



# The **lwarp** package

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

v0.31 — 2017/05/15

© 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 L<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. Several dozen L<sup>A</sup>T<sub>E</sub>X packages are supported with these high-level source compatibility replacements.

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

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

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

## License:

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

# Contents

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

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

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

<b>27</b>	<b>Stacks</b>	<b>161</b>
27.1	Assigning depths . . . . .	161
27.2	Closing actions . . . . .	162
27.3	Closing depths . . . . .	162
27.4	Pushing and popping the stack . . . . .	163
<b>28</b>	<b>Data arrays</b>	<b>164</b>
<b>29</b>	<b>HTML entities</b>	<b>165</b>
<b>30</b>	<b>HTML filename generation</b>	<b>166</b>
<b>31</b>	<b>Homepage link</b>	<b>168</b>
<b>32</b>	<b>\PrintStack diagnostic tool</b>	<b>169</b>
<b>33</b>	<b>Closing stack levels</b>	<b>169</b>
<b>34</b>	<b>Forcing a new PDF page</b>	<b>170</b>
<b>35</b>	<b>HTML tags, spans, divs, elements</b>	<b>170</b>
35.1	Mapping L <sup>A</sup> T <sub>E</sub> X Sections to HTML Sections . . . . .	170
35.2	HTML tags . . . . .	171
35.3	Block tags and comments . . . . .	172
35.4	Div class and element class . . . . .	173
35.5	Single-line elements . . . . .	174
35.6	HTML5 semantic elements . . . . .	175
35.7	High-level block and inline classes . . . . .	175
35.8	Closing HTML tags . . . . .	176
<b>36</b>	<b>Paragraph handling</b>	<b>177</b>
<b>37</b>	<b>Paragraph start/stop handling</b>	<b>180</b>
<b>38</b>	<b>Page headers and footers</b>	<b>183</b>
<b>39</b>	<b>CSS</b>	<b>184</b>
<b>40</b>	<b>HTML meta description and author</b>	<b>184</b>
<b>41</b>	<b>Footnotes</b>	<b>185</b>
41.1	Regular page footnotes . . . . .	185
41.2	Minipage footnotes . . . . .	185
41.3	Titlepage thanks . . . . .	186
41.4	Regular page footnote implementation . . . . .	186
<b>42</b>	<b>Marginpars</b>	<b>187</b>

<b>43</b>	<b>Splitting HTML files</b>	<b>188</b>
<b>44</b>	<b>Sectioning</b>	<b>193</b>
44.1	Book class commands . . . . .	193
44.2	Sectioning support macros . . . . .	194
44.3	\section and friends . . . . .	199
<b>45</b>	<b>Starting a new file</b>	<b>201</b>
<b>46</b>	<b>Starting HTML output</b>	<b>204</b>
<b>47</b>	<b>Ending HTML output</b>	<b>207</b>
<b>48</b>	<b>Titles and the titling package</b>	<b>208</b>
48.1	Setting the title, etc. . . . .	209
48.2	Changes to HTML titlepage and titlingpage . . . . .	212
48.3	Printing the title, etc. in HTML . . . . .	213
48.4	Printing the title, etc. in print form . . . . .	216
48.5	\maketitle for print output . . . . .	216
48.6	\maketitle for HTML output . . . . .	219
<b>49</b>	<b>Abstract</b>	<b>222</b>
<b>50</b>	<b>Quote and verse</b>	<b>223</b>
50.1	Citations and attributions . . . . .	223
50.2	Quotes, quotations . . . . .	223
50.3	Verse . . . . .	224
<b>51</b>	<b>Verbatim</b>	<b>225</b>
<b>52</b>	<b>Fancyvrb</b>	<b>225</b>
<b>53</b>	<b>Theorems</b>	<b>231</b>
<b>54</b>	<b>Lists</b>	<b>232</b>
54.1	Itemize . . . . .	232
54.2	Enumerate . . . . .	233
54.3	Description . . . . .	234
<b>55</b>	<b>Tabular</b>	<b>235</b>
55.1	Token lookahead . . . . .	236
55.2	Booleans . . . . .	237
55.3	Handling & and ! . . . . .	238
55.4	Handling \\ . . . . .	240
55.5	Variables . . . . .	241
55.6	Parsing @, >, <, ! columns . . . . .	242
55.7	Parsing 'l', 'c', or 'r' columns . . . . .	244
55.8	Parsing 'p', 'm', or 'b' columns . . . . .	245

55.9	Parsing ‘D’ columns . . . . .	245
55.10	Parsing the column specifications . . . . .	245
55.11	Starting a new row . . . . .	249
55.12	Data opening tag . . . . .	249
55.13	Midrules . . . . .	251
55.14	Multicolumns . . . . .	252
55.14.1	Parsing multicolumns . . . . .	252
55.14.2	High-level multicolumn interface . . . . .	255
55.14.3	Longtable captions . . . . .	255
55.14.4	\tabledatamulticolumntag . . . . .	258
55.15	Multirow . . . . .	258
55.16	Utility macros inside a table . . . . .	259
55.17	Checking for a new table cell . . . . .	259
55.18	\mrowcell . . . . .	260
55.19	New \tabular definition . . . . .	261
55.20	Array . . . . .	263
<b>56</b>	<b>Cross-references</b>	<b>263</b>
56.1	Setup . . . . .	263
56.2	Zref setup . . . . .	265
56.3	Labels . . . . .	266
56.4	References . . . . .	268
56.5	Hyper-references . . . . .	270
<b>57</b>	<b>Floats</b>	<b>273</b>
57.1	Float captions . . . . .	274
57.1.1	Caption inside a float environment . . . . .	275
57.1.2	Caption and LOF linking and tracking . . . . .	276
<b>58</b>	<b>Table of Contents, LOF, LOT</b>	<b>279</b>
58.1	Reading and printing the TOC . . . . .	279
58.2	High-level TOC commands . . . . .	282
58.3	Side TOC . . . . .	282
58.4	Low-level TOC line formatting . . . . .	283
<b>59</b>	<b>Index and glossary</b>	<b>286</b>
<b>60</b>	<b>Math</b>	<b>288</b>
60.1	Inline and display math . . . . .	290
60.2	MathJax support . . . . .	292
60.3	Equation environment . . . . .	295
60.4	AMS Math environments . . . . .	296
60.4.1	Support macros . . . . .	296
60.4.2	Environment patches . . . . .	298
<b>61</b>	<b>Lateximages</b>	<b>302</b>

<b>62</b>	<b>center, flushleft, flushright</b>	<b>308</b>
<b>63</b>	<b>Siunitx</b>	<b>309</b>
<b>64</b>	<b>Graphics</b>	<b>309</b>
64.1	<code>\graphicspath</code> . . . . .	310
64.2	Length conversions and graphics options . . . . .	310
64.3	<code>\includegraphics</code> . . . . .	313
64.4	<code>\rotatebox</code> , <code>\scalebox</code> , <code>\reflectbox</code> . . . . .	317
64.5	Null functions . . . . .	320
<b>65</b>	<b>Cleverref</b>	<b>320</b>
<b>66</b>	<b>Picture</b>	<b>321</b>
<b>67</b>	<b>Boxes and Minipages</b>	<b>322</b>
67.1	Counters and lengths . . . . .	323
67.2	Footnote handling . . . . .	323
67.3	Minipage handling . . . . .	323
67.4	Parbox, makebox, framebox, fbox, raisebox . . . . .	327
<b>68</b>	<b>Direct formatting</b>	<b>328</b>
<b>69</b>	<b>Skips, spaces, font sizes</b>	<b>331</b>
<b>70</b>	<b><code>\phantomsection</code></b>	<b>338</b>
<b>71</b>	<b><code>\LaTeX</code> and other logos</b>	<b>338</b>
71.1	HTML logos . . . . .	338
71.2	Print logos . . . . .	340
<b>72</b>	<b><code>\AtBeginDocument</code>, <code>\AtEndDocument</code></b>	<b>340</b>
<b>73</b>	<b>Trademarks</b>	<b>341</b>
<b>2</b>	<b>lwarp-abstract.sty</b>	<b>342</b>
<b>74</b>	<b>Abstract</b>	<b>342</b>
<b>3</b>	<b>lwarp-afterpage.sty</b>	<b>344</b>
<b>75</b>	<b>Afterpage</b>	<b>344</b>



---

<b>4</b>	<b>lwarp-algorithmicx.sty</b>	<b>345</b>
76	Algorithmicx	345
<b>5</b>	<b>lwarp-alltt.sty</b>	<b>345</b>
77	Alltt	346
<b>6</b>	<b>lwarp-amsthm.sty</b>	<b>347</b>
78	AMSthm	347
<b>7</b>	<b>lwarp-bookmark.sty</b>	<b>349</b>
79	Bookmark	349
<b>8</b>	<b>lwarp-booktabs.sty</b>	<b>351</b>
80	Booktabs	351
<b>9</b>	<b>lwarp-ccaption.sty</b>	<b>351</b>
81	Ccaption	351
<b>10</b>	<b>lwarp-changepage.sty</b>	<b>352</b>
82	Changepage	352
<b>11</b>	<b>lwarp-cutwin.sty</b>	<b>353</b>
83	Cutwin	353
<b>12</b>	<b>lwarp-dcolumn.sty</b>	<b>353</b>
84	Dcolumn	354
<b>13</b>	<b>lwarp-draftwatermark.sty</b>	<b>355</b>
85	Draftwatermark	355

---

<b>14</b>	<b>lwarp-ellipsis.sty</b>	<b>355</b>
86	Ellipsis	355
<b>15</b>	<b>lwarp-emptypage.sty</b>	<b>356</b>
87	Emtypage	356
<b>16</b>	<b>lwarp-endnotes.sty</b>	<b>357</b>
88	Endnotes	357
<b>17</b>	<b>lwarp-enumerate.sty</b>	<b>358</b>
89	Enumerate	358
<b>18</b>	<b>lwarp-epigraph.sty</b>	<b>359</b>
90	Epigraph	359
<b>19</b>	<b>lwarp-eso-pic.sty</b>	<b>360</b>
91	Eso-pic	360
<b>20</b>	<b>lwarp-everypage.sty</b>	<b>361</b>
92	Everypage	361
<b>21</b>	<b>lwarp-extramarks.sty</b>	<b>361</b>
93	Extramarks	361
<b>22</b>	<b>lwarp-fancyhdr.sty</b>	<b>362</b>
94	Fancyhdr	362
<b>23</b>	<b>lwarp-float.sty</b>	<b>363</b>
95	Float and \newfloat	363

---

<b>24</b>	<b>lwarp-floatflt.sty</b>	<b>365</b>
96	Floatflt	365
<b>25</b>	<b>lwarp-floatrow.sty</b>	<b>366</b>
97	Floatrow	366
<b>26</b>	<b>lwarp-fontenc.sty</b>	<b>372</b>
98	Fontenc	372
<b>27</b>	<b>lwarp-fontspec.sty</b>	<b>372</b>
99	Fontspec	372
<b>28</b>	<b>lwarp-footmisc.sty</b>	<b>372</b>
100	Footmisc	372
<b>29</b>	<b>lwarp-footnote.sty</b>	<b>374</b>
101	Footnote	374
<b>30</b>	<b>lwarp-footnotehyper.sty</b>	<b>375</b>
102	Footnotehyper	375
<b>31</b>	<b>lwarp-framed.sty</b>	<b>376</b>
103	Framed	376
<b>32</b>	<b>lwarp-ftnright.sty</b>	<b>379</b>
104	Ftnright	379
<b>33</b>	<b>lwarp-geometry.sty</b>	<b>379</b>
105	Geometry	379

---

<b>34</b>	<b>lwarp-glossaries.sty</b>	<b>380</b>
106	Glossaries	380
<b>35</b>	<b>lwarp-graphics.sty</b>	<b>381</b>
107	Graphics	381
<b>36</b>	<b>lwarp-graphicx.sty</b>	<b>381</b>
108	Graphicx	381
<b>37</b>	<b>lwarp-hyperref.sty</b>	<b>382</b>
109	Hyperref	382
<b>38</b>	<b>lwarp-indentfirst.sty</b>	<b>385</b>
110	Indentfirst	385
<b>39</b>	<b>lwarp-inputenc.sty</b>	<b>385</b>
111	Inputenc	385
<b>40</b>	<b>lwarp-keyfloat.sty</b>	<b>386</b>
112	Keyfloat	386
<b>41</b>	<b>lwarp-layout.sty</b>	<b>388</b>
113	Layout	388
<b>42</b>	<b>lwarp-letterspace.sty</b>	<b>388</b>
114	Letterspace	388
<b>43</b>	<b>lwarp-lettrine.sty</b>	<b>389</b>
115	Lettrine	389

---

<b>44</b>	<b>lwarp-lips.sty</b>	<b>390</b>
116	Lips	390
<b>45</b>	<b>lwarp-listings.sty</b>	<b>391</b>
117	Listings	391
<b>46</b>	<b>lwarp-longtable.sty</b>	<b>395</b>
118	Longtable	395
<b>47</b>	<b>lwarp-lscape.sty</b>	<b>397</b>
119	Lscape	397
<b>48</b>	<b>lwarp-ltcaption.sty</b>	<b>397</b>
120	Ltcaption	397
<b>49</b>	<b>lwarp-marginfix.sty</b>	<b>398</b>
121	Marginfix	398
<b>50</b>	<b>lwarp-marginnote.sty</b>	<b>399</b>
122	Marginnote	399
<b>51</b>	<b>lwarp-mcaption.sty</b>	<b>399</b>
123	Mcaption	399
<b>52</b>	<b>lwarp-mdframed.sty</b>	<b>400</b>
124	Mdframed	400
<b>53</b>	<b>lwarp-microtype.sty</b>	<b>407</b>
125	Microtype	407

---

<b>54</b>	<b>lwarp-mparhack.sty</b>	<b>408</b>
126	Mparhack	408
<b>55</b>	<b>lwarp-multicol.sty</b>	<b>409</b>
127	Multicol	409
<b>56</b>	<b>lwarp-multirow.sty</b>	<b>411</b>
128	Multirow	411
<b>57</b>	<b>lwarp-nameref.sty</b>	<b>411</b>
129	Nameref	411
<b>58</b>	<b>lwarp-needspace.sty</b>	<b>412</b>
130	Needspace	412
<b>59</b>	<b>lwarp-newclude.sty</b>	<b>412</b>
131	Newclude	412
<b>60</b>	<b>lwarp-newunicodechar.sty</b>	<b>413</b>
132	Newunicodechar	413
<b>61</b>	<b>lwarp-nextpage.sty</b>	<b>413</b>
133	Nextpage	413
<b>62</b>	<b>lwarp-nowidow.sty</b>	<b>413</b>
134	Nowidow	414
<b>63</b>	<b>lwarp-ntheorem.sty</b>	<b>415</b>
135	Ntheorem	415

---

<b>64</b>	<b>lwarp-pagenote.sty</b>	<b>426</b>
136	Pagenote	426
<b>65</b>	<b>lwarp-parskip.sty</b>	<b>427</b>
137	Parskip	427
<b>66</b>	<b>lwarp-placeins.sty</b>	<b>427</b>
138	Placeins	427
<b>67</b>	<b>lwarp-ragged2e.sty</b>	<b>428</b>
139	Ragged2e	428
<b>68</b>	<b>lwarp-rotating.sty</b>	<b>429</b>
140	Rotating	429
<b>69</b>	<b>lwarp-setspace.sty</b>	<b>430</b>
141	Setspace	430
<b>70</b>	<b>lwarp-showidx.sty</b>	<b>431</b>
142	Showidx	431
<b>71</b>	<b>lwarp-showkeys.sty</b>	<b>431</b>
143	Showkeys	431
<b>72</b>	<b>lwarp-sidecap.sty</b>	<b>432</b>
144	Sidecap	432
<b>73</b>	<b>lwarp-sidenotes.sty</b>	<b>433</b>
145	Sidenotes	433

---

<b>74</b>	<b>lwarp-soul.sty</b>	<b>435</b>
146	Soul	435
<b>75</b>	<b>lwarp-subfig.sty</b>	<b>437</b>
147	Subfig	437
<b>76</b>	<b>lwarp-tabularx.sty</b>	<b>443</b>
148	Tabularx	443
<b>77</b>	<b>lwarp-tabulary.sty</b>	<b>444</b>
149	Tabulary	444
<b>78</b>	<b>lwarp-textpos.sty</b>	<b>445</b>
150	Textpos	445
<b>79</b>	<b>lwarp-theorem.sty</b>	<b>446</b>
151	Theorem	446
<b>80</b>	<b>lwarp-threeparttable.sty</b>	<b>450</b>
152	Threeparttable	450
<b>81</b>	<b>lwarp-tikz.sty</b>	<b>451</b>
153	Tikz	451
<b>82</b>	<b>lwarp-titleps.sty</b>	<b>452</b>
154	Titleps	452
<b>83</b>	<b>lwarp-titlesec.sty</b>	<b>454</b>
155	Titlesec	454



<b>84</b>	<b>lwarp-titletoc.sty</b>	<b>455</b>
156	Titletoc	455
<b>85</b>	<b>lwarp-titling.sty</b>	<b>456</b>
157	Titling	456
<b>86</b>	<b>lwarp-tocloft.sty</b>	<b>457</b>
158	Tocloft	457
<b>87</b>	<b>lwarp-trivfloat.sty</b>	<b>462</b>
159	Trivfloat	462
159.1	Combining <code>\newfloat</code> , <code>\trivfloat</code> , and <code>algorithmicx</code> . . . . .	462
<b>88</b>	<b>lwarp-ulem.sty</b>	<b>464</b>
160	Ulem	464
<b>89</b>	<b>lwarp-verse.sty</b>	<b>466</b>
161	Verse	466
<b>90</b>	<b>lwarp-wallpaper.sty</b>	<b>468</b>
162	Wallpaper	468
<b>91</b>	<b>lwarp-wrapfig.sty</b>	<b>469</b>
163	Wrapfig	469
<b>92</b>	<b>lwarp-xcolor.sty</b>	<b>471</b>
164	Xcolor	471

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

---

<b>93</b>	<b>lwarp-xfrac.sty</b>	<b>474</b>
-----------	------------------------	------------

<b>165</b>	<b>Xfrac</b>	<b>474</b>
------------	--------------	------------

	<b>Change History and Index</b>	<b>477</b>
--	---------------------------------	------------

## List of Figures

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

## List of Tables

1	L <sup>A</sup> T <sub>E</sub> X–HTML generation — lwarp package — Supported functions	27
2	Additional supported packages . . . . .	31
3	Required software programs . . . . .	36
4	Files created along with the print version . . . . .	46
5	Package options . . . . .	63
6	Section depths and HTML headings . . . . .	90
7	Tabular baseline . . . . .	246
8	Tabular HTML column conversions . . . . .	246
9	Cross-referencing data structures . . . . .	264
10	Float data structures . . . . .	273

## 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 [477](#).

### v0.31:

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

### v0.30:

#### `lwarp-newproject`

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

To modify existing documents, remove from the document source:

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

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

#### HTML setup changes.

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

Old Macro	New Package Option
<code>\HomeHTMLFileName</code>	<code>HomeHTMLFilename</code>
<code>\HTMLFileName</code>	<code>HTMLFilename</code>
<code>\useLatexmk</code>	<code>latexmk</code>
<code>\warpOSwindows</code>	<code>OSWindows</code>
Old Package Option	New Package Option
<code>lwarpmklang</code> (new)	<code>IndexLanguage</code> <code>xdyFilename</code>
Old Macro	New Macro
<code>\MetaLanguage</code>	<code>\HTMLLanguage</code>
<code>\HTMLauthor</code>	<code>\HTMLAuthor</code>
<code>\NewHTMLdescription</code>	<code>\HTMLDescription</code>
<code>\SetFirstPageTop</code>	<code>\HTMLFirstPageTop</code>
<code>\SetPageTop</code>	<code>\HTMLPageTop</code>
<code>\SetPageBottom</code>	<code>\HTMLPageBottom</code>
<code>\NewCSS</code>	<code>\CSSFilename</code>

