	<code>\LWR@lateximagedepthref</code>	5365
<code>\LWR@ensuredoingapar</code>	3438	<code>LWR@lateximagenumber</code> (counter)	319
<code>\LWR@epubprintpendingfootnotes</code>	3663	<code>\LWR@lateximagenumberref</code>	5368
<code>LWR@externalfilecnt</code> (counter)	303	<code>LWR@LIpage</code> (counter)	320
<code>\LWR@filenamoblanks</code>	3683	<code>\LWR@loadafter</code>	98
<code>\LWR@filestart</code>	4013	<code>\LWR@loadbefore</code>	109
<code>\LWR@findword</code>	320	<code>\LWR@loadnever</code>	118
<code>\LWR@floatbegin</code>	5517	<code>\LWR@longtabledatacaptiontag</code>	5108
<code>\LWR@floatend</code>	5535	<code>\LWR@lookforpackagename</code>	323
<code>\LWR@footnotetext</code>	3604	<code>\LWR@LwarpEnd</code>	4140, 7358
<code>\LWR@forceminwidth</code>	6979	<code>\LWR@LwarpStart</code>	4063, 7358
<code>\LWR@forcenewpage</code>	3145	<code>\LWR@maketitlesetup</code>	4448
<code>LWR@freezethisfloat</code> (boolean)	288	<code>\LWR@maybeinthisfloat</code>	5552
<code>\LWR@futurenonospacelet</code>	4680	<code>\LWR@maybeinthisfloat</code>	4926
<code>\LWR@FVstyle</code>	17	<code>LWR@midrulecounter</code> (counter)	261
<code>\LWR@getexpparray</code>	3038	<code>LWR@midrules</code>	261
<code>\LWR@getmynexttoken</code>	4687	<code>LWR@minipagedepth</code> (counter)	344
<code>\LWR@hidelatexequation</code>	5971	<code>LWR@minipagefullwidth</code> (boolean)	346
<code>\LWR@hspace</code>	7230	<code>LWR@minipageheight</code> (length)	345
<code>\LWR@htmlblockcomment</code>	3319	<code>\LWR@minipagestartpars</code>	7210
<code>\LWR@htmlblocktag</code>	3327	<code>\LWR@minipagestoppars</code>	7213
<code>\LWR@htmlclosecomment</code>	3291	<code>LWR@minipagethispar</code> (boolean)	346
<code>\LWR@htmlcomment</code>	3312	<code>LWR@minipagewidth</code> (length)	344
<code>\LWR@htmldivclass</code>	3354	<code>\LWR@multicolother</code>	5040
<code>\LWR@htmldivclassend</code>	3357	<code>\LWR@multicolpartext</code>	5035
<code>\LWR@htmlmlelement</code>	3367	<code>\LWR@multicolskip</code>	5046
<code>\LWR@htmlmlelementclass</code>	3341	<code>\LWR@mynexttoken</code>	4679
<code>\LWR@htmlmlelementclassend</code>	3346	<code>\LWR@myshorttoc</code>	5634
<code>\LWR@htmlmlelementclassline</code>	3360	<code>\LWR@nameref</code>	5358
<code>\LWR@htmlmlelementend</code>	3370	<code>LWR@nestspan</code> (environment)	3225
<code>LWR@htmlfilenumber</code> (counter)	175	<code>\LWR@newhtmlfile</code>	3749
<code>\LWR@htmlfileref</code>	5361	<code>\LWR@newlabel</code>	5395
<code>\LWR@htmlmathlabel</code>	6016	<code>\LWR@newref</code>	5445
<code>\LWR@htmlopencomment</code>	3291	<code>LWR@nextautofloat</code> (counter)	289
<code>\LWR@htmlrefsectionfilename</code>	3104	<code>LWR@nextautopage</code> (counter)	289
<code>\LWR@HTMLsanitize</code>	6296	<code>LWR@nextequation</code> (counter)	306
<code>\LWR@HTMLsanitizeexpand</code>	6317	<code>\LWR@nohspace</code>	7251
<code>\LWR@htmlsectionfilename</code>	3071	<code>\LWR@nullfonts</code>	7108
<code>\LWR@htmlspan</code>	3266	<code>\LWR@openparagraph</code>	3443
<code>\LWR@htmlspanclass</code>	3274	<code>\LWR@origcolspec</code>	4772
<code>\LWR@htmltag</code>	3282	<code>\LWR@originname</code>	6658
<code>\LWR@htmltagc</code>	3215	<code>\LWR@originnames</code>	6665
<code>\LWR@includegraphicsb</code>	6548	<code>\LWR@parseaftercolumn</code>	4832
<code>\LWR@indexitem</code>	5787, 5791, 5795	<code>\LWR@parseatcolumn</code>	4778
<code>\LWR@indexsection</code>	5771	<code>\LWR@parsebangcolumn</code>	4802
<code>LWR@infloatrow</code> (boolean)	331	<code>\LWR@parsebeforecolumn</code>	4823