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

```

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

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

**v0.29:**

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

**v0.28:**

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

**v0.27:**

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

**v0.26:**

- Improved installation instructions for MiKTeX regarding generating the `lwarpmk` executable.
- Footnotes are now supported by `LaTeX` boxes instead of pagenotes. `pagenote` now works as per the print version. `footnote`, `footnotehyper`, `footmisc`, `endnotes`, `marginnote`, and `sidenotes` are also supported.

- L<sup>A</sup>T<sub>E</sub>X labels now are used to track the page numbers of `lateximages`. This allows the correct inclusion of `lateximages` in footnotes, pagenotes, and endnotes.
- `cutwin` and `floatflt` are also supported.

**v0.25:**

- Fix: Allows `graphicx` and `graphicsx` before `lwarp` because X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X and LuaL<sup>A</sup>T<sub>E</sub>X use `xunicode` which uses `graphics`.
- Package support for `framed`, several theorem packages, and ellipses.

**v0.24:**

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

**v0.22:**

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

**v0.21:**

- Documentation for installing on Windows has been updated and improved.
- For Windows compatibility, the `lateximages` shell script has been replaced with a `lateximages.txt` file, which is parsed by `lwarpmk` to generate `lateximages`. This does not require any changes in the user's code.
- Windows `lwarpmk` again now functions.
- For improved error handling, `lwarp` now verifies the order in which packages are loaded, and signals an error for misplaced packages. `inputenc`, `fontenc`, `newunicode`, and `fontspec` must be loaded before `lwarp`, and the other packages which `lwarp` knows about must be loaded after.
- `lwarp` no longer requires a `\title` be assigned.

**v0.20:**

- The `makefile` and related infrastructure has been replaced by the `lwarpmk` utility. This provides increased portability, reduced dependencies, and much simpler installation and setup.
- The `lwarp-newproject` package is now used to locally create support files.

- The print and HTML versions of a document may co-exist with their own sets of auxiliary files.
- Package handling is now controlled by a modular system which looks for and loads an `lwarp-<package>` version if available.
- High-level source compatibility is provided for all supported packages, almost totally eliminating the need for `warpprint` and `warpHTML` environments.
- A large number of additional packages are supported.
- A new tutorial is included in the documentation, and many obsolete sections have been removed.
- `\NewHTMLdescription` sets the HTML meta description tag for each file. See section 6.8. (v0.30 changes this to `\HTMLDescription`.)
- `\HTMLFilename` may now be empty, allowing filenames without a prefix. Lwarp no longer automatically appends a `-` character. For existing projects, add a `-` to the end of `\HTMLFilename`.
- `\HomeHTMLFilename` and `\HTMLFilename` no longer use escaped underscore characters. Underscores may be used in filenames as-is. (Version 0.30 changes these to package options `HomeHTMLFilename` and `HTMLFilename`.)
- `lwarp` now tries to auto-detect the operating system, and `\warpOSwindows` is only needed if the auto-detection fails to detect Windows. (As of v0.30, `\warpOSwindows` has been converted to the `OSWindows` option.)
- Tabular column types `@`, `>`, and `<` are now supported.
- `BlockClass` and `\InlineClass` add an optional style.
- The `sidebar` and `example` environments have been moved to the test suite, and are no longer included in `lwarp`.

#### v0.19:

- `MATHJAX` now may be used to display math via the `mathjax` option. See sections 6.2 and 6.12.5. To use MathJax with a pre-existing project, copy or link the file `lwarp_mathjax.txt` to the project's directory.
- `\rule` added, supporting `width`, `height`, `raise`, `\textcolor`.
- `\LateximageFontSizeName` provides user-adjustable font size for math and `lateximages`.
- `\minipagefullwidth` requests that the next minipage be full-width in HTML, but still the assigned width in print.
- `minipage` improved side-by-side rendering.
- CSS class `tablenotes` is provided for table note items.
- `\warpprintonly` replaces `\rowprintedonly`, and `\warpHTMLonly` is added. These behave like the `warpprint` and `warpHTML` environments, and are generally useful, so they replace the previously table-specific syntax.



- `cleveref` is loaded `\AtEndPreamble` for improved reliability. See section 65.
- `\xfracHTMLfontsize` controls `xfrac` font size in HTML.
- `Tikz` improved catcode handling.

**v0.18:**

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

**v0.17:**

- `mdframed` package is supported.

**v0.16:**

- Font and input encoding are now controlled by the user, and `lwarp` is loaded after fonts have been selected.
- Support for  $\text{X}\text{E}\text{L}\text{A}\text{T}\text{E}\text{X}$  and  $\text{L}\text{u}\text{a}\text{L}\text{A}\text{T}\text{E}\text{X}$ . See section 6.1.

## 2 Introduction

The `lwarp` project aims to allow a rich  $\text{\LaTeX}$  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 27.)

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

## 2.1 Supported packages and features

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

Table 1 lists some of the various L<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, XeL <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.
Table of Contents, Figures, Tables:	Supported, with hyperlinks.
Title page:	<code>\maketitle</code> , <code>titlepage</code> , <code>titling</code> . Optional titling-based commands for published and subtitle.
<code>abstract</code> :	Supported
Cross-references:	Emulated, with hyperlinks.
<code>hyperref</code> :	Emulated. HTML hyperlinks are generated for TOC, LOF, LOT, <code>\nameref</code> , <code>\ref</code> , the <code>cleveref</code> commands, and index entries.
Footnotes:	<code>footnote</code> , <code>footmisc</code> , <code>marginnote</code> , <code>sidenote</code> , <code>pagenote</code> , <code>endnotes</code> .
Indexing:	<code>texindy</code> is used, with hyperlinks.
Glossary:	<code>glossaries</code> and <code>xindy</code> are used.
Bibliography:	Supported, without hyperlinks so far.
Math:	Supported. Converted to SVG images with HTML ALT tags containing the $\text{\LaTeX}$ 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> .
Floats:	Appear where declared. <code>float</code> , <code>newfloat</code> , <code>caption</code> and <code>subcaption</code> , <code>subfig</code> , <code>capt-of</code> , <code>placeins</code> , <code>trivfloat</code> , <code>floatrow</code> , <code>keyfloat</code> , <code>wrapfig</code> , <code>cutwin</code> , <code>floatflt</code> .

## lwarp Supported Functions — continued

Category	Status
<code>tabular:</code>	Emulated. <code>\multirow</code> and <code>\multicolumn</code> are available, but cannot be used at the same time. Nested tables are not supported.
<code>array:</code>	Supported inside math environments, emulated elsewhere.
<code>tabularx</code> , <code>tabulary</code> , <code>threeparttable</code> , <code>multirow:</code>	Emulated.
<code>longtable:</code>	Emulated. Converted to a <code>tabular</code> . Captions supported. Extra headings and <code>\kill</code> lines must be enclosed in <code>\warpprintonly{}</code>
<code>booktabs:</code>	Emulated. <code>\toprule</code> and <code>\bottomrule</code> form black rules, <code>\midrule</code> forms silver rules, as demonstrated on this table. <code>\cmidrule</code> , demonstrated at this line, does not use width or trim options.
<code>graphics</code> , <code>graphicx:</code>	Emulated. <code>\includegraphics</code> supports <code>width</code> , <code>height</code> , <code>origin</code> , <code>angle</code> , and <code>scale</code> tags, and adds <code>class</code> . References to PDF files are changed to SVG, other image types are accepted as well. <code>\rotatebox</code> and <code>\scalebox</code> are supported as well as HTML can handle.
<code>rotating:</code>	Emulated. All objects are displayed unrotated.
<code>Lists:</code>	Supported
<code>enumitem:</code>	Supported, although spacing is still controlled by CSS.
<code>Environments:</code>	Standard L <sup>A</sup> T <sub>E</sub> X environments are supported.
<code>picture</code> and <code>tikz:</code>	Converted to an SVG image.
<code>minipage:</code>	Supported with some HTML5-imposed limitations. Nested minipages are supported. Footnotes appear at the bottom of the HTML page.
<code>fancyvrb:</code>	Supported except for verbatim footnotes.
<code>framed</code> , <code>mdframed:</code>	Supported

## lwarp Supported Functions — continued

Category	Status
multicol:	Emulated, with CSS3. Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported.
siunitx:	Supported except for <code>per-mode=fraction</code> .
xfrac:	Supported
Direct formatting:	<code>\emph</code> , <code>\textsuperscript</code> , <code>\textbf</code> , etc are supported. <code>\bfseries</code> , etc. are not yet supported. <code>lettrine</code> , <code>ulem</code> , and <code>soul</code> are supported.
Ordinals:	<code>nth</code> , <code>fmtcount</code> , and <code>engord</code> are supported.
Text ligatures:	Ligatures for symbols are supported. Ligatures for f, q, t are intentionally turned off because many simpler browsers do not display them correctly. Modern full-featured browsers re-create these ligatures on-the-fly.
Horizontal space:	HTML output for <code>thin-unbreakable</code> , <code>unbreakable</code> , <code>\enskip</code> , <code>\quad</code> , <code>\qqquad</code> , <code>\hspace</code> .
Rules:	<code>\rule</code> with width, height, raise, text color.
HTML reserved characters:	<code>\&amp;</code> , <code>\textless</code> , and <code>\textgreater</code> are converted to HTML entities.
xcolor:	<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 L <sup>A</sup> T <sub>E</sub> X package is used.
<b>Emulated:</b>	The L <sup>A</sup> T <sub>E</sub> X package is not used, but some/all of its functions are emulated. Null functions, lengths, and counters are provided for source compatibility.

Supported packages include everything listed in the table of contents, plus each of the following in table 2, and probably others which have not yet been tested. Many

Table 2: Additional supported packages

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

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

Enter ⇒ `texlua lwarpmk.lua print`

etc.

Or:

Enter ⇒ `lwarpmk html`

Enter ⇒ `lwarpmk print`

etc.

### 4.3 Installing additional utilities

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

Enter the following in a command line. If each programs' version is displayed, then that utility is already installed. See table 3 on page 36.

Enter ⇒ `luatex -version`

Enter ⇒ `xindy -version`

Enter ⇒ `latexmk -version`

Enter ⇒ `perl -version`

Enter ⇒ `pdftocrop -version`

Enter ⇒ `pdftotext -v`

Enter ⇒ `pdfseparate -version`

Enter ⇒ `pdftocairo -v`



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

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

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

Be sure to include \bin.

9. Click "Ok" when done.

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

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

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

`C:\Strawberry\perl\bin`

Be sure to include `\bin`.

4. Click "Ok" when done.

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

## 5 Tutorial

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

### 5.1 Starting a new project

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

File `tutorial.tex`

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

#### Copy from the documentation PDF:

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

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

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

Enter  $\Rightarrow$  `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 [6.12.24](#).

### 5.3 Compiling the HTML version with lwarpmk

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

1. Compile the HTML version:

Enter  $\Rightarrow$  `lwarpmk html`

- (a) `lwarpmk` uses  $\text{\LaTeX}$  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 6.12.24.



## 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` tags. 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.

 **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 12 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 pages 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 description 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 9.3.

## 6.2 lwarp package loading and options

lwarp supports book, report, and article classes.

Pkg	lwarp	Load the lwarp package immediately after the font and UTF-8 setup commands.
Opt	warpprint	Select the warpprint option to generate print output (default), or the warpHTML option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual pdflatex, etc. When lwarp is loaded in print mode, it creates <code>&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 6.12.5.
Opt	mathjax	

See table 5 for the full list of options.

Table 5: Package options

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

### 6.3 Selecting the operating system

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

### 6.4 Selecting actions for print or HTML output

The following environments and macros are used to select actions which only apply to either traditional L<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`
- Variable spaces such as `\vspace`. `\hfill` is turned into a `\quad`. Fixed spaces such as `\quad` are emulated correctly.
- Manual page positions such as the `textpos` package, which is emulated but only in a limited way.

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

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

## 6.6 Commands for a successful HTML conversion

Some commonly-used  $\text{\LaTeX}$  expressions should be modified to allow for a smooth conversion to both HTML and print-formatted outputs:

**Page references:** The printed page does not translate to the HTML page, so references to page numbers are converted to parentheses containing

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

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

**\centering, \raggedright, \raggedleft:**

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

**Superscripts and other non-math uses of math mode:**

Use `\textsuperscript{x}` instead of  $\text{\textasciitilde{x}}$

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

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

**Filenames in lists:**

filename underscore

Escape underscores in the filenames:

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

**Side-by-side minipages:**

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

**\fbox around a minipage:**

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

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

- Place the `\fbox` command and its closing brace inside `warpprint` environments.
- Use `\mdframed` instead.
- Use a custom environment to create a sidebar, containing a `BlockClass` environment with custom CSS formatting, and `\warpprintonly{\hrule}` command:

---

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

---

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

## 6.7 Title page

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

---

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

---

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

- `\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 `titledate`.

`\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.

## 6.8 HTML page meta descriptions

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

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

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

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

## 6.9 HTML page meta author

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

## 6.10 CSS

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

---

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

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

---

```

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

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

```

---

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

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

```

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

```

## 6.11 Modifying xindy index processing

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

To customize index processing:

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

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

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

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

```

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

```

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

## 6.12 Special cases and limitations

### 6.12.1 Text formatting

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

### 6.12.2 Cross-references

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

### 6.12.3 `cleveref` and `varioref` packages

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

### 6.12.4 Footnotes and page notes

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

### 6.12.5 Math

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

**SVG math option** For SVG math, math is rendered as usual by  $\text{\LaTeX}$  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  $\text{\LaTeX}$  with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML ALT tag carries the  $\text{\LaTeX}$  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}
```

---

<sup>8</sup>See section 165 regarding fonts and fractions.

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

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

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

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

**MathJax math option** The popular MathJax alternative ([mathjax.org](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	<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>
⚠ siunitx inside an equation	<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>
L <sup>A</sup> T <sub>E</sub> X macros	<ul style="list-style-type: none"> <li>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.</li> </ul>
custom MathJax macros	For an example of using custom L <sup>A</sup> T <sub>E</sub> X macros with MathJax, see page 289.

### 6.12.6 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	<code>ntheorem</code> has a bug with equation numbering in AMS environments when the option <code>thref</code> is used. <code>lwarp</code> does not share this bug, so equations with <code>\split</code> , etc, are numbered correctly with <code>lwarp</code> 's HTML output, but not with the print output. It is recommended to use <code>cleveref</code> instead of <code>ntheorem</code> 's <code>thref</code> option.

### 6.12.7 Graphics


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



**HTML class** With HTML output, `\includegraphics` accepts an optional `class=xyz` keyval combination, and if this is given then the HTML output will include that class for the image. The class is ignored for print output.

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

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

 **browser support** `\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The CSS3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike L<sup>A</sup>T<sub>E</sub>X, so expect some ugly results for scaling and rotating.

### 6.12.8 xcolor package

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

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


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

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


### 6.12.9 Tabular

**column types**

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

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

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

 **`\multirow`**

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

- table for HTML output.
- If a `\multirow` reaches to the bottom of a table, and `\bottomrule` does not go over to that edge, try adding a line of empty cells below the `\bottomrule`. This may be a browser bug.
  - If a `\midrule` is desired after the last row, an additional row of blank cells must be used.
  - Multiple paragraphs in one cell of a p, b, m column must have `\newline` between paragraphs.
  - `\cmidrule` does not support width or trim options due to CSS limitations.
  - For `longtable`, place headings and footings which do not apply to HTML inside `\warpprintonly{}`.
  - For `\toprule` and `\bottomrule`, when combined with a `warpprint` or `warppHTML` environment, if a “misplaced `\noalign`” error occurs, change
 

```
This & That \endhead
```

 to
 

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

 and likewise with the other `\end` headings. Keep the `\endfirsthead` row unchanged, as it is still relevant to HTML output.
  - For `S` columns (from the `siunitx` package), while producing print output, anything non-numeric must be placed inside `{ }` braces, including commands such as `\multirow`. While producing HTML output, though, anything placed inside braces is not seen by `lwrap`’s tabular handling algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:
 

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

#### 6.12.10 longtable package

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


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

### 6.12.11 Save Boxes

L<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.

### 6.12.12 Minipages

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

 **inline** A line of text with an inline minipage or parbox will have the minipage or parbox placed onto its own line, because a paragraph is a block element and cannot be made `inline-block`.


**side-by-side** Side-by-side minipages may be separated by `\quad`, `\qquad`, `\enskip`, `\hspace`, `\hfill`, or a `\rule`. When inside a `center` environment, the result is similar in print and HTML. Paragraph tags are suppressed between side-by-side minipages and these spacing commands, but not at the start or end of the paragraph.

**in a span** There is limited support for minipages inside an HTML `<span>`. An HTML `<div>` cannot appear inside a `<span>`. While in a `<span>`, minipages and parboxes are ignored. Use `\newline` or `\par` for an HTML break.


**size** When using `\linewidth`, `\textwidth`, and `\textheight`, widths and heights are scaled proportionally to a 6×9 inch text area.

**no-width minipages** A minipage of width exactly `\linewidth` is automatically given no HTML width.

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

 **text alignment** Nested minipages adopt their parent's text alignment in HTML, whereas in regular L<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.

### 6.12.13 mdframed package

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


```
frametitlefont=\textbf,
```

 instead of
 

```
frametitlefont=\bfseries,
```

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

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

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


### 6.12.15 caption and subcaption packages

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

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


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


### 6.12.16 subfig package

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

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

### 6.12.17 floatrow package

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

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

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

### 6.12.18 abstract package

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

### 6.12.19 verse and memoir

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

---

```

\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}


```

---


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

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

### 6.12.20 siunitx package

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

### 6.12.21 newclude package

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

*Ex:*

---

```

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

```


---

### 6.12.22 newtxmath package

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

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

### 6.12.23 babel package

Pkg `babel` If using `babel` with French, use  
 `French` `\frenchbsetup{StandardLists=true}`

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

`\CaptionSeparator` Also, when French is used, the caption separator is changed to a dash. The following may be used to restore it to a colon:

```
\renewcommand*{\CaptionSeparator}{:~}
```

### 6.12.24 glossaries package

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

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

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

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

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

### 6.12.25 enumitem package

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

### 6.12.26 enumerate package

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

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



## 7 EPUB conversion

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

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

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

Bool	<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.

---

## 8 Word-processor conversion

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

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

Bool    **FormatWordProcessor**

---

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

---

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

```
\booltrue{FormatWordProcessor}
```

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

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

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

Bool    **HTMLMarkFloats**

---

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

---

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

## 9 Modifying lwarp

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

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

An environment may be converted to a `lateximage` then displayed with an image of the resulting L<sup>A</sup>T<sub>E</sub>X output. See section 66 for an example of the `picture` environment.

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

### 9.1 Creating an lwarp version of a package

When creating HTML, `lwarp` redefines the `\usepackage` and `\RequirePackage` macros such that it first looks to see if a `lwarp-<packagename>.sty` version exists. If so, the `lwarp` version is used instead. This modular system allows users to create their own versions of packages for `lwarp` to use for HTML, simply by creating a new package with a `lwarp-` prefix. If placed in the local directory along with the source code, it will be seen by that project alone. If placed alongside the other `lwarp-` packages where T<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 135 (Ntheorem) for an example of selectively disabling user options for a package. Use this when HTML output only requires some modifications of the

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

## 9.2 Testing lwarp

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

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

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

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

## 9.3 Modifying lwarpmk

Prog `lwarpmk`  
File `lwarpmk.lua`

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

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

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

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

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

## 10 Troubleshooting

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

Also see:

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

Section 6.6: Commands for a successful HTML conversion

Section 6.12: Special cases and limitations

#### Text is not converting:

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

#### Undefined html settings:

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

#### Obscure error messages:

- Be sure that a print version of the document compiles and that your document's  $\text{\LaTeX}$  code is correct, before attempting to generate an HTML version.

#### Missing sections:

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

#### Missing html files:

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

#### Missing / incorrect cross-references:

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

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

Use  $\text{\XeLaTeX}$  or  $\text{\LuaLaTeX}$ .

**Floats out of sequence:**

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

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

**Print document contains html tags:**

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

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

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

**Images are appearing in strange places:**

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

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

**Plain-looking document:**

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

**Broken fragments of HTML:**

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

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

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

**Un-matched conditional compiles:**

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

### 10.1.1 Debug tracing output

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

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

## 10.2 Compiling the `lwarp.dtx` file

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

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

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

**Un-nested environments:**

Be sure to properly nest:

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

## 11 Implementation

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

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

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



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

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

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

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

## 12 Stack depths

Stacks are created to track depth inside the  $\text{\LaTeX}$  document structure. This depth is translated to HTML headings as shown in table 6. “Depth” here is not depth in the traditional computer-science stack-usage sense, but rather a representation of the nesting depth inside the  $\text{\LaTeX}$  document structure.

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

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

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

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

Table 6: Section depths and HTML headings

Section	$\text{\LaTeX}$ depth	HTML headings
title of the entire website		h1
none	-5	new for this package
book	-2	<b>not yet used</b>
part	-1	h2
chapter	0	h3
section	1	h4
subsection	2	h5
subsubsection	3	h6
paragraph	4	span class = "paragraph"
subparagraph	5	span class = "subparagraph"
listitem	7	new for this package, used for list items

## 13 Source Code


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

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

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

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

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

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

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

lwarp source code begins on the following page.

## 14 Detecting the T<sub>E</sub>X Engine — pdf<sub>l</sub>atex, lua<sub>l</sub>-<sub>l</sub>atex, x<sub>l</sub>atex

```

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

```

## 15 Unicode Input Characters

**for HTML & PRINT:** If using pdf<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><sub>E</sub>X, preserve upright quotes in verbatim text:

```

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

```

## 16 Early package requirements

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

## 17 Operating-System portability

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

### 17.1 Common portability code

Bool usingOSWindows Set if the OSWindows option is used.

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

### 17.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

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

### 17.3 MS-Windows

For MS-Windows:

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

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

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

```

35 \ifwindows
36 \LWR@setOSWindows
37 \fi

```

## 18 Package options

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

```

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

```

**Bool** **warpingprint**

**Bool** **warpingHTML**

**Bool** **mathjax**

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

```

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

```

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

Only process the contents if producing printed output.

```

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

```

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

Only process the contents if producing HTML output.

```

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

```

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

```

45 \includecomment{warpall}

```

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

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

```

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

```

```

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

```

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

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

```

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

```

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

```

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

```

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

```

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

```

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

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

```

68 \DeclareStringOption[\jobname]{BaseJobname}

```

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

```

69 \DeclareStringOption[english]{IndexLanguage}

```



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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

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

```

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

```

87 \excludecomment{LWR@createlwarpmk}

```

**Execute options** Execute the package options, with the defaults which have been set just above:

```

88 \ProcessKeyvalOptions*\relax

```

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

```

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

```

Defaults unless already over-ridden by the user:

```

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

```

## 19 Misplaced packages

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

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

`lwarp-<packagename>.sty`

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

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

```

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

```

```

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

```

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

```

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

```

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

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

```

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

```

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

The following packages must be loaded after lwarp:

```

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

```

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

---

```

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

```

## 20 Required packages

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

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

**for HTML & PRINT:** `202 \begin{warppall}`

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

Detects X<sub>Y</sub>TeX and Lua<sup>A</sup>TeX:

```

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

214 \end{warppall}

```

for HTML output: 215 \begin{warpHTML}

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

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

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

pdf<sub>l</sub>at<sub>e</sub>x only: Only pre-loaded if pdf<sub>l</sub>at<sub>e</sub>x 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.

```
222 \RequirePackage {microtype}
223
224 \microtypesetup{
225 protrusion=false,
226 expansion=false,
227 tracking=false,
228 kerning=false,
229 spacing=false}
230
231 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}
232
233 \fi
234
235 \end{warpHTML}
```

Pkg geometry Tactics to avoid unwanted page breaks and margin overflow:

- Uses a very long and wide page to minimize page breaks and margin overflow.
- Uses a scriptsize font.
- Uses extra space at the margin to avoid HTML tag overflow off the page.
- Forces a new PDF page before some environments.
- Forces line break between major pieces of long tags.

**for HTML output:** 234 `\begin{warpHTML}`  
235 `\RequirePackage[paperheight=190in,paperwidth=20in,%`  
236 `left=2in,right=12in,%`  
237 `top=1in,bottom=1in,%`  
238 `]{geometry}`  
239 `\@twosidefalse`  
240 `\@mparswitchfalse`  
241 `\end{warpHTML}`

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

Pkg `xparse`

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

243 `\RequirePackage{xparse}`

244 `\end{warpall}`

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

Pkg `expl3`

L<sup>A</sup>T<sub>E</sub>X3 programming

246 `\RequirePackage{expl3}`

Pkg `getttitlestring`

Used to emulate `\nameref`.

247 `\RequirePackage{getttitlestring}`

Pkg `everyhook`

`everyhook` is used to patch paragraph handling.

248 `\RequirePackage{everyhook}`

249 `\end{warpHTML}`

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

Pkg `fancyvrb`

Used for Verbatim, verse.

251 `\RequirePackage{fancyvrb}`

252 `\end{warpall}`

**for HTML output:** 253 \begin{warpHTML}

Pkg xifthen

254 \RequirePackage{xifthen}

Pkg xstring

255 \RequirePackage{xstring}

Pkg makeidx

256 \RequirePackage{makeidx}

257 \makeindex

Pkg calc

258 \RequirePackage{calc}

Pkg refcount

259 \RequirePackage{refcount}

Pkg newfloat

260 \RequirePackage{newfloat}

Pkg caption

261 \RequirePackage{caption}

Pkg enumitem

enumitem is patched to support \newlist with HTML.

262 \RequirePackage{enumitem}

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

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

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

266 \end{warpHTML}

**for HTML & PRINT:** 267 \begin{warpall}

Pkg titling

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

268 \RequirePackage{titling}



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

for HTML output: 

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

Pkg **zref**

Used for cross-references.

```
271 \RequirePackage{zref}
```

Pkg **amsmath**

Equation numbers are placed to the left for HTML.

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

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

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

Pkg **environ**

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

```
274 \RequirePackage{environ}
```

Pkg **titleps**

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

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

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

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

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

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

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

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

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

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

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

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

Pkg `xfrac`

Patched for HTML use. See section 165.

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

Used to convert lengths for image width/height options.

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

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

## 21 Loading packages

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

Remember the original `\RequirePackage`:

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

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

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

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

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

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

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

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

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

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

See if the package name was found:

```
293 \IfStrEq{\LWR@strresult}{}  
294 {}% no filename  
295 {}% yes filename
```

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

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

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

For each of many package names in a comma-separated list, if an `lwarp` version of a package exists, select it instead of the L<sup>A</sup>T<sub>E</sub>X version.

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

Redirect up to nine names:

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

`\RequirePackage` depending on the options and version:

```
317 \IfValueTF{#1}  
318 {}% options given  
319 \IfValueTF{#3}% version given?
```

```

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

```

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

Uses the original package, including options.

```

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

```

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

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

```

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

350 \end{warpHTML}

```

## 22 Copying a file

for HTML output: 351 \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.

```

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

368 \end{warpHTML}

```

## 23 Debugging messages

369 \begin{warppall}

Bool LWR@tracinglwarp True if tracing is turned on.

370 \newbool{LWR@tracinglwarp}

\tracinglwarp Turns on the debug tracing messages.

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

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

```

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

```

```

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

```

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

```

380 \newbool{HTMLDebugComments}
381 \boolfalse{HTMLDebugComments}

```

## 24 HTML-conversion output modifications

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

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

```

382 \newbool{FormatEPUB}
383 \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.

```

384 \newbool{FormatWordProcessor}
385 \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.)

```

386 \newbool{HTMLMarkFloats}
387 \booltrue{HTMLMarkFloats}

```

```

388 \end{warpall}

```

## 25 Remembering original formatting macros

for HTML output: 389 \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.

```

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

```

## 26 Configuration Files

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

### 26.1 project\_html.tex

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

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

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

### 26.2 lwarpmk.conf

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

An example `lwarpmk.conf`:

---

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

---

for PRINT output:

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



```

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

```

### 26.3 project.lwarpmkconf

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

```

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

```

```

490 }
491 \immediate\write\LWR@file{htmlfilename = "\HTMLFilename"}
492 \immediate\write\LWR@file{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
493 \immediate\write\LWR@file{language = "\LWR@IndexLanguage"}
494 \immediate\write\LWR@file{xdyfile = "\LWR@xdyFilename"}
495 \immediate\closeout\LWR@file
496 \end{warpprint}

```

## 26.4 lwarp.css

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

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

```

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

```

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

```
579
580 span.textsf {
581     font-family: "DejaVu Sans", "Bitstream Vera Sans",
582         Geneva, Verdana, sans-serif ;
583 }
584
585 span.attribution {
586     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
587 }
588
589 span.citetitle {
590     margin-left: 1em ; font-size: 80% ; font-style: oblique;
591 }
592
593 span.poemtitle {
594     font-size: 120% ; font-weight: bold;
595 }
596
597 blockquote {
598     margin-left: 0px ;
599     margin-right: 0px ;
600 }
601
602 blockquote p {
603     line-height: 1.5;
604     text-align: left ;
605     font-size: .85em ;
606     margin-left: 3em ;
607     margin-right: 3em ;
608 }
609
610 blockquotation {
611     margin-left: 0px ;
612     margin-right: 0px ;
613 }
614
615 blockquotation p {
616     line-height: 1.5;
617     text-align: left ;
618     font-size: .85em ;
619     margin-left: 3em ;
620     margin-right: 3em ;
621 }
622
623 div.epigraph {
624     line-height: 1.2;
625     text-align: left ;
626     padding: 3ex 1em 0ex 1em ;
627 /*     margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
628     margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
```

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

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

```
729 }
730 h2 { font-size: 2.25em }
731 h3 { font-size: 2em }
732 h4 { font-size: 1.75em }
733 h5 { font-size: 1.5em }
734 h6 { font-size: 1.25em }
735 span.paragraph {font-size: 1em ; font-variant: normal ;
736     margin-right: 1em ; }
737 span.subparagraph {font-size: 1em ; font-variant: normal ;
738     margin-right: 1em ; }
739
740
741
742 /* Title of the file */
743 h1 {
744     margin: 0ex 0em 0ex 0em ;
745     line-height: 1.3;
746     text-align: center ;
747 }
748
749 /* Part */
750 h2 {
751     margin: 1ex 0em 1ex 0em ;
752     line-height: 1.3;
753     text-align: center ;
754 }
755
756 /* Chapter */
757 h3 {
758     margin: 3ex 0em 1ex 0em ;
759     line-height: 1.3;
760 }
761
762 /* Section */
763 h4 {
764     margin: 3ex 0em 1ex 0em ;
765     line-height: 1.3;
766 }
767
768 /* Sub-Section */
769 h5 {
770     margin: 3ex 0em 1ex 0em ;
771     line-height: 1.3;
772 }
773
774 /* Sub-Sub-Section */
775 h6 {
776     margin: 3ex 0em 1ex 0em ;
777     line-height: 1.3;
778 }
```

```
779
780
781 div.titlepage {
782   text-align: center ;
783 }
784
785 .footnotes {
786   font-size: .85em ;
787   margin: 3ex 1em 0ex 1em ;
788   padding-bottom: 1ex ;
789   border-top: 1px solid silver ;
790 }
791
792 .marginpar {
793   max-width:50%;
794   float:right;
795   text-align:left;
796   margin: 1ex 0.5em 1ex 1em ;
797   padding: 1ex 0.5em 1ex 0.5em ;
798   font-size: 85% ;
799   border-top: 1px solid silver ;
800   border-bottom: 1px solid silver ;
801   overflow-x: auto;
802 }
803
804 .marginpar br { margin-bottom: 2ex ; }
805
806 div.marginblock {
807   max-width:50%;
808   float:right;
809   text-align:left;
810   margin: 1ex 0.5em 1ex 1em ;
811   padding: 1ex 0.5em 1ex 0.5em ;
812   overflow-x: auto;
813 }
814
815 div.marginblock div.minipage {
816   display: block ;
817   margin: 0pt auto 0pt auto ;
818 }
819
820 div.marginblock div.minipage p { font-size: 85%}
821
822 div.marginblock br { margin-bottom: 2ex ; }
823
824
825 section.textbody div.footnotes{
826   margin: 3ex 0em 0ex 0em ;
827   border-bottom: 2px solid silver ;
828 }
```



```
829
830 .footnoteheader {
831     border-top: 2px solid silver ;
832     margin-top: 3ex ;
833     padding-top: 1ex ;
834     font-weight: bold ;
835 }
836
837 .mpfootnotes {
838     text-align: left ;
839     font-size: .85em ;
840     margin-left: 1em ;
841     border-top: 1px solid silver ;
842 }
843
844 /* Remove footnote top border in the title page. */
845 div.titlepage div.mpfootnotes {
846     border-top: none ;
847 }
848
849
850
851 ol {
852     margin: 1ex 1em 1ex 0em;
853     line-height: 1.2;
854 }
855
856 ul, body dir, body menu {
857     margin: 1ex 1em 1ex 0em;
858     line-height: 1.2;
859 }
860
861 li { margin: 0ex 0em 1ex 0em; }
862
863 html {
864     margin: 0;
865     padding: 0;
866 }
867
868 .programlisting {
869     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
870         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
871         "Courier New", monospace;
872     margin: 1ex 0ex 1ex 0ex ;
873     padding: .5ex 0pt .5ex 0pt ;
874     overflow-x: auto;
875 }
876
877 section.textbody>pre.programlisting {
878     border-top: 1px solid silver ;
```

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

```
929     padding: .5ex 0pt .5ex 0pt ;
930     overflow-x: auto ;
931 }
932
933 .lverbatim {
934     margin: 1ex 0pt 1ex 0pt ;
935     padding: .5ex 0pt .5ex 0pt ;
936     overflow-x: auto ;
937 }
938
939 .fancyvrb {
940     font-size:.85em ;
941     margin: 3ex 0pt 3ex 0pt
942 }
943
944 .fancyvrblabel {
945     font-weight:bold;
946     text-align: center ;
947 }
948
949
950 .verse {
951     font-family: "Linux Libertine Mono O", "Lucida Console",
952                 "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
953                 "Liberation Mono", "FreeMono", "Andale Mono",
954                 "Nimbus Mono L", "Courier New", monospace;
955     margin-left: 1em ;
956 }
957
958
959 div.singlespace { line-height: 1.2 ; }
960 div.onehalfspace { line-height: 1.5 ; }
961 div.doublespace { line-height: 2 ; }
962
963
964
965
966
967 /* Minipage environments, vertically aligned to top, center, bottom: */
968 .minipage {
969     /* display: inline-block ; */
970     /* Mini pages which follow each other will be tiled. */
971     margin: .25em .25em .25em .25em;
972     padding: .25em .25em .25em .25em;
973     display: inline-flex;
974     flex-direction: column ;
975     overflow: auto;
976 }
977
978 /* Paragraphs in the flexbox did not collapse their margins. */
```

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

```
1029 }
1030
1031
1032 /* framed package */
1033 .framed {
1034     margin: 3ex 0em 3ex 0em ;
1035     border: 1px solid black;
1036     border-radius: 0px ;
1037     padding: .3ex 1em 0ex 1em ;
1038     display: block ;
1039 }
1040
1041 .snugframed {
1042     margin: 3ex 0em 3ex 0em ;
1043     border: 1px solid black;
1044     border-radius: 0px ;
1045     display: block ;
1046 }
1047
1048 .framedleftbar {
1049     margin: 3ex 0em 3ex 0em ;
1050     border-left: 3pt solid black;
1051     border-radius: 0px ;
1052     padding: .3ex .2em .3ex 1em ;
1053     display: block ;
1054 }
1055
1056 .framedtitle {
1057 margin: 0em ;
1058 padding: 0em ;
1059     font-size: 130%
1060 }
1061
1062 .framedtitle p { padding: .3em }
1063
1064
1065
1066 dl {
1067     margin: 1ex 2em 1ex 0em;
1068     line-height: 1.3;
1069 }
1070
1071 dl dt {
1072     margin-top: 1ex;
1073     font-weight: bold;
1074 }
1075
1076 dl dd p { margin-top: 0em; }
1077
1078
```

```
1079 nav.toc, nav.lof, nav.lot, nav.lol, nav.lothm {
1080     font-family: "URW Classico", Optima, "Linux Biolinum O",
1081         "DejaVu Sans", "Bitstream Vera Sans",
1082         Geneva, Verdana, sans-serif ;
1083     margin-bottom: 4ex ;
1084 }
1085
1086 nav.toc p, nav.lof p, nav.lot p, nav.lol p, nav.lothm p {
1087     line-height: 1.2 ;
1088     margin-top:.5ex ;
1089     margin-bottom:.5ex;
1090     font-size: .9em ;
1091 }
1092
1093
1094
1095 img, img.hyperimage, img.borderimage {
1096     max-width: 600px;
1097     border: 1px solid silver;
1098     box-shadow: 3px 3px 3px #808080 ;
1099     padding: .5% ;
1100     margin: .5% ;
1101     background: none ;
1102 }
1103
1104 img.inlineimage{
1105     padding: 0px ;
1106     box-shadow: none ;
1107     border: none ;
1108     background: none ;
1109     margin: 0px ;
1110     display: inline-block ;
1111     border-radius: 0px ;
1112 }
1113
1114 img.logoimage{
1115     max-width: 300px ;
1116     box-shadow: 3px 3px 3px #808080 ;
1117     border: 1px solid black ;
1118     background:none ;
1119     padding:0 ;
1120     margin:.5ex ;
1121     border-radius: 10px ;
1122 }
1123
1124
1125 .section {
1126 /*
1127     To have each section float relative to each other:
1128 */
```

```
1129 /*
1130     display: block ;
1131     float: left ;
1132     position: relative ;
1133     background: white ;
1134     border: 1px solid silver ;
1135     padding: .5em ;
1136 */
1137     margin: 0ex .5em 0ex .5em ;
1138     padding: 0 ;
1139 }
1140
1141
1142 figure {
1143     margin: 3ex auto 3ex auto ;
1144     padding: 1ex 1em 1ex 1em ;
1145     overflow-x: auto ;
1146 }
1147
1148
1149 /* To automatically center images in figures: */
1150 /*
1151 figure img.inlineimage {
1152     margin: 0ex auto 0ex auto ;
1153     display: block ;
1154 }
1155 */
1156
1157 /* To automatically center minipages in figures: */
1158 /*
1159 figure div.minipage, figure div.minipage div.minipage {
1160     margin: 1ex auto 1ex auto ;
1161     display: block ;
1162 }
1163 */
1164
1165 figure div.minipage p { font-size: 85% ; }
1166
1167 figure.subfigure, figure.subtable {
1168     display: inline-block ; margin: 3ex 1em 3ex 1em ;
1169 }
1170
1171 figcaption .minipage { margin:0 ; padding: 0 }
1172
1173 div.floatrow { text-align: center; }
1174
1175 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
1176
1177 div.floatfoot { font-size: .85em ;
1178     border-top: 1px solid silver ; line-height: 1.2 ; }
```

```

1179
1180 figcaption , .lstlistingtitle {
1181     font-size: .85em ;
1182     text-align: center ;
1183     font-weight: bold ;
1184 margin-top: 1ex ;
1185 margin-bottom: 1ex ;
1186 }
1187
1188 figure.subfigure figcaption, figure.subtable figcaption {
1189     border-bottom: none ; background: none ;
1190 }
1191
1192 div.nonfloatcaption {
1193     margin: 1ex auto 1ex auto ;
1194     font-size: .85em ;
1195     text-align: center ;
1196     font-weight: bold ;
1197 }
1198
1199 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
1200 figure div.floatrow div.minipage figcaption {
1201 border: none ;
1202 background: none ;
1203 }
1204
1205
1206 table {
1207     margin: 1ex auto 1ex auto ;
1208     border-collapse: collapse ;
1209     border-spacing: 0px ;
1210     line-height: 1.3 ;
1211 }
1212
1213 tr.hline {border-top: 1px solid silver ; margin-top: 0ex ;
1214     margin-bottom: 0ex ; } /* for \hline */
1215
1216 tr.tbrule {border-top: 1px solid black ; margin-top: 0ex ;
1217     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
1218
1219 td {padding: 1ex .5em 1ex .5em ;}
1220
1221 table td.tdl { text-align: left ; vertical-align: middle ; }
1222 table td.tdc { text-align: center ; vertical-align: middle ; }
1223 table td.tdr { text-align: right ; vertical-align: middle ; }
1224 table td.tdp { text-align: left ; vertical-align: bottom ; }
1225 table td.tdm { text-align: left ; vertical-align: middle ; }
1226 table td.tdb { text-align: left ; vertical-align: top ; }
1227 table td.tdP { text-align: center ; vertical-align: bottom ; }
1228 table td.tdM { text-align: center ; vertical-align: middle ; }

```