<code>LWR@intabularmetadata</code> (boolean)	246	<code>\LWR@parseDcolumn</code>	4860
<code>\LWR@itemizeitem</code>	4605	<code>\LWR@parsenormalcolumn</code>	4845

<code>\LWR@parsepcolumn</code> . . . . .	4856	<code>\LWR@tablecolspec</code> . . . . .	4769
<code>\LWR@parseskipcolumn</code> . . . . .	4842	<code>LWR@tablecolspos (counter)</code> . . . . .	251
<code>\LWR@parsetablecols</code> . . . . .	4865	<code>LWR@tablecolswidth (counter)</code> . . . . .	251
<code>\LWR@patchlists</code> . . . . .	4668	<code>\LWR@tabledatacolumnntag</code> . . . . .	5215
<code>LWR@prevFileDepth (counter)</code> . . . . .	207	<code>\LWR@tabledatacolumnntag</code> . . . . .	5193
<code>\LWR@printatbang</code> . . . . .	4942	<code>\LWR@tabledatamultirowtag</code> . . . . .	5201
<code>\LWR@printmccoldata</code> . . . . .	5049	<code>\LWR@tabledatasinglecolumnntag</code> . . . . .	4956
<code>\LWR@printmccoltype</code> . . . . .	5019	<code>LWR@tableparcell (boolean)</code> . . . . .	246
<code>\LWR@printpendingfootnotes</code> . . . . .	3653	<code>LWR@tabletotalcols (counter)</code> . . . . .	251
<code>\LWR@printheftitle</code> . . . . .	4299	<code>LWR@tabletotalcolsnext (counter)</code> . . . . .	251
<code>\LWR@ProvidesPackageDrop</code> . . . . .	375	<code>LWR@tabular (environment)</code> . . . . .	5272
<code>\LWR@ProvidesPackagePass</code> . . . . .	363	<code>\LWR@tabularendoffline</code> . . . . .	4753
<code>\LWR@pushoneclose</code> . . . . .	3829	<code>\LWR@tabularhtmlcolumns</code> . . . . .	5174
<code>\LWR@requesttoc</code> . . . . .	4133	<code>LWR@tempheight (length)</code> . . . . .	361
<code>\LWR@requirepackagenames</code> . . . . .	319	<code>LWR@tempraise (length)</code> . . . . .	361
<code>\LWR@restoremoreorigxcolor</code> . . . . .	6802	<code>LWR@tempwidth (length)</code> . . . . .	361
<code>\LWR@restoreorigformatting</code> . . . . .	5905	<code>\LWR@thisfilename</code> . . . . .	3681
<code>\LWR@rotboxorigin</code> . . . . .	6657	<code>LWR@thisfloat (counter)</code> . . . . .	288
<code>\LWR@rotstyle</code> . . . . .	6537	<code>\LWR@thisnewfilename</code> . . . . .	3682
<code>\LWR@rule</code> . . . . .	7263	<code>\LWR@titlingmaketitle</code> . . . . .	4486
<code>\LWR@scalestyle</code> . . . . .	6540	<code>\LWR@topnavigation</code> . . . . .	3116
<code>\LWR@section</code> . . . . .	3836	<code>\LWR@traceinfo</code> . . . . .	405
<code>\LWR@sectionnumber</code> . . . . .	3823	<code>LWR@tracinglwarp (boolean)</code> . . . . .	115
<code>\LWR@setexparray</code> . . . . .	3033	<code>LWR@validtablecol (boolean)</code> . . . . .	255
<code>\LWR@setlatestname</code> . . . . .	5337	<code>LWR@verbtags (boolean)</code> . . . . .	237
<code>\LWR@setOSWindows</code> . . . . .	31	<code>\LWR@vspace</code> . . . . .	7253
<code>\LWR@sidetoc</code> . . . . .	5703	<code>LWRprint@fminipage (environment)</code> . . . . .	7036
<code>\LWR@singledollar</code> . . . . .	5851	<code>\LWRsetnextfloat</code> . . . . .	5560
<code>LWR@skipatbang (boolean)</code> . . . . .	246	<code>\LyX</code> . . . . .	7339
<code>LWR@skippingmrowcell (boolean)</code> . . . . .	246		
<code>LWR@spandepth (counter)</code> . . . . .	188		
<code>\LWR@splabel</code> . . . . .	5371		
<code>LWR@starredlongtable (boolean)</code> . . . . .	266		
<code>LWR@startedrow (boolean)</code> . . . . .	246		
<code>\LWR@startnewdepth</code> . . . . .	3830		
<code>\LWR@startpars</code> . . . . .	3502		
<code>\LWR@startref</code> . . . . .	5408		
<code>\LWR@stoppars</code> . . . . .	3518		
<code>\LWR@stripperiod</code> . . . . .	5336		