```

1229 table td.tdB { text-align: center ; vertical-align: top ; }
1230 table td.tdlrule { text-align: left ; border-top: 1px solid silver ;
1231     vertical-align: middle ; } /* for cmidrule */
1232 table td.tdcrule { text-align: center ; border-top: 1px solid silver ;
1233     vertical-align: middle ; }
1234 table td.tdrrule { text-align: right ; border-top: 1px solid silver ;
1235     vertical-align: middle ; }
1236 table td.tdprule { text-align: left ; border-top: 1px solid silver ;
1237     vertical-align: bottom ; }
1238 table td.tdmrule { text-align: left ; border-top: 1px solid silver ;
1239     vertical-align: middle ; }
1240 table td.tdbrule { text-align: left ; border-top: 1px solid silver ;
1241     vertical-align: top ; }
1242 table td.tdPrule { text-align: center ; border-top: 1px solid silver ;
1243     vertical-align: bottom ; }
1244 table td.tdMrule { text-align: center ; border-top: 1px solid silver ;
1245     vertical-align: middle ; }
1246 table td.tdBrule { text-align: center ; border-top: 1px solid silver ;
1247     vertical-align: top ; }
1248
1249 /* Margins of paragraphs inside table cells: */
1250 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
1251     padding-bottom: 1ex ; margin: 0ex ; }
1252 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
1253     padding-bottom: 1ex ; margin: 0ex ; }
1254 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
1255     padding-bottom: 1ex ; margin: 0ex ; }
1256
1257 td.tdp , td.tdprule , td.tdP , td.tdPrule
1258     { padding: 0ex .5em 0ex .5em ; }
1259 td.tdm , td.tdmrule , td.tdM , td.tdMrule
1260     { padding: 0ex .5em 0ex .5em ; }
1261 td.tdb , td.tdbrule , td.tdB , td.tdBrule
1262     { padding: 0ex .5em 0ex .5em ; }
1263
1264
1265 /* table notes: */
1266 .tnotes {
1267     margin: 0ex 5% 1ex 5% ;
1268     padding: 0.5ex 1em 0.5ex 1em;
1269     font-size:.85em;
1270     text-align: left ;
1271 }
1272
1273 .tnotes dl dt p {margin-bottom:0px;}
1274
1275 .tnoteitemheader {margin-right: 1em;}
1276
1277
1278

```

```
1279 /* center, flushleft, flushright environments */
1280 div.center{text-align:center;}
1281 div.center table {margin-left:auto;margin-right:auto;}
1282 div.flushleft{text-align:left;}
1283 div.flushleft table {margin-left:0em ; margin-right:auto;}
1284 div.flushright{text-align:right;}
1285 div.flushright table {margin-left:auto ; margin-right: 0em ;}
1286
1287
1288
1289
1290 /* program listing callouts: */
1291 span.callout {
1292     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1293     Geneva, Verdana, sans-serif ;
1294     border-radius: .5em;
1295     background-color:black;
1296     color:white;
1297     padding:0px .25em 0px .25em;
1298     margin: 0 ;
1299     font-weight: bold;
1300     font-size:.72em ;
1301 }
1302
1303 div.programlisting pre.verbatim span.callout{
1304 font-size: .85em ;
1305 }
1306
1307
1308
1309
1310
1311 div.published
1312 {
1313     text-align: center ;
1314     font-variant: normal ;
1315     font-style: italic ;
1316     font-size: 1em ;
1317     margin: 3ex 0em 3ex 0em ;
1318 }
1319
1320 div.subtitle
1321 {
1322     text-align: center ;
1323     font-variant: normal ;
1324     font-style: italic ;
1325     font-size: 1.25em ;
1326     margin: 3ex 0em 3ex 0em ;
1327 }
1328
```

```
1329 div.subtitle p { margin: 1ex ; }
1330
1331 div.author
1332 {
1333     font-variant: normal ;
1334     font-style: normal ;
1335     font-size: 1em ;
1336     margin: 3ex 0em 3ex 0em ;
1337 }
1338
1339 div.author table {
1340     margin: 3ex auto 0ex auto ;
1341     background: none ;
1342 }
1343
1344 div.author table tbody tr td { padding: .25ex ; }
1345
1346 span.affiliation {font-size: .85em ; font-variant: small-caps; }
1347
1348 div.titledate {
1349     text-align: center ;
1350     font-size: .85em ;
1351     font-style: italic;
1352     margin: 6ex 0em 6ex 0em ;
1353 }
1354
1355
1356 nav.topnavigation{
1357     text-align: left ;
1358     padding: 0.5ex 1em 0.5ex 1em ;
1359 /*     margin: 2ex 0em 3ex 0em ; */
1360     margin: 0 ;
1361     border-bottom: 1px solid silver ;
1362     border-top: 1px solid silver ;
1363     clear:right ;
1364 }
1365
1366 nav.botnavigation{
1367     text-align: left ;
1368     padding: 0.5ex 1em 0.5ex 1em ;
1369 /*     margin: 3ex 0em 2ex 0em ; */
1370     margin: 0 ;
1371     border-top: 1px solid silver ;
1372     border-bottom: 1px solid silver ;
1373     clear:right ;
1374 }
1375
1376
1377 header{
1378     line-height: 1.2 ;
```

```
1379     font-size: 1em ;
1380 /*     border-bottom: 2px solid silver ; */
1381     margin: 0px ;
1382     padding: 0ex 1em 0ex 1em ;
1383     text-align:center ;
1384 }
1385
1386 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
1387
1388
1389 footer{
1390     font-size: .85em ;
1391     line-height: 1.2 ;
1392     margin-top: 1ex ;
1393     border-top: 2px solid silver ;
1394     padding: 2ex 1em 2ex 1em ;
1395     clear:right ;
1396     text-align:left ;
1397 }
1398
1399
1400 a.linkhome { font-weight:bold ; font-size: 1em ;}
1401
1402
1403 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
1404
1405 img.lateximage{
1406     padding: 0px 0px 0px 0px ;
1407     box-shadow: none ;
1408     border: none ;
1409     background: none ;
1410     margin: 0px 0px -.15ex 0px ;
1411     /* pdfcrop leaves a slight margin, adjust to baseline */
1412     max-width: 100% ;
1413     border-radius: 0ex ;
1414     border: none ;
1415 }
1416
1417
1418
1419 nav.sidetoc {
1420     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1421         "Lucida Bright", Georgia, serif;
1422     float:right ;
1423     width: 20%;
1424     border-left: 1px solid silver;
1425     border-top: 1px solid silver;
1426     border-bottom: 1px solid silver;
1427 /*     border-top: 2px solid #808080 ; */
1428     background: #FAF7F4 ;
```

```
1429     padding: 2ex 0em 2ex 1em ;
1430     margin: 0ex 0em 2ex 1em ;
1431     font-size:.9em ;
1432     border-radius: 20px 0px 0px 20px ;
1433 }
1434
1435 div.sidetoccontents {
1436 /*     border-top: 1px solid silver ; */
1437     overflow-y: auto ;
1438     width: 100% ;
1439     text-align: left ;
1440 }
1441
1442 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
1443     text-indent: 0 ; }
1444 nav.sidetoc p a {color:black ; font-size: .7em ;}
1445 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
1446     border-bottom: 1px solid silver ; }
1447 nav.sidetoc a:hover {text-decoration: underline ; }
1448
1449
1450
1451 section.textbody { margin: 0ex 1em 0ex 1em ;}
1452
1453
1454 div.multicolsheading { -webkit-column-span: all;
1455     -moz-column-span: all; column-span: all; }
1456 div.multicols { -webkit-columns: 3 380px ;
1457     -moz-columns: 3 380px ; columns: 3 380px ; }
1458 div.multicols p {margin-top: 0ex}
1459
1460
1461
1462 /* Used to support algorithmicx: */
1463 span.floatright { float: right ; }
1464
1465
1466
1467
1468 /* Native LaTeX theorems: */
1469
1470 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
1471 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
1472
1473
1474 /* theorem, amsthm, and ntheorem packages */
1475
1476 span.theoremheader,
1477 span.theoremheaderplain,
1478 span.theoremheaderdefinition,
```

```
1479 span.theoremheaderbreak,
1480 span.theoremheadermarginbreak,
1481 span.theoremheaderchangebreak,
1482 span.theoremheaderchange,
1483 span.theoremheadermargin
1484 {
1485 font-style:normal ; font-weight: bold ; margin-right: 1em ;
1486 }
1487
1488 span.amsthmnameplain,
1489 span.amsthmnamedefinition,
1490 span.amsthmnumberplain,
1491 span.amsthmnumberdefinition
1492 {
1493 font-style:normal ; font-weight: bold ;
1494 }
1495
1496
1497 span.amsthmnameremark,
1498 span.amsthmnumberremark
1499 {font-style:italic ; font-weight: normal ; }
1500
1501
1502 span.amsthmnoteplain,
1503 span.amsthmnotedefinition
1504 {font-style:normal ;}
1505
1506
1507 span.theoremheaderremark,
1508 span.theoremheaderproof,
1509 span.amsthmproofname
1510 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
1511
1512 span.theoremheadersc
1513 {
1514 font-style:normal ;
1515 font-variant: small-caps ;
1516 font-weight: normal ;
1517 margin-right: 1em ;
1518 }
1519
1520 .theoremdemark {float:right}
1521
1522 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
1523 div.theorembodybreak, div.theorembodynonumberbreak,
1524 div.theorembodymarginbreak,
1525 div.theorembodychangebreak,
1526 div.theorembodychange,
1527 div.theorembodymargin
1528 {
```

```
1529 font-style:italic;
1530 margin-top: 3ex ; margin-bottom: 3ex ;
1531 }
1532
1533 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
1534 div.theorembodyplainupright, nonumberplainuprightsc,
1535 div.amsthmbodydefinition, div.amsthmbodyremark,
1536 div.amsthmproof
1537 {
1538 font-style: normal ;
1539 margin-top: 3ex ; margin-bottom: 3ex ;
1540 }
1541
1542 span.amsthmnoteremark {}
1543
1544
1545
1546 /*
1547 For CSS LaTeX and related logos:
1548 Based on:
1549 http://edward.oconnor.cx/2007/08/tex-poshlet
1550 http://nitens.org/taraborelli/texlogo
1551 */
1552
1553 .latexlogofont {
1554     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1555                 "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1556     font-variant: normal ;
1557 }
1558
1559 .latexlogo {
1560     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1561                 "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1562     letter-spacing: .03em ;
1563     font-size: 1.1em;
1564 }
1565
1566 .latexlogo sup {
1567     text-transform: uppercase;
1568     letter-spacing: .03em ;
1569     font-size: 0.85em;
1570     vertical-align: 0.15em;
1571     margin-left: -0.36em;
1572     margin-right: -0.15em;
1573 }
1574
1575 .latexlogo sub {
1576     text-transform: uppercase;
1577     vertical-align: -0.5ex;
1578     margin-left: -0.1667em;
```

```
1579 margin-right: -0.125em;
1580 font-size: 1em;
1581 }
1582
1583 .xetexlogo {
1584     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1585         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1586     letter-spacing: .03em ;
1587     font-size: 1.1em;
1588 }
1589
1590 /* A smaller gap between Xe and Tex v.s. LaTeX: */
1591 .xetexlogo sub {
1592     text-transform: uppercase;
1593     vertical-align: -0.5ex;
1594     margin-left: -0.0667em;
1595     margin-right: -0.2em;
1596     font-size: 1em;
1597     letter-spacing: .03em ;
1598 }
1599
1600 /* A large gap between Xe and LaTeX v.s. TeX: */
1601 .xelatexlogo sub {
1602     text-transform: uppercase;
1603     vertical-align: -0.5ex;
1604     margin-left: -0.0667em;
1605     margin-right: -.05em;
1606     font-size: 1em;
1607     letter-spacing: .03em ;
1608 }
1609
1610 .amslogo {
1611     font-family: "TeXGyreChorus","URW Chancery L",
1612         "Apple Chancery","ITC Zapf Chancery","Monotype Corsiva",
1613         "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
1614         "Hoefler Text", Times, "Times New Roman", serif;
1615     font-style: italic;
1616 }
1617
1618 .lyxlogo {
1619     font-family: "URW Classico", Optima, "Linux Biolinum O",
1620         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
1621     Verdana, sans-serif ;
1622 }
1623
1624
1625
1626
1627 /* Only display top and bottom navigation if a small screen: */
1628 /* Hide the sidetoc if a small screen: */
```



```
1629 nav.topnavigation { display:none; }
1630 nav.botnavigation { display:none; }
1631
1632 @media screen and (max-width: 45em) {
1633 /*      nav.sidetoc {display:none;} */
1634     nav.sidetoc {
1635         float: none ;
1636         width: 100% ;
1637         margin: 5ex 0px 5ex 0px ;
1638         padding: 0 ;
1639         border-radius: 0 ;
1640         border-bottom: 1px solid black ;
1641         border-top: 1px solid black ;
1642         box-shadow: none ;
1643     }
1644 /*      nav.topnavigation { display:block } */
1645     nav.botnavigation { display:block }
1646     .marginpar {
1647         max-width: 100%;
1648         float: none;
1649         display:block ;
1650         margin: 1ex 1em 1ex 1em ;
1651     }
1652 }
1653
1654 @media print {
1655     body {
1656         font-family: "Linux Libertine O",
1657             "DejaVu Serif", "Bitstream Vera Serif",
1658             "Liberation Serif", "Nimbus Roman No 9 L",
1659             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1660     }
1661     nav.sidetoc { display:none; }
1662     nav.topnavigation { display: none; }
1663     nav.botnavigation { display: none; }
1664 }
1665
1666 @media handheld {
1667     nav.sidetoc { display:none; }
1668     nav.topnavigation { display:block }
1669     nav.botnavigation { display:block }
1670 }
1671
1672 @media projection {
1673     nav.sidetoc { display:none; }
1674     nav.topnavigation { display:block }
1675     nav.botnavigation { display:block }
1676 }
1677 \end{VerbatimOut}
1678 % \end{Verbatim}% for syntax highlighting
```

```
1679 \end{warpprint}
```

## 26.5 lwarp\_\_sagebrush.css

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

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

```
1680 \begin{warpprint}
1681 \begin{VerbatimOut}{lwarp_sagebrush.css}
1682 @import url("lwarp.css") ;
1683
1684
1685 A:link {color:#105030 ; text-decoration: none ; }
1686 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
1687 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
1688 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
1689
1690
1691
1692 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1693 {
1694     font-family: "URW Classico", Optima, "Linux Biolinum O",
1695         "Linux Libertine O", "Liberation Serif",
1696         "Nimbus Roman No 9 L", "FreeSerif",
1697         "Hoefler Text", Times, "Times New Roman", serif;
1698     font-variant: small-caps ;
1699 font-weight: normal ;
1700     color: #304070 ;
1701     text-shadow: 2px 2px 3px #808080;
1702 }
1703
1704 h1 { /* title of the entire website, used on each page */
1705     font-variant: small-caps ;
1706     color: #304070 ;
1707     text-shadow: 2px 2px 3px #808080;
1708     background-color: #F7F7F0 ;
1709     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
1710 }
1711
1712 h1 {
1713     border-bottom: 1px solid #304070;
1714     border-top: 2px solid #304070;
1715 }
1716
1717 h2 {
```

```
1718 border-bottom: 1px solid #304070;
1719 border-top: 2px solid #304070;
1720 background-color: #F7F7F0 ;
1721 background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
1722 }
1723
1724
1725
1726 div.abstract {
1727     background: #f5f5eb ;
1728     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1729
1730     border: 1px solid silver;
1731     border-radius: 1em ;
1732 }
1733
1734 div.abstract dl {line-height:1.5;}
1735 div.abstract dt {color:#304070;}
1736
1737 div.abstracttitle{
1738     font-family: "URW Classico", Optima, "Linux Biolinum O",
1739         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1740         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1741     font-weight:bold;
1742     font-variant: small-caps ;
1743     font-size:1.5em;
1744     border-bottom: 1px solid silver ;
1745     color: #304070 ;
1746     text-align: center ;
1747     text-shadow: 1px 1px 2px #808080;
1748 }
1749
1750 span.abstracrunintitle{
1751     font-family: "URW Classico", Optima, "Linux Biolinum O",
1752         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1753         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1754     font-weight:bold;
1755 }
1756
1757
1758 div.epigraph {
1759     background: #f5f5eb ;
1760     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1761
1762     border: 1px solid silver ;
1763     border-radius: 1ex ;
1764     box-shadow: 3px 3px 3px #808080 ;
1765 }
1766
1767
```

```
1768 .example {
1769     background-color: #f5f5eb ;
1770     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1771
1772 }
1773
1774 div.exampletitle{
1775     font-family: "URW Classico", Optima, "Linux Biolinum O",
1776         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1777         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1778     font-weight:bold;
1779     font-variant: small-caps ;
1780     border-bottom: 1px solid silver ;
1781     color: #304070 ;
1782     text-align: center ;
1783     text-shadow: 1px 1px 2px #808080;
1784 }
1785
1786
1787 .sidebar {
1788     background-color: #f5f5eb ;
1789     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1790
1791 }
1792
1793 div.sidebar{
1794     font-family: "URW Classico", Optima, "Linux Biolinum O",
1795         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1796         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1797     font-weight:bold;
1798     font-variant: small-caps ;
1799     border-bottom: 1px solid silver ;
1800     color: #304070 ;
1801     text-align: center ;
1802     text-shadow: 1px 1px 2px #808080;
1803 }
1804
1805
1806 .fancyvrblabel {
1807     font-family: "URW Classico", Optima, "Linux Biolinum O",
1808         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1809         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1810     font-weight:bold;
1811     font-variant: small-caps ;
1812     font-size: 1.5em ;
1813     color: #304070 ;
1814     text-align: center ;
1815     text-shadow: 1px 1px 2px #808080;
1816 }
1817
```

```
1818
1819
1820 .minipage {
1821     background-color: #eeeeee7 ;
1822     border: 1px solid silver ;
1823     border-radius: 1ex ;
1824 }
1825
1826 .framed .minipage , .framedleftbar .minipage {
1827     border: none ;
1828     background: none ;
1829     padding: 0ex ;
1830     margin: 0ex ;
1831 }
1832
1833 figure.figure .minipage, figcaption .minipage { border: none; }
1834
1835 div.marginblock div.minipage { border: none; }
1836
1837 figure , div.marginblock {
1838     background-color: #eeeeee7 ;
1839     border: 1px solid silver ;
1840     border-radius: 1ex ;
1841     box-shadow: 3px 3px 3px #808080 ;
1842 }
1843
1844 figure figure {
1845     border: 1px solid silver ;
1846     margin: 0em ;
1847     box-shadow: none ;
1848 }
1849
1850 /*
1851 figcaption {
1852     border-top: 1px solid silver ;
1853     border-bottom: 1px solid silver ;
1854     background-color: #e8e8e8 ;
1855 }
1856 */
1857
1858
1859 div.table {
1860     box-shadow: 3px 3px 3px #808080 ;
1861 }
1862
1863 /*
1864 .tnotes {
1865     background: #e8e8e8;
1866     border: 1px solid silver;
1867 }
```

```
1868 */
1869
1870
1871 nav.topnavigation{
1872     background-color: #b0b8b0 ;
1873     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
1874 }
1875
1876 nav.botnavigation{
1877     background-color: #b0b8b0 ;
1878     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
1879 }
1880
1881
1882
1883 header{
1884     background-color: #F7F7F0 ;
1885     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
1886 }
1887
1888 footer{
1889     background-color: #F7F7F0 ;
1890     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
1891 }
1892
1893
1894
1895 nav.sidetoc {
1896     background-color: #F7F7F0 ;
1897     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
1898     box-shadow: 3px 3px 3px #808080 ;
1899     border-radius: 0px 0px 0px 20px ;
1900 }
1901
1902 div.sidetoc title {color: #304070 ; }
1903
1904 nav.sidetoc a: hover {
1905     color: #006000 ;
1906     text-decoration: none ;
1907     text-shadow: 0px 0px 2px #a0a0a0;
1908 }
1909
1910
1911 @media screen and (max-width: 45em) {
1912     nav.sidetoc { border-radius: 0 ; }
1913 }
1914
1915
1916 \end{VerbatimOut}
1917 % \end{Verbatim}% for syntax highlighting
```

```
1918 \end{warpprint}
```

## 26.6 lwarp\_formal.css

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

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

```
1919 \begin{warpprint}
1920 \begin{VerbatimOut}{lwarp_formal.css}
1921 @import url("lwarp.css") ;
1922
1923
1924
1925 A:link {color:#802020 ; text-decoration:none; }
1926 A:visited {color:#802020 ; text-shadow:none ;}
1927 A:hover {color:#400000 ; text-shadow:none ;}
1928 A:active {color:#C00000 ; text-shadow:none ;}
1929
1930
1931 body {
1932     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1933         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1934         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1935         "Times New Roman", serif;
1936     background: #fffcf5;
1937 }
1938
1939 span.textrm {
1940     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1941         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1942         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1943         "Times New Roman", serif;
1944 }
1945
1946 span.textsf {
1947     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1948         Geneva, Verdana, sans-serif ;
1949 }
1950
1951
1952
1953 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1954 {
1955     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1956         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
```

```
1957         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1958         "Times New Roman", serif;
1959     color: #800000 ;
1960     text-shadow: none ;
1961 }
1962
1963 h1, h2 {
1964     background-color: #fffcf5 ;
1965     background-image: none ;
1966     border-bottom: 1px solid #808080;
1967     border-top: 2px solid #808080;
1968 }
1969
1970 div.abstracttitle {
1971     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1972         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1973         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1974         "Times New Roman", serif;
1975     color: black ;
1976     text-shadow: none ;
1977 }
1978
1979 span.abstracrunintitle {
1980     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1981         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1982         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1983         "Times New Roman", serif;
1984     color: black ;
1985     text-shadow: none ;
1986 }
1987
1988 div.abstract { font-size: 100% }
1989
1990 .sidebar {
1991     background: #fffcf5;
1992     background-image: none ;
1993     margin: 2em 5% 2em 5%;
1994     padding: 0.5em 1em;
1995     border: none ;
1996     border-top : 1px solid silver;
1997     border-bottom : 1px solid silver;
1998     font-size: 90% ;
1999 }
2000
2001 div.sidebartitle{
2002     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2003         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2004         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2005         "Times New Roman", serif;
2006     color: #800000 ;
```



```
2007     text-shadow: none ;
2008     border: none ;
2009 }
2010
2011 .example {
2012     background: #fffcf5;
2013     background-image: none ;
2014     margin: 2em 5% 2em 5%;
2015     padding: 0.5em 1em;
2016     border: none ;
2017     border-top : 1px solid silver;
2018     border-bottom : 1px solid silver;
2019 }
2020
2021 div.exampletitle{
2022     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2023         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2024         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2025         "Times New Roman", serif;
2026     color: #800000 ;
2027     text-shadow: none ;
2028     border: none ;
2029 }
2030
2031 div.fancyvrblabel{
2032     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2033         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2034         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2035         "Times New Roman", serif;
2036     color: #800000 ;
2037     text-shadow: none ;
2038     border: none ;
2039 }
2040
2041
2042
2043 .verse {
2044     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2045         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2046         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2047         "Times New Roman", serif;
2048 }
2049
2050
2051 figure {
2052     margin: 3ex 5% 3ex 5% ;
2053     padding: 1ex 1em 1ex 1em ;
2054     background-color: #fffcf5 ;
2055     overflow-x: auto ;
2056     border: none ;
```

```
2057 /*      border-top: 1px solid silver; */
2058 /*      border-bottom: 1px solid silver; */
2059 }
2060
2061
2062 figcaption , .lstlisting {
2063     border: none ;
2064 /*      border-top: 1px solid silver ; */
2065 /*      border-bottom: 1px solid silver ; */
2066     background-color: #ffffcf5 ;
2067 }
2068
2069 .tnotes {
2070     background: #ffffcf5 ;
2071 }
2072
2073 .theorem {
2074     background: none ;
2075 }
2076
2077 .minipage {
2078     background-color: #ffffcf5 ;
2079     border: none ;
2080 }
2081
2082 div.floatrow figure { border: none ; }
2083
2084 figure figure { border: none ; }
2085
2086
2087 nav.toc, nav.lof, nav.lot, nav.lol {
2088     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2089         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2090         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2091         "Times New Roman", serif;
2092 }
2093
2094 nav.sidetoc {
2095     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2096         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2097         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2098         "Times New Roman", serif;
2099     background-image: linear-gradient(to bottom, #ffffcf5, #C0C0C0);
2100     border-radius: 0px 0px 0px 20px ;
2101 }
2102
2103 div.sidetoctitle{
2104     color: #800000 ;
2105 }
2106
```

```

2107 header{
2108     background-color: #e0e0e0 ;
2109     background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
2110     text-align:center ;
2111 }
2112
2113 footer{
2114     background-color: #e0e0e0 ;
2115     background-image: linear-gradient(to bottom, #fffcf5, #b0b0b0);
2116     padding: 2ex 1em 2ex 1em ;
2117     clear:right ;
2118     text-align:left ;
2119 }
2120
2121 nav.botnavigation {
2122     background: #dedcd5 ;
2123     border-top: 1px solid black ;
2124 }
2125 \end{VerbatimOut}
2126 % \end{Verbatim}% for syntax highlighting
2127 \end{warpprint}

```

## 26.7 sample\_project.css

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

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

```

2128 \begin{warpprint}
2129 \begin{VerbatimOut}{sample_project.css}
2130 /* ( --- Start of project.css --- ) */
2131 /* A sample project-specific CSS file for lwarp --- ) */
2132
2133 /* Load default lwarp settings: */
2134 @import url("lwarp.css") ;
2135 /* or lwarp_formal.css, lwarp_sagebrush.css */
2136
2137 /* Project-specific CSS setting follow here. */
2138 /* . . . */
2139
2140 /* ( --- End of project.css --- ) */
2141 \end{VerbatimOut}
2142 % \end{Verbatim}% for syntax highlighting
2143 \end{warpprint}

```

## 26.8 lwarp.xdy

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

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

```

2144 \begin{warpprint}
2145 \begin{VerbatimOut}{lwarp.xdy}
2146 (require "tex/inputenc/latin.xdy")
2147 (merge-rule "\\PS *" "Postscript")
2148 (require "texindy.xdy")
2149 (require "page-ranges.xdy")
2150 (require "book-order.xdy")
2151 (markup-locref :open "\hyperindexref{" :close "}")
2152 \end{VerbatimOut}
2153 % \end{Verbatim}% for syntax highlighting
2154 \end{warpprint}

```

## 26.9 lwarp\_mathjax.txt

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

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

```

2155 \begin{warpprint}
2156 \begin{VerbatimOut}{lwarp_mathjax.txt}
2157 <!-- https://groups.google.com/forum/#!topic/
2158                               mathjax-users/jUtewUcE2bY -->
2159 <script type="text/x-mathjax-config">
2160 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
2161     var seteqsectionDefault = {name: "", num: 0};
2162     var seteqsections = {}, seteqsection = seteqsectionDefault;
2163     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
2164     var AMS = MathJax.Extension["TeX/AMSmath"];
2165     TEX.Definitions.Add({
2166     macros: {
2167         seteqsection: "mySection",
2168         seteqnumber: "mySetEqNumber"
2169     }
2170     });
2171
2172     PARSE.Augment({
2173     mySection: function (name) {
2174         seteqsection.num = AMS.number;
2175         var n = this.GetArgument(name);

```

```

2176         if (n === "") {
2177             seteqsection = seteqsectionDefault;
2178         } else {
2179             if (!seteqsections["_"+n])
2180                 seteqsections["_"+n] = {name:n, num:0};
2181             seteqsection = seteqsections["_"+n];
2182         }
2183         AMS.number = seteqsection.num;
2184     },
2185     mySetEqNumber: function (name) {
2186         var n = this.GetArgument(name);
2187         if (!n || !n.match(/^ *[0-9]+ *$/))
2188             n = ""; else n = parseInt(n)-1;
2189         <!-- $ syntax highlighting -->
2190         if (n === "" || n < 1)
2191             TEX.Error
2192                 ("Argument to "+name+" should be a positive integer");
2193         AMS.number = n;
2194     }
2195 });
2196 MathJax.Hub.Config({
2197   TeX: {
2198     equationNumbers: {
2199       formatTag: function (n)
2200         {return "("+(seteqsection.name+"."+n).replace(/\./,"")+"}"},
2201       formatID: function (n) {
2202         n = (seteqsection.name+'.'+n).replace
2203           (/[:'>&]/g,"").replace(/\./,"");
2204         return 'mjax-eqn-' + n;
2205       }
2206     }
2207   }
2208 });
2209 }>
2210 </script>
2211
2212 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
2213 <script type="text/x-mathjax-config">
2214   MathJax.Ajax.config.path["Contrib"] =
2215     "https://cdn.mathjax.org/mathjax/contrib";
2216 </script>
2217
2218 <!-- https://github.com/mathjax/MathJax-third-party-extensions/
2219         tree/master/siunitx -->
2220 <script type="text/x-mathjax-config">
2221   MathJax.Hub.Config({
2222     extensions: ["tex2jax.js", "[Contrib]/siunitx/siunitx.js"],
2223     jax: ["input/TeX", "output/HTML-CSS"],
2224     tex2jax: {inlineMath: [["$", "$"], ["\\(", "\\)"]]},
2225     TeX: {extensions: ["AMSmath.js", "AMSsymbols.js", "sinuitx.js"]}

```

```

2226 });
2227 </script>
2228
2229 <script type="text/x-mathjax-config">
2230 MathJax.Hub.Config({
2231     TeX: {
2232         equationNumbers: {
2233             autoNumber: "AMS"
2234         }
2235     }
2236 });
2237 </script>
2238
2239 <!-- Alternative CDN provider: -->
2240 <script type="text/javascript" async
2241 src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.0/MathJax.js?config=TeX-AMS_HTML-full">
2242 </script>
2243
2244 <!-- No longer supported after April 30, 2017: -->
2245 <!--
2246 <script
2247     src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
2248 </script>
2249 -->
2250
2251 \end{VerbatimOut}
2252 % \end{Verbatim}% for syntax highlighting
2253 \end{warpprint}

```

## 26.10 lwarpmk option

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

```

2254 \begin{LWR@createlwarpmk}

```

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

```

2255 \begin{VerbatimOut}{lwarpmk.lua}
2256 #!/usr/bin/env texlua
2257
2258 -- Copyright 2016-2017 Brian Dunn
2259
2260 -- Print the usage of the lwarpmk command:
2261
2262 printversion = "v0.31"
2263
2264 function printhelp ()

```

```

2265 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
2266 end
2267
2268 function printusage ()
2269 print ( [[
2270
2271 lwarpmk print [project]: Compile a print version.
2272 lwarpmk printindex [project]: Process the index for the print version.
2273 lwarpmk printglossary [project]: Process the glossary for the print version.
2274 lwarpmk html [project]: Compile an HTML version.
2275 lwarpmk htmlindex [project]: Process the index for the html version.
2276 lwarpmk htmlglossary [project]: Process the glossary for the html version.
2277 lwarpmk again [project]: Touch the source code to trigger recompiles.
2278 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
2279 lwarpmk pdftohtml [project]:
2280     For use with latexmk or a Makefile:
2281     Convert project_html.pdf to project_html.html and
2282     individual HTML files.
2283 lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, .gl*
2284 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
2285 lwarpmk -h: Print this help message.
2286 lwarpmk --help: Print this help message.
2287
2288 ]] )
2289 printconf ()
2290 end
2291
2292 -- Print the format of the configuration file lwarpmk.conf:
2293
2294 function printconf ()
2295 print ( [[
2296 An example lwarpmk.conf or <project>.lwarpmkconf project file:
2297 --
2298 opsystem = "Unix"      (or "Windows")
2299 latexname = "pdflatex" (or "lualatex", or "xelatex")
2300 sourcename = "projectname" (the source-code filename w/o .tex)
2301 homehtmlfilename = "index" (or perhaps the project name)
2302 htmlfilename = "" (or "projectname" - filename prefix)
2303 latexmk = "false" (or "true" to use latexmk to build PDFs)
2304 language = "english" (use a language supported by xindy)
2305 xdyfile = "lwarp.xdy" (or a custom file based on lwarp.xdy)
2306 --
2307 Filenames must contain only letters, numbers, underscore, or dash.
2308 Values must be in "quotes".
2309
2310 ]] ) ;
2311 end
2312
2313
2314 -- Split one large sourcefile into a number of files,

```

```
2315 -- starting with destfile.
2316 -- The file is split at each occurrence of <!--|Start file|newfilename|*
2317
2318 function splitfile (destfile,sourcefile)
2319 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
2320 io.input(sourcefile)
2321 io.output(destfile)
2322 for line in io.lines() do
2323 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*|") ;
2324 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
2325 io.output(newfilename) ;
2326 else -- not a splitpoint
2327 io.write (line .. "\n") ;
2328 end
2329 end -- do
2330 end -- function
2331
2332 -- Incorrect value, so print an error and exit.
2333
2334 function cvalueerror ( line, linenum , cvalue )
2335     print ( linenum .. " : " .. line ) ;
2336     print ("lwarpmk: incorrect variable value \"" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
2337     printconf () ;
2338     os.exit(1) ;
2339 end
2340
2341 -- Load settings from the project's "lwarpmk.conf" file:
2342
2343 function loadconf ()
2344 -- Default configuration filename:
2345 local conffile = "lwarpmk.conf"
2346 -- Optional configuration filename:
2347 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
2348 -- Default language:
2349 language = "english"
2350 -- Default xdyfile:
2351 xdyfile = "lwarp.xdy"
2352 -- Verify the file exists:
2353 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
2354 print("lwarpmk: " .. conffile .. " does not exist.")
2355 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
2356 printhelp () ;
2357 os.exit(1) -- exit the entire lwarpmk script
2358 else -- file exists
2359 -- Read the file:
2360 print ("lwarpmk: Reading " .. conffile .. ".")
2361 io.input(conffile) ;
2362 -- Scan each line:
2363 local linenum = 0
2364 for line in io.lines() do -- scan lines
```



```

2365 linenum = linenum + 1
2366 i,j,cvarname,cvalue = string.find (line,"([%w-_*])%s*=%s*\"([%w%-_.]*)\"") ;
2367 -- Error if incorrect enclosing characters:
2368 if ( i == nil ) then
2369 print ( linenum .. " : " .. line ) ;
2370 print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
2371 printconf ( ) ;
2372 os.exit(1) ;
2373 end
2374 if ( cvarname == "opsystem" ) then
2375     -- Verify choice of opsystem:
2376     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
2377         opsystem = cvalue
2378     else
2379         valueerror ( line, linenum , cvalue )
2380     end
2381 elseif ( cvarname == "latexname" ) then
2382     -- Verify choice of LaTeX compiler:
2383     if (
2384         (cvalue == "pdflatex") or
2385         (cvalue == "xelatex") or
2386         (cvalue == "lualatex")
2387     ) then
2388         latexname = cvalue
2389     else
2390         valueerror ( line, linenum , cvalue )
2391     end
2392 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
2393 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
2394 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
2395 elseif ( cvarname == "latexmk" ) then latexmk = cvalue
2396 elseif ( cvarname == "language" ) then language = cvalue
2397 elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue
2398 else
2399 print ( linenum .. " : " .. line ) ;
2400 print ("lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. conffile .. ".\n" ) ;
2401 printconf ( ) ;
2402 os.exit(1) ;
2403 end
2404 end -- do scan lines
2405 end -- file exists
2406 -- Select some operating-system commands:
2407 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
2408     rmname = "rm"
2409     mvname = "mv"
2410     touchnamepre = "touch"
2411     touchnamepost = ""
2412     dirslash = "/"
2413     opquote= "\""
2414 elseif opsystem=="Windows" then -- For Windows

```

```
2415 rmname = "DEL"
2416 mvname = "MOVE"
2417 touchnamepre = "COPY /b"
2418 touchnamepost = "+,,"
2419 dirslash = "\\"
2420 opquote= "\""
2421 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
2422 end --- for Windows
2423
2424 -- set xindycmd according to pdflatex vs xelatex/lualatex:
2425 if ( latexname == "pdflatex" ) then
2426 xindycmd = "texindy -C utf8"
2427 glossarycmd = "xindy -C utf8"
2428 else
2429 xindycmd = "xindy -M texindy -C utf8"
2430 glossarycmd = "xindy -C utf8"
2431 end
2432
2433 end -- loadconf
2434
2435
2436 function refreshdate ()
2437 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
2438 end
2439
2440
2441 -- Scan the LaTeX log file for the phrase "Rerun to get",
2442 -- indicating that the file should be compiled again.
2443 -- Return true if found.
2444
2445 function reruntoget (filesorce)
2446 io.input(filesorce)
2447 for line in io.lines() do
2448 if ( string.find(line,"Rerun to get") ~= nil ) then return true end
2449 end
2450 return false
2451 end
2452
2453
2454 -- Compile one time, return true if should compile again.
2455 -- fsuffix is "" for print, "_html" for HTML output.
2456
2457 function onetime (fsuffix)
2458 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
2459 err = os.execute(
2460 --      "echo " ..
2461      latexname .. " " .. sourcename..fsuffix )
2462 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2463 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
2464 end
```

```
2465
2466
2467 -- Compile up to five times.
2468 -- fsuffix is "" for print, "_html" for HTML output
2469
2470 function manytimes (fsuffix)
2471 if onetime(fsuffix) == true then
2472 if onetime(fsuffix) == true then
2473 if onetime(fsuffix) == true then
2474 if onetime(fsuffix) == true then
2475 if onetime(fsuffix) == true then
2476 end end end end end
2477 end
2478
2479 -- Exit if the given file does not exist.
2480
2481 function verifyfileexists (filename)
2482 if (lfs.attributes ( filename , "modification" ) == nil ) then
2483 print ( "lwarpmk: " .. filename .. " not found." ) ;
2484 os.exit (1) ;
2485 end
2486 end
2487
2488
2489 -- Convert <project>_html.pdf into HTML files:
2490
2491 function pdftohtml ()
2492     -- Convert to text:
2493     print ("lwarpmk: Converting " .. sourcename
2494           .. "_html.pdf to " .. sourcename .. "_html.html")
2495     os.execute("pdftotext -enc UTF-8 -nopgbrk -layout "
2496               .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
2497     -- Split the result into individual HTML files:
2498     splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
2499 end
2500
2501
2502 -- Remove auxiliary files:
2503
2504 function removeaux ()
2505     os.execute ( rmname .. " " ..
2506                 sourcename .. ".aux " .. sourcename .. "_html.aux " ..
2507                 sourcename .. ".toc " .. sourcename .. "_html.toc " ..
2508                 sourcename .. ".lof " .. sourcename .. "_html.lof " ..
2509                 sourcename .. ".lot " .. sourcename .. "_html.lot " ..
2510                 sourcename .. ".idx " .. sourcename .. "_html.idx " ..
2511                 sourcename .. ".ind " .. sourcename .. "_html.ind " ..
2512                 sourcename .. ".log " .. sourcename .. "_html.log " ..
2513                 sourcename .. ".gl*" .. sourcename .. "_html.gl*" "
2514     )
```

```

2515 end
2516
2517
2518
2519 -- Create lateximages based on lateximages.txt:
2520 function createlateximages ()
2521 print ("lwarpmk: Creating lateximages.")
2522 io.input("lateximages.txt")
2523 -- Create the lateximages directory, ignore error if already exists
2524 err = os.execute("mkdir lateximages")
2525 -- Scan lateximages.txt
2526 for line in io.lines() do
2527 -- lwimgpage is the page number in the PDF which has the image
2528 -- lwimgnum is the sequential lateximage number to assign for the image
2529 i,j,lwimgpage,lwimgnum = string.find (line,"|(.*)|(.*)|")
2530 -- For each entry:
2531 if ( i~=nil ) then
2532 -- Separate out the image into its own single-page pdf:
2533 err = os.execute(
2534 "pdfseparate -f " .. lwimgpage .. " -l " ..
2535 lwimgpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
2536 -- Crop the image:
2537 err = os.execute(
2538 "pdfcrop -- hires lateximagetemp-" .. lwimgpage .. ".pdf lateximage-" .. lwimgnum .. ".pdf")
2539 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2540 -- Convert the image to svg:
2541 err = os.execute(
2542 "pdftocairo -svg lateximage-" .. lwimgnum .. ".pdf lateximage-" .. lwimgnum .. ".svg")
2543 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2544 -- Move the result into lateximages/:
2545 err = os.execute(
2546 mvname .. " lateximage-" .. lwimgnum .. ".svg lateximages" .. dirslash )
2547 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2548 -- Remove the temporary files:
2549 err = os.execute(
2550 rmname .. " lateximage-" .. lwimgnum .. ".pdf lateximagetemp-" .. lwimgpage .. ".pdf")
2551 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2552 end
2553 end -- do
2554 end -- function
2555
2556
2557 -- Use latexmk to compile source and index:
2558 -- fsuffix is "" for print, or "_html" for HTML
2559 function compilelatexmk ( fsuffix )
2560 -- The recorder option is required to detect changes in <project>.tex
2561 -- while we are loading <project>_html.tex.
2562 err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
2563 .. "-e "
2564 .. opquote

```