<code>\LWR@strresult</code> . . . . .	4770		
<code>\LWR@subcmidrule</code> . . . . .	5007		
<code>\LWR@subhtml elementclass</code> . . . . .	3332		
<code>\LWR@subhyperref</code> . . . . .	5469		
<code>\LWR@subhyperrefclass</code> . . . . .	5473		
<code>\LWR@subinlineimage</code> . . . . .	5510		
<code>\LWR@subminipage</code> . . . . .	6867		
<code>\LWR@subnewref</code> . . . . .	5439		
<code>\LWR@subsublabel</code> . . . . .	5374		
<code>\LWR@subtableofcontents</code> . . . . .	5649		
<code>\LWR@synchronmathjax</code> . . . . .	5885		

	<b>M</b>
<code>Mac OS (program)</code> . . . . .	66, 99
<code>Madcap (program)</code> . . . . .	36
<code>\mainmatter</code> . . . . .	3813
<code>\makebox</code> . . . . .	6951
<code>makeidx (package)</code> . . . . .	110
<code>\MakeIndex</code> . . . . .	7331
<code>\maketitle</code> . . . . .	68, 229, 4461
<code>\maketitlehookaa</code> . . . . .	221
<code>\maketitlehookaaa</code> . . . . .	221
<code>margin</code>	
<code>numbers</code> . . . . .	96
<code>tags</code> . . . . .	96
<code>marginfit (package)</code> . . . . .	447
<code>marginfix (package)</code> . . . . .	447
<code>marginnote (package)</code> . . . . .	448
<code>\marginpar</code> . . . . .	3670
<code>markup languages</code> . . . . .	36
<code>math</code>	
<code>font size — SVG</code> . . . . .	77, 301, 319



amsthm	375	fullpage	425
array	276	geometry	108, 426
babel	85	getttitlestring	109
balance	378	glossaries	85, 427
bookmark	379	graphics	326, 428
booktabs	379	graphicx	327, 428
boxedminipage	380	hyperref	284, 429
boxedminipage2e	380	hyperxmp	432
calc	110	idxlayout	433
caption	110	ifplatform	98
ccaption	381	indentfirst	434
changepage	381	inputenc	63, 434
cleveref	342	keyfloat	435
cmap	64	kvoptions	100
comment	98	layout	437
crop	382	letltxmacro	98
cutwin	383	letterspace	437
dcolumn	384	lettrine	438
draftwatermark	385	lips	439
ellipsis	385	listings	440
emptypage	386	lmodern	63
endnotes	387	longtable	444
enumerate	388	lscape	446
enumitem	389	ltcaption	446
environ	110	lwrap	64
epigraph	390	makeidx	110
eso-pic	391	marginfit	447
etoolbox	98	marginfix	447
everyhook	109	marginnote	448
everypage	392	mcaption	448
everyshi	393	mdframed	449
expl3	109	memoir	530
extramarks	393	microtype	64, 107, 458
fancybox	394	moreverb	459
fancyhdr	397	mparhack	461
fancyvrb	399	multicol	462
figcaps	405	multirow	270, 464
filecontents	109	nameref	464
float	406	needspace	465
floatflt	408	newclude	84, 465
floatpag	409	newfloat	110
floatrow	410	newtxmath	79
flushend	416	newunicodechar	63, 466
fontenc	63, 417	nextpage	466
fontspec	63, 417	nicefrac	79
footmisc	418	nowidow	467
footnote	420	ntheorem	468
footnotehyper	421	pagenote	479
framed	422	paralist	479
ftnright	425	parskip	480

pdfscape	480	\pagecolor	44
pdfsync	481	pagenote (package)	479
placeins	481	\pagenumbering	3150
prelim2e	482	\pageref	5452
ragged2e	483	\pagerefPageFor	5451
refcount	110	\pagestyle	3148
rotating	484	Pandoc (program)	35
rotfloat	485	\paragraph	3988
savetrees	486	paralist (package)	479
setspace	487	\parbox	6944
shadow	488	\parsemulticolumnalignment	5076
showidx	489	parskip (package)	480
showkeys	489	\part	3936
sidecap	490	pdfcrop (program) [requirement]	38
sidenotes	491	pdfLaTeX (program) [requirement]	38
siunitx	79, 326	pdfscape (package)	480
soul	493	pdfseparate (program) [requirement]	38, 43
subfig	495	pdfsync (package)	481
syntonly	501	pdftocairo (program) [requirement]	38, 43
tabularx	502	pdftotext (program) [requirement]	38, 43
tabulary	503	Perl	43
textcomp	64	perl (program) [requirement]	43
textpos	504	\phantomsection	7308