```
2565     .. "$makeindex = q/"
2566     .. xindycmd
2567     .. " -M " .. xdyfile
2568     .. " -L " .. language .. " /"
2569     .. opquote
2570     .. " -pdflatex=\"\" .. latexname .. " %O %S\" "
2571     .. sourcename..fsuffix ..".tex" ) ;
2572     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2573 end
2574
2575
2576
2577 -- lwarpmk --version :
2578
2579 if (arg[1] == "--version") then
2580 print ( "lwarpmk: " .. printversion )
2581
2582 else -- not -- version
2583
2584 -- print intro:
2585
2586 print ( "lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package." )
2587
2588 -- lwarpmk print:
2589
2590 if arg[1] == "print" then
2591 loadconf ( )
2592 if ( latexmk == "true" ) then
2593     compilelatexmk ( "" )
2594     print ( "lwarpmk: Done." )
2595 else -- not latexmk
2596     verifyfileexists ( sourcename .. ".tex" ) ;
2597     -- See if up to date:
2598     if (
2599         ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
2600         (
2601             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2602             lfs.attributes ( sourcename .. ".pdf" , "modification" )
2603         )
2604     ) then
2605         -- Recompile if not yet up to date:
2606         manytimes( "" )
2607         print ( "lwarpmk: Done." ) ;
2608     else
2609         print ( "lwarpmk: " .. sourcename .. ".pdf is up to date." ) ;
2610     end
2611 end -- not latexmk
2612
2613 -- lwarp printindex:
2614 -- Compile the index then touch the source
```

```
2615 -- to trigger a recompile of the document:
2616
2617 elseif arg[1] == "printindex" then
2618 loadconf ()
2619 print ("lwarpmk: Processing the index.")
2620 os.execute(
2621     xindycmd
2622     .. " -M " .. xdyfile
2623     .. " -L " .. language
2624     .. " " .. sourcename .. ".idx")
2625 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2626 refreshdate ()
2627 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2628 print ("lwarpmk: Done.")
2629
2630 -- lwarp printglossary:
2631 -- Compile the glossary then touch the source
2632 -- to trigger a recompile of the document:
2633
2634 elseif arg[1] == "printglossary" then
2635 loadconf ()
2636 print ("lwarpmk: Processing the glossary.")
2637
2638 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
2639     " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
2640     .. sourcename .. ".glo")
2641 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2642 refreshdate ()
2643 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2644 print ("lwarpmk: Done.")
2645
2646 -- lwarpmk html:
2647
2648 elseif arg[1] == "html" then
2649 loadconf ()
2650 if ( latexmk == "true" ) then
2651     compilelatexmk ("_html")
2652     pdftohtml ()
2653     print ("lwarpmk: Done.")
2654 else -- not latexmk
2655     verifyfileexists ( sourcename .. ".tex" ) ;
2656     -- See if exists and is up to date:
2657     if (
2658         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
2659         (
2660             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2661             lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
2662         )
2663     ) then
2664         -- Recompile if not yet up to date:
```

```
2665         manytimes("_html")
2666         pdftohtml ()
2667         print ("lwarpmk: Done.")
2668     else
2669         print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
2670     end
2671 end -- not latexmk
2672
2673 elseif arg[1] == "pdftohtml" then
2674     loadconf ()
2675     pdftohtml ()
2676
2677 -- lwarpmk htmlindex:
2678 -- Compile the index then touch the source
2679 -- to trigger a recompile of the document:
2680
2681 elseif arg[1] == "htmlindex" then
2682     loadconf ()
2683     print ("lwarpmk: Processing the index.")
2684     os.execute(
2685         xindycmd
2686         .. " -M " .. xdyfile
2687         .. " -L " .. language
2688         .. " " .. sourcename .. "_html.idx"
2689     )
2690     print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2691     refreshdate ()
2692     print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2693     print ("lwarpmk: Done.")
2694
2695 -- lwarpmk htmlglossary:
2696 -- Compile the glossary then touch the source
2697 -- to trigger a recompile of the document:
2698
2699 elseif arg[1] == "htmlglossary" then
2700     loadconf ()
2701     print ("lwarpmk: Processing the glossary.")
2702
2703     os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
2704         "_html -t " .. sourcename .. "_html.glg -o " .. sourcename ..
2705         "_html.gls " .. sourcename .. "_html.glo")
2706
2707     print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2708     refreshdate ()
2709     print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2710     print ("lwarpmk: Done.")
2711
2712 -- lwarpmk limages:
2713 -- Scan the lateximages.txt file to create lateximages,
2714 -- then touch the source to trigger a recompile.
```

```
2715
2716 elseif arg[1] == "limages" then
2717 loadconf ()
2718 print ("lwarpmk: Processing images.")
2719 createlateximages ()
2720 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2721 refreshdate ()
2722 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2723 print ("lwarpmk: Done.")
2724
2725 -- lwarpmk again:
2726 -- Touch the source to trigger a recompile.
2727
2728 elseif arg[1] == "again" then
2729 loadconf ()
2730 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2731 refreshdate ()
2732 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2733 print ("lwarpmk: Done.")
2734
2735 -- lwarpmk clean:
2736 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, .gl*
2737
2738 elseif arg[1] == "clean" then
2739 loadconf ()
2740 removeaux ()
2741 print ("lwarpmk: Done.")
2742
2743 -- lwarpmk cleanall
2744 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, .gl*
2745 -- and also project.pdf, *.html
2746
2747 elseif arg[1] == "cleanall" then
2748 loadconf ()
2749 removeaux ()
2750 os.execute ( rmname .. " " ..
2751     sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
2752     "*.html"
2753 )
2754 print ("lwarpmk: Done.")
2755
2756 -- lwarpmk with no argument :
2757
2758 elseif (arg[1] == nil) then
2759 printhelp ()
2760
2761 -- lwarpmk -h or lwarpmk --help :
2762
2763 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
2764 printusage ()
```



```

2765
2766 else
2767 print ("lwarpmk: Unknown command \"\"..arg[1]..\"\".\n")
2768 printhelp ()
2769 end
2770
2771 end -- not --version
2772 \end{VerbatimOut}
2773 % \end{Verbatim}% for syntax highlighting

2774 \end{LWR@createlwarpmk}

```

## 27 Stacks

for HTML output: 2775 \begin{warpHTML}



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

### 27.1 Assigning depths

initial depths for empty stack entries:

```
2776 \newcommand*{\LWR@depthnone}{-5}
```

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

```

2777 \newcommand*{\LWR@depthfinished}{-4}
2778 \newcommand*{\LWR@depthpart}{-1}
2779 \newcommand*{\LWR@depthchapter}{0}
2780 \newcommand*{\LWR@depthsection}{1}
2781 \newcommand*{\LWR@depthsubsection}{2}
2782 \newcommand*{\LWR@depthsubsubsection}{3}
2783 \newcommand*{\LWR@depthparagraph}{4}
2784 \newcommand*{\LWR@depthsubparagraph}{5}

```

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

```
2785 \newcommand*\LWR@depthlist}{6}
```

used by `\item`:

```
2786 \newcommand*\LWR@depthlistitem}{7}
```


## 27.2 Closing actions

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

```
2787 \newcommand*\LWR@closeone}{}% top of the stack
2788 \newcommand*\LWR@closetwo}{}
2789 \newcommand*\LWR@closethree}{}
2790 \newcommand*\LWR@closefour}{}
2791 \newcommand*\LWR@closefive}{}
2792 \newcommand*\LWR@closesix}{}
2793 \newcommand*\LWR@closeseven}{}
2794 \newcommand*\LWR@closeeight}{}
2795 \newcommand*\LWR@closenine}{}
2796 \newcommand*\LWR@closeten}{}
2797 \newcommand*\LWR@closeeleven}{}
2798 \newcommand*\LWR@closetwelve}{}%
```

## 27.3 Closing depths

A stack to record the depth of each level:

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

*Ex:*

---

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

---

```

2799 \newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
2800 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
2801 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
2802 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
2803 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
2804 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
2805 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
2806 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
2807 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
2808 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}
2809 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
2810 \newcommand*{\LWR@closedepthtwelve}{\LWR@depthnone}

```

## 27.4 Pushing and popping the stack

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

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

```

2811 \NewDocumentCommand{\pushclose}{m m}
2812 {
2813 \let\LWR@closetwelve\LWR@closeeleven
2814 \let\LWR@closeeleven\LWR@closeten
2815 \let\LWR@closeten\LWR@closenine
2816 \let\LWR@closenine\LWR@closeeight
2817 \let\LWR@closeeight\LWR@closeseven
2818 \let\LWR@closeseven\LWR@closesix
2819 \let\LWR@closesix\LWR@closefive
2820 \let\LWR@closefive\LWR@closefour
2821 \let\LWR@closefour\LWR@closethree
2822 \let\LWR@closethree\LWR@closetwo
2823 \let\LWR@closetwo\LWR@closeone
2824 \let\LWR@closeone#1
2825 \let\LWR@closedepthtwelve\LWR@closedeptheleven
2826 \let\LWR@closedeptheleven\LWR@closedepthten
2827 \let\LWR@closedepthten\LWR@closedepthnine
2828 \let\LWR@closedepthnine\LWR@closedeptheight
2829 \let\LWR@closedeptheight\LWR@closedepthseven
2830 \let\LWR@closedepthseven\LWR@closedepthsix
2831 \let\LWR@closedepthsix\LWR@closedepthfive
2832 \let\LWR@closedepthfive\LWR@closedepthfour
2833 \let\LWR@closedepthfour\LWR@closedepththree
2834 \let\LWR@closedepththree\LWR@closedepthtwo
2835 \let\LWR@closedepthtwo\LWR@closedepthone
2836 \let\LWR@closedepthone#2
2837 }

```

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

```

2838 \newcommand*{\popclose}
2839 {
2840 \let\LWR@closeone\LWR@closetwo
2841 \let\LWR@closetwo\LWR@closethree
2842 \let\LWR@closethree\LWR@closefour
2843 \let\LWR@closefour\LWR@closefive
2844 \let\LWR@closefive\LWR@closesix
2845 \let\LWR@closesix\LWR@closeseven
2846 \let\LWR@closeseven\LWR@closeeight
2847 \let\LWR@closeeight\LWR@closenine
2848 \let\LWR@closenine\LWR@closeten
2849 \let\LWR@closeten\LWR@closeeleven
2850 \let\LWR@closeeleven\LWR@closetwelve
2851 \let\LWR@closedepthone\LWR@closedepthtwo
2852 \let\LWR@closedepthtwo\LWR@closedepththree
2853 \let\LWR@closedepththree\LWR@closedepthfour
2854 \let\LWR@closedepthfour\LWR@closedepthfive
2855 \let\LWR@closedepthfive\LWR@closedepthsix
2856 \let\LWR@closedepthsix\LWR@closedepthseven
2857 \let\LWR@closedepthseven\LWR@closedeptheight
2858 \let\LWR@closedeptheight\LWR@closedepthnine
2859 \let\LWR@closedepthnine\LWR@closedepthten
2860 \let\LWR@closedepthten\LWR@closedeptheleven
2861 \let\LWR@closedeptheleven\LWR@closedepthtwelve
2862 }

2863 \end{warpHTML}

```

## 28 Data arrays

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

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

To assign an empty value:

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

for HTML output: 2864 `\begin{warpHTML}`

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

```

2865 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
2866 \expandafter\edef\csname #1#2\endcsname{\expandonce#3}%
2867 }

```

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

```
2868 \newcommand*{\LWR@getexparray}[2]{\csuse{#1#2}}
```

```
2869 \end{warpHTML}
```

## 29 HTML entities

for HTML output: 2870 \begin{warpHTML}

HTML entites and HTML Unicode entities:

```
2871 \let\LWR@origampersand\&
```

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

```
2872 \newcommand*{\HTMLentity}[1]{\LWR@origampersand#1;}
```

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

```
2873 \newcommand*{\HTMLunicode}[1]{\HTMLentity{\x#1}}
```

```
\&
```

```
2874 \renewcommand*{\&}{\HTMLentity{amp}}
```

```

\textless
\textgreater

```

```
2875 \let\LWR@origtextless\textless
```

```
2876 \renewcommand*{\textless}{\HTMLentity{lt}}
```

```
2877
```

```
2878 \let\LWR@origtextgreater\textgreater
```

```
2879 \renewcommand*{\textgreater}{\HTMLentity{gt}}
```

```
2880 \end{warpHTML}
```

## 30 HTML filename generation

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

**for HTML & PRINT:** 2881 `\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.

2882 `\providecommand*{\BaseJobname}{\jobname}`

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

2883 `\providecommand*{\HTMLFilename}{}`

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

2884 `\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.

2885 `\newcommand*{\SetHTMLFileNumber}[1]{%`  
 2886 `\setcounter{LWR@htmlfilename}{#1}%`  
 2887 `}`

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

Defaults to use section names in the filenames.

2888 `\newbool{FileSectionNames}`  
 2889 `\booltrue{FileSectionNames}`

2890 `\end{warpall}`

**for HTML output:** 2891 `\begin{warpHTML}`

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

```
2892 \newcounter{LWR@htmlfilename}
2893 \setcounter{LWR@htmlfilename}{0}
```

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

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

```
2894 \newcommand*{\LWR@htmlsectionfilename}[1]{%
2895 \LWR@traceinfo{LWR@htmlsectionfilename A}%
```

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

```
2896 \LWR@traceinfo{about to assign temp}%
2897 \edef\LWR@tempone{#1}%
2898 \LWR@traceinfo{about to compare with ??}%
2899 \ifthenelse{\equal{\LWR@tempone}{??}}{%
2900 {%
2901 \LWR@traceinfo{found ??}%
2902 }{%
2903 \LWR@traceinfo{not found ??}%
2904 }%
2905 \LWR@traceinfo{about to compare with zero or empty}%
2906 \ifthenelse{%
2907 \equal{\LWR@tempone}{0}}%
2908 \OR \equal{\LWR@tempone}{}}%
2909 \OR \equal{\LWR@tempone}{??}}%
2910 }%
2911 {%
2912 \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
2913 \HomeHTMLFilename.html%
2914 }%
```

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`:

```
2915 {%
2916 \LWR@traceinfo{LWR@htmlsectionfilename C \LWR@tempone}%
2917 \ifthenelse{%
2918 \equal{\HTMLFilename}{ } \AND \equal{\LWR@tempone}{Index} \OR \equal{\LWR@tempone}{index}}%
2919 }%
2920 {%
2921 \LWR@traceinfo{prefixing the index name with an underscore.}%
2922 \_#1.html}%
```

Otherwise, create a filename with the chosen prefix:

```
2923 {\HTMLFilename#1.html}%
2924 }%
2925 \LWR@traceinfo{\LWR@htmlsectionfilename Z}%
2926 }
```

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

Prints the filename for the given label

```
2927 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
2928 \LWR@traceinfo{\LWR@htmlrefsectionfilename A: !#1!}%
2929 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
2930 \LWR@traceinfo{\LWR@htmlrefsectionfilename B}%
2931 }

2932 \end{warpHTML}
```

## 31 Homepage link

**for HTML output:** 2933 \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.

```
2934 \newcommand*{\LinkHome}{%
2935 \LWR@subhyperrefclass{%
2936 \HomeHTMLFilename.html}%
2937 {Home}{linkhome}%
2938 }
```

`\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.

```
2939 \newcommand*{\LWR@topnavigation}{
2940 \LWR@htmlclassline{nav}{topnavigation}{\LinkHome}
2941 }
```

`\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.

```
2942 \newcommand*{\LWR@botnavigation}{
2943 \LWR@htmlclassline{nav}{botnavigation}{\LinkHome}
2944 }
```



```
2945 \end{warpHTML}
```

## 32 \PrintStack diagnostic tool



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

for HTML output: 2946 \begin{warpHTML}

`\PrintStack` Prints the closedepth stack.

```
2947 \newcommand*{\PrintStack}{
2948 \LWR@startpars
2949 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
2950 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
2951 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
2952 \LWR@closedephten{} \LWR@closedeptheleven{} \LWR@closedephtwelve{}
2953 }

2954 \end{warpHTML}
```

## 33 Closing stack levels

for HTML output: 2955 \begin{warpHTML}

Close one nested level:

```
2956 \newcommand*{\LWR@closeoneprevious}{%
2957
2958 \LWR@closeone{}
2959
2960 \popclose{}
2961 }
```

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

```
2962 \newcommand*{\LWR@closeprevious}[1]{
```

Close any pending paragraph:

```
2963 \LWR@stoppars
```

Close anything nested deeper than the desired depth:

```
2964 \whiledo{\not\(\LWR@closedepthone<#1\)}{\LWR@closeoneprevious}
2965 }

2966 \end{warpHTML}
```

## 34 Forcing a new PDF page

for HTML output: 2967 \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.

```
2968 \newcommand{\LWR@forcenewpage}{%
2969 \LWR@stoppars\LWR@orignewpage\LWR@startpars%
2970 }

2971 \end{warpHTML}
```

## 35 HTML tags, spans, divs, elements

for HTML output: 2972 \begin{warpHTML}

### 35.1 Mapping L<sup>A</sup>T<sub>E</sub>X Sections to HTML Sections

```
2973 \newcommand*{\LWR@tagpart}{h2}
2974 \newcommand*{\LWR@tagpartend}{/h2}
2975 \newcommand*{\LWR@tagchapter}{h3}
2976 \newcommand*{\LWR@tagchapterend}{/h3}
2977 \newcommand*{\LWR@tagsection}{h4}
2978 \newcommand*{\LWR@tagsectionend}{/h4}
2979 \newcommand*{\LWR@tagsubsection}{h5}
2980 \newcommand*{\LWR@tagsubsectionend}{/h5}
2981 \newcommand*{\LWR@tagsubsubsection}{h6}
2982 \newcommand*{\LWR@tagsubsubsectionend}{/h6}
2983 \newcommand*{\LWR@tagparagraph}{span class="paragraph"{} }
2984 \newcommand*{\LWR@tagparagraphend}{/span}
2985 \newcommand*{\LWR@tagsubparagraph}{span class="subparagraph"{} }
```

```

2986 \newcommand*{\LWR@tagsubparagraphend}{/span}
2987
2988 \newcommand*{\LWR@tagregularparagraph}{p}

```

## 35.2 HTML tags

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

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

```

2989
2990 \newcommand*{\LWR@htmltagc}[1]{%
2991 {%
2992 \protect\LWR@origttfamily%
2993 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
2994 }%
2995 }

```

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

```

2996 \newenvironment*\LWR@nestspan
2997 {%
2998 \addtocounter{\LWR@spandepth}{1}%
2999 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{-}{-}%
3000 }%
3001 {\addtocounter{\LWR@spandepth}{-1}}

```

`\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.

```

3002 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
3003 \LWR@ensuredoingapar%
3004 \LWR@htmltagc{#1}%
3005 \begin{\LWR@nestspan}%
3006 #2%
3007 \LWR@htmltagc{/#1}%
3008 \end{\LWR@nestspan}%
3009 }

```

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

```

3010 \NewDocumentCommand{\LWR@htmlspanclass}{m o +m}{%
3011 \LWR@ensuredoingapar%
3012 \LWR@subhtmlmlelementclass{span}{#1}{#2}%
3013 \begin{LWR@nestspan}%
3014 #3%
3015 \LWR@htmltagc{/span}%
3016 \end{LWR@nestspan}%
3017 }

```

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

Print an HTML tag: `<tag>`

```

3018 \newcommand*{\LWR@htmltagb}[1]{%
3019 \LWR@htmltagc{#1}%
3020 \endgroup%
3021 }
3022
3023 \newcommand*{\LWR@htmltag}{%
3024 \begingroup\catcode'\_ =12
3025 \LWR@htmltagb%
3026 }

```

### 35.3 Block tags and comments

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

`\LWR@htmlopencomment`

`\LWR@htmlclosecomment`

```

3027 \newcommand*{\LWR@htmlopencomment}{%
3028 {\LWR@origttfamily\LWR@origtextless}{!{-}{-}}%
3029 }
3030
3031 \newcommand*{\LWR@htmlclosecomment}{%
3032 {\LWR@origttfamily{-}{-}\LWR@origtextgreater{}}%
3033 }

```

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

```

3034 \newcommand{\LWR@htmlcomment}[1]{%
3035 \LWR@htmlopencomment}%
3036 {%

```

```

3037 \LWR@origttfamily% break ligatures
3038 #1%
3039 }%
3040 \LWR@htmlclosecomment{}}

```

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

```

3041 \newcommand{\LWR@htmlblockcommentb}[1]
3042 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
3043
3044 \newcommand{\LWR@htmlblockcomment}
3045 {%
3046 \begingroup\catcode'\_ =12%
3047 \LWR@htmlblockcommentb%
3048 }

```

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

```

3049 \newcommand*{\LWR@htmlblocktag}[1]{%
3050 \LWR@stoppars%
3051 \LWR@htmltag{#1}%
3052 \LWR@startpars%
3053 }

```

## 35.4 Div class and element class

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

Factored and reused in several places.

```

3054 \NewDocumentCommand{\LWR@subhtmlclass}{m m o}{%
3055 \IfValueTF{#3}%
3056 {% option
3057 \ifthenelse{\equal{#3}{}}{%
3058 {\LWR@htmltag{#1 class="#2"}}% empty option
3059 {\LWR@htmltag{#1 class="#2" style="#3"}}% non-empty option
3060 }% option
3061 {\LWR@htmltag{#1 class="#2"}}% no option
3062 }

```

`\LWR@htmlclass`    `{\element}` `{\class}` `[\style]`

```

3063 \NewDocumentCommand{\LWR@htmlclass}{m m o}{%
3064 \LWR@stoppars%
3065 \LWR@subhtmlclass{#1}{#2}{#3}%

```

```
3066 \LWR@startpars%
3067 }
```

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

```
3068 \newcommand*{\LWR@htmlelementclassend}[2]{%
3069 \LWR@stoppars%
3070 \LWR@htmltag{/#1}%
3071 \ifbool{HTMLDebugComments}{%
3072 \LWR@htmlcomment{End of #1 ‘‘#2’’}%
3073 }{}%
3074 \LWR@startpars%
3075 }
```

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

```
3076 \NewDocumentCommand{\LWR@htmldivclass}{m o}{%
3077 \LWR@htmlelementclass{div}{#1}[#2]%
3078 }
```

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

```
3079 \newcommand*{\LWR@htmldivclassend}[1]{%
3080 \LWR@htmlelementclassend{div}{#1}%
3081 }
```

## 35.5 Single-line elements

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

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

```
3082 \NewDocumentCommand{\LWR@htmlelementclassline}{m m o +m}{%
3083 \LWR@stoppars
3084 \LWR@subhtmlelementclass{#1}{#2}[#3]%
3085 #4%
3086 \LWR@htmltag{/#1}
3087 \LWR@startpars
3088 }
```

## 35.6 HTML5 semantic elements

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

```
3089 \newcommand*\LWR@htmlelement}[1]{%
3090 \LWR@htmlblocktag{#1}
3091 }
```

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

```
3092 \newcommand*\LWR@htmlelementend}[1]{%
3093 \LWR@stoppars
3094 \LWR@htmltag{/#1}
3095 \LWR@startpars
3096 }
3097
3098 \end{warpHTML}
```

## 35.7 High-level block and inline classes

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

For other direct-formatting commands, see section [68](#).

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

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

**for HTML output:**

```
3099 \begin{warpHTML}
3100 \NewDocumentEnvironment{BlockClass}{m o}
3101 {
3102 \LWR@htmldivclass{#1}[#2]
3103 }
3104 {
3105 \LWR@htmldivclassend{#1}
3106 }
3107 \end{warpHTML}
```

**for PRINT output:**

```
3108 \begin{warpprint}
3109 \NewDocumentEnvironment{BlockClass}{m o}{}{}
3110 \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: 3111 \begin{warpHTML}  
 3112 \newcommand{\BlockClassSingle}[2]{%  
 3113 \LWR@html@element@classline{div}{#1}{#2}%  
 3114 }  
 3115 \end{warpHTML}

for PRINT output: 3116 \begin{warpprint}  
 3117 \newcommand{\BlockClassSingle}[2]{#2}  
 3118 \end{warpprint}

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

for HTML output: 3119 \begin{warpHTML}  
 3120 \NewDocumentCommand{\InlineClass}{m o +m}{%  
 3121 \LWR@html@spanclass{#1}[#2]{#3}%  
 3122 }  
 3123 \end{warpHTML}

for PRINT output: 3124 \begin{warpprint}  
 3125 \NewDocumentCommand{\InlineClass}{m o +m}{#3}  
 3126 \end{warpprint}

## 35.8 Closing HTML tags

for HTML output: 3127 \begin{warpHTML}

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

```
3128 \newcommand*{\LWR@printclosepart}  
3129     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}  
3130 \newcommand*{\LWR@printclosechapter}  
3131     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}  
3132 \newcommand*{\LWR@printclosesection}  
3133     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}  
3134 \newcommand*{\LWR@printclosesubsection}  
3135     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}  
3136 \newcommand*{\LWR@printclosesubsubsection}  
3137     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}  
3138 \newcommand*{\LWR@printcloseparagraph}  
3139     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}  
3140 \newcommand*{\LWR@printclosesubparagraph}  
3141     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}
```

Lists require closing HTML tags:



```

3142 \newcommand*{\LWR@printcloselistitem}
3143     {\LWR@htmltag{/li}}
3144 \newcommand*{\LWR@printclosedescitem}
3145     {\LWR@htmltag{/dd}}
3146 \newcommand*{\LWR@printcloseitemize}
3147     {\LWR@htmltag{/ul}}
3148 \newcommand*{\LWR@printcloseenumerate}
3149     {\LWR@htmltag{/ol}}
3150 \newcommand*{\LWR@printclosedescription}
3151     {\LWR@htmltag{/dl}}

3152 \end{warpHTML}

```

## 36 Paragraph handling

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

Paragraph tags are or are not allowed depending on many conditions. Section 37 has high-level commands which allow paragraph-tag generation to start/stop. Even when allowed (`\LWR@doingstartpars`), tags are not generated until a  $\text{\LaTeX}$  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:** 3153 `\begin{warpHTML}`

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

```

3154 \newcounter{LWR@spandepth}
3155 \setcounter{LWR@spandepth}{0}

```

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

```

3156 \newbool{LWR@doingstartpars}
3157 \boolfalse{LWR@doingstartpars}

```

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

```

3158 \newbool{LWR@doingapar}
3159 \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:

```

3160 \newcommand*{\LWR@ensuredoingapar}{%
3161 \ifbool{\LWR@doingstartpars}%
3162 {\global\booltrue{\LWR@doingapar}}%
3163 {}%
3164 }

```

`\LWR@openparagraph`

```

3165 \newcommand*{\LWR@openparagraph}
3166 {%

```

See if paragraph handling is enabled:

```

3167 \ifbool{\LWR@doingstartpars}%
3168 {% handling pars

```

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

```

3169 \ifthenelse{%
3170 \cnttest{\value{\LWR@lateximagedepth}}{>}{0} \OR%
3171 \cnttest{\value{\LWR@spandepth}}{>}{0}%
3172 }{% 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>`.

```

3173 {}% no nested par tags

```

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

```

3174 {% yes nest par tags
3175 \LWR@htmltagc{\LWR@tagregularparagraph}%

```

Manually indent item list labels to avoid left margin intrusion:

L<sup>A</sup>T<sub>E</sub>X default list environments use `\@itemdepth` and `\@enumdepth`, but `lwarp` uses the `enumitem` package, which uses `\@listdepth`.

See if are nested inside an item list:

```

3176 \ifnumcomp{\@listdepth}{>}{0}%
3177 {%

```

If so, leave some horizontal room in the L<sup>A</sup>T<sub>E</sub>X PDF output for list labels:

```

3178 \LWR@orighspace{1in}%
3179 }{}%

```

Now have started a paragraph.

```

3180 \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.)

```

3181 \let\par\LWR@closeparagraph%
3182 }% end of yes nest par tags
3183 }% end of handling pars
3184 {}% not handling pars
3185 }

```

\LWR@closeparagraph

```

3186 \newcommand*{\LWR@closeparagraph}
3187 {%

```

See if paragraph handling is enabled:

```

3188 \ifbool{LWR@doingapar}%

```

If currently in paragraph mode:

```

3189 {% handling pars

```

See if already started a lateximage or a <span>:

```

3190 \ifthenelse{%
3191 \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3192 \cnttest{\value{LWR@spandepth}}{>}{0}%
3193 }%

```

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

```

3194 {% no nested par tags
3195 \ifthenelse{%
3196 \cnttest{\value{LWR@spandepth}}{>}{0}%
3197 \AND%
3198 \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3199 }%
3200 {%
3201 \ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}%

```

```

3202 }%
3203 {}%
3204 }% no nested par tags

```

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

```

3205 {% yes nest par tags

```

Print a closing tag:

```

3206 \unskip%
3207 \LWR@htmltagc{/\LWR@tagregularparagraph}%

```

No longer doing a paragraph:

```

3208 \global\boolfalse{LWR@doingapar}%
3209 % Disable the special \env{minipage} \& \cs{hspace} interaction
3210 % until a new minipage is found:
3211 % \begin{macrocode}
3212 \global\boolfalse{LWR@minipagethispar}%
3213 }% end of yes nest par tags
3214 }% end of handling pars

```

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

```

3215 {% not handling pars
3216 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
3217 {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
3218 {}%
3219 }% not handling pars

```

Finish with regular paragraph processing

```

3220 \LWR@origpar%
3221 }

```

```

3222 \end{warpHTML}

```

## 37 Paragraph start/stop handling

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

Section 36 has the commands which actually generate the tags.

The `everyhook` package is used to generate the opening paragraph tags. The closing tags are generated by `\par`.

for HTML output: 3223 \begin{warpHTML}

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

```
3224 \newcommand*{\LWR@startpars}%
3225 {%
```

See if currently handling HTML paragraphs:

```
3226 \ifbool{\LWR@doingstartpars}%
```

If already in paragraph mode, do nothing.

```
3227 {}%
```

If not currently in paragraph mode:

```
3228 {%
```

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

```
3229 \PushPreHook{par}{\LWR@openparagraph}%
```

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

```
3230 \let\par\LWR@closeparagraph
3231
3232 }% an intentionally blank line
```

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

```
3233 \global\setbool{\LWR@doingstartpars}{true}%
```

No <par> tag yet to undo:

```
3234 \global\boolfalse{\LWR@doingapar}%
3235 }
```

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

```
3236 \newcommand*{\LWR@stoppars}%
3237 {%
```

See if currently handling HTML paragraphs:

```
3238 \ifbool{\LWR@doingapar}%
```

if currently in an HTML paragraph:

```
3239 {%
```

Print a closing tag:

```
3240 \unskip%
```

```
3241 \LWR@htmltagc{/\LWR@tagregularparagraph}%
```

No longer have an open HTML paragraph:

```
3242 \global\boolfalse{LWR@doingapar}%
```

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

```
3243 \global\boolfalse{LWR@minipagethispar}
```

```
3244
```

```
3245 }% an intentionally blank line
```

If was not in an HTML paragraph:

```
3246 {}%
```

See if currently allowing HTML paragraphs:

```
3247 \ifbool{LWR@doingstartpars}%
```

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

```
3248 {%
```

```
3249 \ClearPreHook{par}%
```

```
3250 }%
```

Else: do nothing

```
3251 {}%
```

no longer in paragraph mode

```
3252 \global\setbool{LWR@doingstartpars}{false}%
```

no `<p>` tag to undo:

```
3253 \global\boolfalse{LWR@doingapar}%
```

```
3254 }
```

```
3255 \end{warpHTML}
```

## 38 Page headers and footers

for HTML & PRINT: 3256 \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.

```
3257 \newcommand{\LWR@firstpagetop}{ } % for the home page alone
3258 \newcommand{\LWR@pagetop}{ } % for all other pages
3259 \newcommand{\LWR@pagebottom}{ }
3260
3261 \newcommand{\LWR@setfirstpagetopb}[1]{%
3262 \renewcommand{\LWR@firstpagetop}{#1}
3263 \catcode'\_ =8
3264 }
```

\HTMLFirstPageTop {*{text and logos}*}

```
3265 \newcommand{\HTMLFirstPageTop}{%
3266 \catcode'\_ =12
3267 \LWR@setfirstpagetopb
3268 }

3269 \newcommand{\LWR@setpagetopb}[1]{%
3270 \renewcommand{\LWR@pagetop}{#1}
3271 \catcode'\_ =8
3272 }
```

\HTMLPageTop {*{text and logos}*}

```
3273 \newcommand{\HTMLPageTop}{%
3274 \catcode'\_ =12
3275 \LWR@setpagetopb
3276 }

3277 \newcommand{\LWR@setpagebottomb}[1]{%
3278 \renewcommand{\LWR@pagebottom}{#1}
3279 \catcode'\_ =8
3280 }
```

\HTMLPageBottom {*{text and logos}*}

```
3281 \newcommand{\HTMLPageBottom}{%
3282 \catcode'\_ =12
3283 \LWR@setpagebottomb
3284 }
```

```
3285 \end{warpall}
```

## 39 CSS

**for HTML output:** 3286 \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.

```
3287 \newcommand*{\LWR@currentcss}{lwarp.css}
```

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

```
3288 \newcommand*{\LWR@newcssb}[1]{%
3289 \renewcommand*{\LWR@currentcss}{#1}
3290 \catcode'\_ =8
3291 }
3292
3293 \newcommand*{\CSSFilename}{
3294 \catcode'\_ =12
3295 \LWR@newcssb
3296 }
3297 \end{warpHTML}
```

**for PRINT output:** 3298 \begin{warpprint}  
 3299 \newcommand\*{\CSSFilename}[1]{}  
 3300 \end{warpprint}

## 40 HTML meta description and author

**for HTML & PRINT:** 3301 \begin{warpall}

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

```
3302 \newcommand{\theHTMLAuthor}{\theauthor}
3303
3304 \newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}

3305 \end{warpall}
```



for HTML & PRINT: 3306 \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` `{\langle New HTML meta description.\rangle}` Assigns the HTML file's description meta tag.

```
3307 \newcommand{\LWR@currentHTMLDescription}{}
3308
3309 \newcommand{\HTMLDescription}[1]{%
3310 \renewcommand{\LWR@currentHTMLDescription}{#1}
3311 }
3312
3313 \end{warpall}
```

## 41 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.

### 41.1 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 41.4 for the implementation.

### 41.2 Minipage footnotes

See section 67.2 for minipage footnotes.

### 41.3 Titlepage thanks

See section 48.6 for titlepage footnotes.

### 41.4 Regular page footnote implementation

for HTML output: 3314 \begin{warpHTML}

Patch L<sup>A</sup>T<sub>E</sub>X footnotes to use a new \box for lwarp footnotes.

3315 \newbox\LWR@footnotes

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

3316 \long\def\@makefnmark#1{\textsuperscript{\@thefnmark} #1}

3317

3318 \def\@makefnmark{\hbox{\textsuperscript{\@thefnmark}}}

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

3319 \long\def\@footnotetext#1{%

3320 \global\setbox\LWR@footnotes=\vbox{\unvbox\LWR@footnotes%

3321 \protected@edef\@currentlabel{%

3322 \csname p@footnote\endcsname\@thefnmark%

3323 }% \@currentlabel

3324 \color@begingroup%

3325 \ifbool{LWR@doingstartpars}{\LWR@htmltagc{\LWR@tagregularparagraph}}%

3326 \@makefnmark{#1}%

3327 \ifbool{LWR@doingstartpars}{\par}{\LWR@htmltagc{/LWR@tagregularparagraph}}%

3328 \color@endgroup%

3329 }% vbox

3330 }%

3331

3332 \long\def\@mpfootnotetext#1{%

3333 \global\setbox\@mpfootins\vbox{%

3334 \unvbox\@mpfootins

3335 \reset@font\footnotesize

3336 \hsize\columnwidth

3337 \@parboxrestore

3338 \protected@edef\@currentlabel

3339 {\csname p@mpfootnote\endcsname\@thefnmark}%

3340 \color@begingroup

3341 \@makefnmark{#1}%

3342 \ignorespaces#1%

3343 }%

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

```
3344 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{%
3345 {}%
3346 {\LWR@htmltagc{/LWR@tagregularparagraph}}}%

3347 \color@endgroup}%
3348 }
```

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

```
3349 \newcommand*{\LWR@printpendingfootnotes}{%
3350 \ifvoid\LWR@footnotes\else
3351 \LWR@forcenewpage
3352 \begin{BlockClass}{footnotes}
3353 \LWR@originedskip
3354 \unvbox\LWR@footnotes
3355 \setbox\LWR@footnotes=\vbox{}
3356 \end{BlockClass}
3357 \fi
3358 }
```

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

```
3359 \newcommand*{\LWR@epubprintpendingfootnotes}{%
3360 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}{%
3361 {\LWR@printpendingfootnotes}%
3362 {}%
3363 }

3364 \end{warpHTML}
```

## 42 Marginpars

for HTML output: 3365 \begin{warpHTML}

`\marginpar` [*left*] [*right*]

```
3366 \renewcommand{\marginpar}[2][]{%
3367 \LWR@htmlspanclass{marginpar}{#2}%
3368 }

3369 \end{warpHTML}
```

## 43 Splitting HTML files

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

**for HTML & PRINT:** 3370 `\begin{warpall}`

`Ctr FileDepth` `{\section depth}` determines how deeply to break into new HTML files, similar to `tocdepth`. The default of -5 produces one large HTML file.

```
3371 \newcounter{FileDepth}
3372 \setcounter{FileDepth}{-5}
```

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

```
3373 \newbool{CombineHigherDepths}
3374 \booltrue{CombineHigherDepths}
```

```
3375 \end{warpall}
```

**for HTML output:** 3376 `\begin{warpHTML}`

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

```
3377 \newcommand*{\LWR@thisfilename}{}
```

`\LWR@thisnewfilename` The filename being sanitized.

```
3378 \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.

```
3379 \newcommand*{\LWR@filenamenoblanks}[1]{%
3380 \begingroup
```

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

```
3381 \renewcommand*\HTMLUnicode}[1]{}
3382 \renewcommand*\HTMLentity}[1]{##1}
3383 \renewcommand*\LWR@htmltagc}[1]{}
3384 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}
```

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

```
3385 \renewcommand*\&{\and}
3386 \renewcommand\textit}[1]{##1}
3387 \renewcommand\textsc}[1]{##1}
3388 \renewcommand\textsl}[1]{##1}
3389 \renewcommand\textbf}[1]{##1}
3390 \renewcommand\texttt}[1]{##1}
3391 \renewcommand\textsf}[1]{##1}
3392 \renewcommand\textrm}[1]{##1}
3393 \renewcommand\textsuperscript}[1]{##1}
3394 \renewcommand\textsubscript}[1]{##1}
```

Replaces common symbols and short words with hyphens:

```
3395 \edef\LWR@thisnewfilename{#1}
3396 \fullexpandarg
```

Convert spaces into hyphens:

```
3397 \StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctuation into hyphens:

```
3398 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
3399 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
3400 \StrSubstitute{\LWR@thisnewfilename}%
3401 {\LWR@origampersand}{-}[\LWR@thisnewfilename]
3402 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
3403 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
3404 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
3405 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
3406 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
3407 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
3408 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
3409 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
3410 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
3411 \StrSubstitute{\LWR@thisnewfilename}%
3412 {\textless}{-}[\LWR@thisnewfilename]
3413 \StrSubstitute{\LWR@thisnewfilename}%
3414 {\textgreater}{-}[\LWR@thisnewfilename]
```

```

3415 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
3416 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
3417 \StrSubstitute{\LWR@thisnewfilename}{\}{-}[\LWR@thisnewfilename]
3418 \StrSubstitute{\LWR@thisnewfilename}{\}{-}[\LWR@thisnewfilename]
3419 \StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
3420 \StrSubstitute{\LWR@thisnewfilename}%
3421 {\textbackslash}{-}[\LWR@thisnewfilename]
3422 \StrSubstitute{\LWR@thisnewfilename}{^}{-}[\LWR@thisnewfilename]
3423 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
3424 %      "~{" for babel
3425 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
3426 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
3427 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]

```

Convert short words:

```

3428 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
3429 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
3430 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
3431 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
3432 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
3433 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
3434 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
3435 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
3436 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
3437 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
3438 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
3439 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
3440 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
3441 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
3442 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
3443 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
3444 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
3445 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

3446 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
3447 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
3448 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
3449 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
3450 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
3451 %      emdash
3452 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
3453 %      endash
3454 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
3455 \endgroup
3456 }

```

`\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.

```
3457 \newcommand*{\LWR@newhtmlfile}[1]{
```

At the bottom of the ending file:

```
3458 \LWR@htmlend{section}{textbody}
3459
3460 \LWR@printpendingfootnotes
3461
```

No footer between files if EPUB:

```
3462 \ifbool{FormatEpub}
3463 {}
3464 {
3465 \LWR@htmlend{footer}
3466
3467 \LWR@pagebottom
3468
3469 \LWR@htmlend{footer}
3470 }
```

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

```
3471 \ifthenelse{\boolean{FormatEpub}\OR\boolean{FormatWordProcessor}}
3472 {}
3473 {\ifnumcomp{\value{\LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}}
```

End of this HTML file:

```
3474 \LWR@stoppars
3475 \LWR@htmltag{/body}\LWR@orignewline
3476 \LWR@htmltag{/html}\LWR@orignewline
3477 \LWR@orignewpage
3478
3479 \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.

```

3480 \ifbool{FileSectionNames}%
3481 {\LWR@filenameno blanks{#1}}
3482 {\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.