theorem	505	picture (environment)	343, 6843
threeparttable	509	placeins (package)	481
tikz	510	Plastex (program)	34
titleps	512	\popclose	3006
titleref	514	Poppler	38, 43
titlesec	515	\postpublished	221
titletoc	516	\postsubtitle	221
titling	110, 517	prelim2e (package)	482
tocloft	518	\prepublished	221
transparent	523	\presubtitle	221, 231
trivfloat	524	\printauthor	222, 4316, 4331
ulem	526	\printdate	222, 4321, 4333
units	79	\printindex	5774
upref	528	\printpublished	222, 4282, 4328
verse	529, 530	\PrintStack	3124
wallpaper	531	\prints subtitle	222, 4307, 4330
wrapfig	532	\printthanks	221, 225
xcolor	534	\printtitle	222, 4291, 4329
xfrac	111, 543	problems	91
xifthen	109	programs	
xmpincl	546	utility	37
xparse	108	programs:	
xstring	110	[requirement]:	
zref	111	LuaLaTeX	38
page		MathJax	38
inaccessible	55		
\pagebreak	7257		



SideTOCDepth (counter) . . . . .	55, 295	\textsubscript . . . . .	7153
\sidetocname . . . . .	5700	\textsuperscript . . . . .	7152
siunitx (package) . . . . .	79, 326	\texttt . . . . .	7079
soul (package) . . . . .	493	\textup . . . . .	7082
\sp . . . . .	7150	\tfl@chapter@fix . . . . .	524
space		\thanks . . . . .	68
horizontal . . . . .	71	\thanksmarkseries . . . . .	4490
between minipages . . . . .	360	\theauthor . . . . .	222
split		\thedata . . . . .	222
miss-numbered . . . . .	77, 79, 301, 468	theindex (environment) . . . . .	5782
stack depths . . . . .	94	theorem (package) . . . . .	505
\StartDefiningTabulars . . . . .	4734	\thepublished . . . . .	222
subequations		\thesubtitle . . . . .	222
MathJax . . . . .	78, 302	\thetitle . . . . .	222
subfig (package) . . . . .	495	\thispagestyle . . . . .	3149
\subparagraph . . . . .	3996	threeparttable (package) . . . . .	509
\subsection . . . . .	3971	tikz	
\subsubsection . . . . .	3979	catcodes . . . . .	510
\subtitle . . . . .	68, 4191	dollar redefined . . . . .	510
SVG		tikz (package) . . . . .	510
images showing as HTML . . . . .	51, 92	tikzpicture (environment) . . . . .	8
math summary . . . . .	77, 301	\title . . . . .	68
mathsvg option . . . . .	101	titlepage (environment) . . . . .	68, 4204
syntonly (package) . . . . .	501	titles (package) . . . . .	512
<b>T</b>			
\tableofcontents . . . . .	57	titleref (package) . . . . .	514
tabular		titlesec (package) . . . . .	515
baseline . . . . .	256	titletoc (package) . . . . .	516
HTML columnn conversion . . . . .	256	titling (package) . . . . .	110, 517
tabularx (package) . . . . .	502	titlingpage (environment) 68, 4210, 4223	
tabulary (package) . . . . .	503	tocdepth (counter) . . . . .	54
tagged equations		tocloft (package) . . . . .	518
Mathjax . . . . .	78, 302	\tracinglwrap . . . . .	93, 404
\TeX . . . . .	7311	transparent (package) . . . . .	523
TeX2page (program) . . . . .	34	trivfloat	
TeX4ht (program) . . . . .	34	with newfloat, algorithmic . . . . .	524
\textbf . . . . .	7070	trivfloat (package) . . . . .	524
\textcolor . . . . .	34	troubleshooting . . . . .	91
textcomp (package) . . . . .	64	HTML debug comments . . . . .	115
\textgreater . . . . .	3052	\ttfamily . . . . .	7145
\textit . . . . .	7085	TtH (program) . . . . .	34
\textless . . . . .	3052	tutorial.tex (file) . . . . .	45
\textmd . . . . .	7067	<b>U</b>	
\textnormal . . . . .	7094	ulem (package) . . . . .	526
textpos (package) . . . . .	504	underscore	
\textrm . . . . .	7073	filename . . . . .	54, 71
\textsc . . . . .	7088	Unicode	
\textsf . . . . .	7076	enhanced coverage . . . . .	63
\textsl . . . . .	7091	file & section names . . . . .	206
		input characters . . . . .	97