```

3483 \LWR@htmlblockcomment{%
3484 |Start file|
3485 \LWR@htmlsectionfilename{\LWR@thisfilename}|
3486 }

```

At the top of the starting file:

```

3487 \LWR@stoppars
3488
3489 \LWR@filestart{ -- #1}% there is an Emdash in front of the #1
3490

```

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

```

3491 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3492 {}
3493 {\LWR@topnavigation}
3494

```

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

```

3495 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3496 {}
3497 {
3498 \LWR@html element{header}
3499
3500 \LWR@pagetop
3501
3502 \LWR@html elementend{header}
3503 }
3504

```

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

```

3505 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3506 {}
3507 {\ifcvoid{thetitle}{ }\LWR@printthetitle}
3508

```



No sideTOC if formatting for an EPUB or word processor:

```
3509 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3510 {}
3511 {\LWR@sidetoc}
3512
```

Start of the <textbody>:

```
3513 \LWR@htmlclass{section}{textbody}
3514
```

Keep paragraph tags disabled for now:

```
3515 \LWR@stoppars
3516
```

Track the page numbers:

```
3517 \setcounter{LWR@latestautopage}{\value{page}}
3518 }
```

```
3519 \end{warpHTML}
```

## 44 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:** 3520 \begin{warpHTML}

### 44.1 Book class commands

**\mainmatter** Declare the main matter section of the document. Does not reset the page number, which must be consecutive arabic numbers for the HTML conversion.



```

3521 \newbool{LWR@mainmatter}
3522 \DeclareDocumentCommand{\mainmatter}{-}{-}{%
3523 \booltrue{LWR@mainmatter}%
3524 }

```

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

```

3525 \DeclareDocumentCommand{\frontmatter}{-}{-}{%
3526 \boolfalse{LWR@mainmatter}%
3527 }

```

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

```

3528 \DeclareDocumentCommand{\backmatter}{-}{-}{%
3529 \boolfalse{LWR@mainmatter}
3530 }

```

## 44.2 Sectioning support macros

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

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

```

3531 \newcommand*{\LWR@sectionnumber}[1]{%
3532 \InlineClass{sectionnumber}{#1} %
3533 }

```

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

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

Create an autosection tag.

```

3534 \newcommand*{\LWR@createautosec}[1]{%
3535 \LWR@htmltag{#1 id="autosec-\thepage"{} }%
3536 }

```

**\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.

```

3537 \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.

```
3538 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
3539 \LWR@closeprevious{#1}%
```

Push a new section depth:

```
3540 \LWR@pushoneclose{#1}{#2}%
```

```
3541 }
```

**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.

```
3542 \newcounter{LWR@prevFileDepth}
```

```
3543 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

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

The common actions for the high-level sectioning commands.

```
3544 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
```

```
3545 \LWR@traceinfo{LWR@section}%
```

```
3546 \LWR@stoppars%
```

Cancel special `minipage` horizontal space interaction:

```
3547 \global\boolfalse{LWR@minipagethispar}%
```

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

```
3548 \LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
```

```
3549 \IfBooleanTF{#1}{% starred
```

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

```
3550 \LWR@orignewpage%
```

```
3551
```

```

3552 }{% not starred
3553 \ifthenelse{%
3554 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}}%
3555 \AND%
3556 \(%
3557 \NOT\boolean{CombineHigherDepths}\OR%
3558 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}}%
3559 \)%
3560 }%

```

If so: start a new HTML file:

```

3561 {% new file
3562 \LWR@traceinfo{LWR@section: new HTML file}%

```

See if there was an optional TOC name entry:

```

3563 \IfNoValueTF{#2}%

```

If no optional entry

```

3564 {\LWR@newhtmlfile{#3}}%

```

If yes an optional entry

```

3565 {\LWR@newhtmlfile{#2}}%
3566 }% new file

```

Else: No new HTML file:

```

3567 {% not new file

```

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

```

3568 \LWR@orignewpage%
3569
3570 }% not new file
3571 }% not starred

```

Remember this section's name for \nameref:

```

3572 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
3573 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%

```

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

```

3574

```

```

3575 \ifbool{HTMLDebugComments}{%
3576 \LWR@htmlcomment{Opening #4 ‘‘#3’’}{}}
3577 }{}
3578

```

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

```

3579 \ifthenelse{%
3580 \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
3581 }%
3582 {\LWR@startpars}
3583 {}

```

Create the opening tag with an autosec:

```

3584 \LWR@createautosec{\csuse{LWR@tag#4}}%

```

If not starred, step counter and add to TOC:

```

3585 \IfBooleanTF{#1}%
3586 {}% starred
3587 {% not starred

```

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

```

3588 \ifthenelse{%
3589 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
3590 }%
3591 {% if secnumdepth

```

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

```

3592 \LWR@traceinfo{LWR@section: about to test main matter}%
3593 \ifbool{LWR@mainmatter}%
3594 {%
3595 \LWR@traceinfo{LWR@section: yes mainmatter}%
3596 \refstepcounter{#4}%

```

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

```

3597 \LWR@traceinfo{LWR@section: about to addcontentsline}%
3598 \addcontentsline{toc}{#4}%
3599 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
3600 \IfValueTF{#2}{#2}{#3}}%
3601 \LWR@traceinfo{LWR@section: finished addcontentsline}%
3602 }% end of if main matter

```

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

```
3603 {% not main matter
3604 \LWR@traceinfo{LWR@section: no main matter}%
3605 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
3606 }% end of not main matter
3607 }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```
3608 {%
3609 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
3610 }%
```

For part, print the section type:

```
3611 \ifbool{LWR@mainmatter}%
3612 {%
3613 \ifthenelse{%
3614 \(\cnttest{\csuse{LWR@depth#4}}{<=}%
3615 {\value{secnumdepth}}\)\ \AND
3616 \(\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\)
3617 }%
3618 {\csuse{#4name}~{}}%
3619 {}%
```

Print the section number:

```
3620 \LWR@traceinfo{LWR@section: about to print section number}%
3621 \ifthenelse{%
3622 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
3623 }%
3624 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
3625 {}%
3626 \LWR@traceinfo{LWR@section: finished print section number}%
3627 }{}%
3628 }% end of not starred
```

Print the section name:

```
3629 #3
```

close the heading tag, such as /H2

```
3630 \LWR@htmltag{\csuse{LWR>tag#4end}}%
```

Generate a L<sup>A</sup>T<sub>E</sub>X label:

```
3631 \label{autopage-\thepage}%
```

Start paragraph handing unless is an inline paragraph or subparagraph:

```

3632 \ifthenelse{%
3633 \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}}%
3634 {\LWR@startpars}%
3635 {}%

```

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.

```

3636 \IfBooleanTF{#1}{\% not starred
3637 \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
3638 }{\% not starred
3639 \LWR@traceinfo{LWR@section: done}%
3640 }

```

### 44.3 \section and friends

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

```

3641 \@ifundefined{chapter}
3642 {}
3643 {%
3644 \DeclareDocumentCommand{\part}{s o m}{%
3645 \LWR@epubprintpendingfootnotes%
3646 \LWR@stoppars%
3647
3648 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
3649
3650 \LWR@section{#1}{#2}{#3}{part}%
3651 }
3652 }

```

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

```

3653 \@ifundefined{chapter}
3654 {}
3655 {%
3656 \DeclareDocumentCommand{\chapter}{s o m}{%
3657 \LWR@traceinfo{chapter #3}%
3658 \LWR@epubprintpendingfootnotes%

```

```

3659 \LWR@stoppars%
3660 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
3661
3662 \LWR@traceinfo{chapter: about to LWR@section}%
3663 \LWR@section{#1}{#2}{#3}{chapter}%
3664 \LWR@traceinfo{chapter: done}%
3665 }
3666 }

```

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

```

3667 \DeclareDocumentCommand{\section}{s o m}{%
3668 \LWR@epubprintpendingfootnotes%
3669 \LWR@stoppars%
3670
3671 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
3672
3673 \LWR@section{#1}{#2}{#3}{section}%
3674 }

```

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

```

3675 \DeclareDocumentCommand{\subsection}{s o m}{%
3676 \LWR@epubprintpendingfootnotes%
3677 \LWR@stoppars%
3678
3679 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
3680
3681 \LWR@section{#1}{#2}{#3}{subsection}%
3682 }

```

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

```

3683 \DeclareDocumentCommand{\subsubsection}{s o m}{%
3684 \LWR@epubprintpendingfootnotes%
3685 \LWR@stoppars%
3686
3687 \LWR@startnewdepth{\LWR@depthsubsubsection}%
3688 {\LWR@printclosesubsubsection}%
3689
3690 \LWR@section{#1}{#2}{#3}{subsubsection}%
3691 }

```

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

```

3692 \DeclareDocumentCommand{\paragraph}{s o m}{%
3693 \LWR@epubprintpendingfootnotes%

```



```

3694 \LWR@stoppars%
3695
3696 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
3697
3698 \LWR@section{#1}{#2}{#3}{paragraph}%
3699 }

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

3700 \DeclareDocumentCommand{\subparagraph}{s o m}{%
3701 \LWR@epubprintpendingfootnotes%
3702 \LWR@stoppars%
3703
3704 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
3705
3706 \LWR@section{#1}{#2}{#3}{subparagraph}%
3707 }

3708 \end{warpHTML}

```

## 45 Starting a new file

**for HTML & PRINT:** 3709 \begin{warpall}

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

```

3710 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
3711
3712 \newcommand*{\HTMLLanguage}[1]{%
3713 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
3714 }

3715 \end{warpall}

```

**for HTML output:** 3716 \begin{warpHTML}

`\LWR@filestart` {(<title\_\_suffix>)}

Creates the opening HTML tags.

```

3717 \newcommand*{\LWR@filestart}[1]{

```

Locally temporarily disable direct-formatting commands:

```

3718 \begingroup
3719 \renewcommand{\textit}[1]{##1}% not used in filenames
3720 \renewcommand{\textsc}[1]{##1}
3721 \renewcommand{\textsl}[1]{##1}
3722 \renewcommand{\textbf}[1]{##1}
3723 \renewcommand{\texttt}[1]{##1}
3724 \renewcommand{\textsf}[1]{##1}
3725 \renewcommand{\textrm}[1]{##1}
3726 \renewcommand{\textsuperscript}[1]{##1}
3727 \renewcommand{\textsubscript}[1]{##1}
3728 \renewcommand*{\HTMLUnicode}[1]{ }
3729 \renewcommand*{\HTMLentity}[1]{ }
3730 \RenewDocumentCommand{\LWR@htmlspanclass}{m o +m}{##3}
3731 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}

```

Create the page's HTML header:

```

3732 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline

```

The language is user-adjustable:

```

3733 \LWR@htmltag{html lang="\LWR@currentHTMLELanguage"}\LWR@orignewline

```

Start of the meta data:

```

3734 \LWR@htmltag{head}\LWR@orignewline

```

Charset is fixed at UTF-8:

```

3735 \LWR@htmltag{meta charset="UTF-8" /\LWR@orignewline

```

Author:

```

3736 \ifcempty{theHTMLAuthor}{ }{
3737 \LWR@htmltag{meta name="author" content="\theHTMLAuthor" /\LWR@orignewline
3738 }

```

lwarp is the generator:

```

3739 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /\LWR@orignewline
3740

```

If there is a description, add it now:

```

3741 \ifdefempty{\LWR@currentHTMLDescription}{ }{%
3742 \LWR@htmltag{%
3743 meta name="description" content="\LWR@currentHTMLDescription" /\LWR@orignewline
3744
3745 }%

```

Mobile-friendly viewport:

```
3746 \LWR@htmltag{meta name="viewport" %
3747 content="width=device-width, initial-scale=1.0" /}%
3748 \LWR@orignewline
```

IE patch:

```
3749 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
3750 \LWR@htmltag{%
3751 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{}%
3752 \LWR@htmltag{/script}\LWR@orignewline
3753 \LWR@htmltag{![endif]{-}{-}}\LWR@orignewline
```

The page's title:

```
3754 \ifcvoid{thetitle}{}{%
3755 \LWR@htmltag{title}\thetitle#1\LWR@htmltag{/title}\LWR@orignewline%
3756 }%
```

The page's stylesheet:

```
3757 \LWR@htmltag{%
3758 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
3759 \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.

```
3760 \ifbool{mathjax}%
3761 {%
3762 \boolfalse{LWR@verbtags}
3763 \VerbatimInput{lwarp_mathjax.txt}%
3764 \booltrue{LWR@verbtags}
3765 \LWR@stoppars
3766 }% end of mathjax
3767 {}%
```

End of the header:

```
3768 \LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
3769 \LWR@htmltag{body}\LWR@orignewline
3770 \endgroup
3771 }
```

```
3772 \end{warpHTML}
```

## 46 Starting HTML output

for HTML output: 3773 \begin{warpHTML}

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

```
3774 \newcommand*{\LWR@LwarpStart}
3775 {%
```

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

```
3776 \ifbool{FormatWordProcessor}{%
3777 \setcounter{FileDepth}{-5}%
3778 \boolfalse{HTMLDebugComments}%
3779 }{}
```

Expand and detokenize \HomeHTMLFilename and \HTMLFilename:

```
3780 \edef\LWR@strresult{\HomeHTMLFilename}
3781 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
3782 \edef\LWR@strresult{\HTMLFilename}
3783 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn:

```
3784 \LWR@origonecolumn%
```

Reduce chance of line overflow in verbatim environments:

```
3785 \LWR@origscriptsize%
```

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

```
3786 \LWR@origraggedright%
3787 \let\\LWR@endoffline%
```

Spread the lines for pdftotext to read them well:

```
3788 \linespread{1.3}%
```

For pdftotext to reliably identify paragraph splits:

```
3789 \setlength{\parindent}{0pt}
3790 \setlength{\parskip}{2ex}
```

For the `lateximages` record file:

```
3791 \immediate\openout\LWR@file=lateximages.txt
```

Removes space after the caption in the HTML:

```
3792 \setlength{\belowcaptionskip}{-3ex}
```

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

```
3793 \renewcommand{\ps@plain}{} 
```

```

\centering Not used in the HTML environment:
\raggedleft
\raggedright 3794 \renewcommand*{\centering}{}
3795 \renewcommand*{\raggedleft}{}
3796 \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:

```
3797 \let\LWR@origtabular\tabular
3798 \let\LWR@origendtabular\endtabular
3799 \let\tabular\LWR@tabular
3800 \let\endtabular\endLWR@tabular
```

Float captions:

```
3801 \let\LWR@origcaption\caption
```

Labels: `\ltx@label` is used in `amsmath` environments and is also patched by `cleveref`.

Label in HTML

```
3802 \let\LWR@origltx@label\ltx@label
3803 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
3804 \let\TeX\LWR@TeX
3805 \let\LaTeX\LWR@LaTeX
3806 \let\LuaTeX\LWR@LuaTeX
3807 \let\LuaLaTeX\LWR@LuaLaTeX
3808 \let\XeTeX\LWR@XeTeX
```

```
3809 \let\XeLaTeX\LWR@XeLaTeX
3810 \let\ConTeXt\LWR@ConTeXt
```

Graphics:

```
3811 \let\rotatebox\LWR@rotatebox
3812 \let\scalebox\LWR@scalebox
3813 \let\reflectbox\LWR@reflectbox
```

Not yet started any paragraph handling:

```
3814 \global\boolfalse\LWR@doingapar}
3815 \global\boolfalse\LWR@doingstartpars}
```

Start a new HTML file and a header:

```
3816 \LWR@filestart{}
3817 \LWR@htmltag{header}\LWR@orignewline
3818 \LWR@startpars
3819 \LWR@firstpagetop
3820 \LWR@stoppars
3821 \LWR@htmltag{/header}\LWR@orignewline
3822 \LWR@htmltag{section class="textbody"{} }
3823 \LWR@origpagestyle{empty}
```

Document and page settings:

```
3824 \mainmatter
3825 \LWR@origpagenumbering{arabic}
```

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

```
3826 \if@titlepage
3827   \thanksmarkseries{arabic}
3828 \else
3829   \thanksmarkseries{fnsymbol}
3830 \fi
```

Initial default patch for fancyvrb:

```
3831 \fvset{frame=none}%
```

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

```
3832 \catcode'\&=\active
```

Allow HTML paragraphs to begin:

```

3833 \LWR@startpars
3834 }

3835 \end{warpHTML}

```

## 47 Ending HTML output

**for HTML output:** 3836 \begin{warpHTML}

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

```

3837 \newcommand*{\LWR@requesttoc}[2]{%
3838 \ifbool{#1}
3839 {
3840 \expandafter\newwrite\csuse{tf@#2}
3841 \immediate\openout \csuse{tf@#2} \jobname.#2\relax
3842 }{}
3843 }

```

\LWR@LwarpEnd Final stop of all HTML output:

```

3844 \newcommand*{\LWR@LwarpEnd}
3845 {
3846 \LWR@stoppars
3847 \LWR@closeprevious{\LWR@depthfinished}

```

At the bottom of the ending file:

Close the textbody:

```
3848 \LWR@html-elementclassend{section}{textbody}
```

Print any pending footnotes:

```
3849 \LWR@printpendingfootnotes
```

Create the footer:

```

3850 \LWR@html-element{footer}
3851
3852 \LWR@pagebottom
3853
3854 \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.

```
3855 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3856 {}
3857 {
3858 \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{
3859 }
```

```
3860 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
3861 \LWR@htmltag{/body}\LWR@orignewline
3862 \LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
3863 \LWR@orignewpage
```

For lateximage commands:

```
3864 \immediate\closeout\LWR@file
3865 }
```

```
3866 \end{warpHTML}
```

## 48 Titles and the titling package

Supports and extends the titling package.

Additional functions include `\published` and `\subtitle`, and the `\author` command has an additional `\affiliation` command to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using `\theauthor` in the main text.

The titling package maintains the definitions of `\thetitle`, `\theauthor`, etc., after the title has been typeset. These commands are to be used to refer to the document's title and author, etc., in the main text. These definitions have the `\thanks` and `\affiliation` removed, and for author the `\and` is replaced to generate a simple inline list of authors separated by commas.

`\printtitle`, `\printauthor`, etc., are provided for use inside the `titlepage` or `titlingpage` environments, and these retain the `\thanks` and `\affiliation`.



Several additional hooks are provided in addition to `titling`:

<code>\maketitlehookaa</code>	<code>\maketitlehookaa:</code> Between “published” and the title.
<code>\maketitlehookaaa</code>	<code>\maketitlehookaaa:</code> Between the title and the subtitle.
<code>\prepublished</code>	<code>\prepublished:</code> Before the “published” field.
<code>\postpublished</code>	<code>\postpublished:</code> After the “published” field.
<code>\presubtitle</code>	<code>\presubtitle:</code> Before the subtitle.
<code>\postsubtitle</code>	<code>\postsubtitle:</code> After the subtitle.
<code>\printthanks</code>	<code>\printthanks</code> has been added to force the printing of thanks inside a <code>titlingpage</code> environment when <code>\maketitle</code> is not used.

### ⚠ No footnotes!

Inside a `\titlepage` or `\titlingpage` environment, use `\thanks` for footnotes, do not use `\footnote`.

At the end of the `titlingpage` environment, footnote marks are forced to reset to zero.

Inside a `titlingpage` environment with the `article` document class, thanks marks will be `fnsymbol` instead of `arabic`. `arabic` is usually used when inside `titlepage` environments where the title page is on its own page, but is not automatically used inside a `titlingpage` environment.

To force the thanks marks to be `arabic`:

---

```
\begin{titlingpage}
\thanksmarkseries{arabic}
...
```

---

## 48.1 Setting the title, etc.

The following provide setting commands for both HTML and print outputs.

<code>\published</code>	<code>\@title</code> , <code>\@subtitle</code> , <code>\@author</code> , etc. store the values as originally assigned, including any <code>\thanks</code> , <code>\and</code> , or <code>\affiliation</code> . These are low-level macros intended to be used by other macros only inside a <code>titlepage</code> or <code>titlingpage</code> , and are used by <code>\maketitle</code> . The author is printed inside a single-column table, which becomes multiple single-column tables if multiples authors are included.
<code>\title</code>	
<code>\subtitle</code>	
<code>\author</code>	
<code>\date</code>	

`\printpublished` `\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.

`\thepublished` `\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:

`\HTMLPageBottom` `{\text}`

---

```
\HTMLPageBottom{
\begin{center}\textcopyright~2016 \theauthor\end{center}
}
```

---

`\author` `{\author}` While using `\maketitle`, the author is treated as a single-column table and the `\and` feature finishes the current table then starts a new one for the next author. Each author thus is placed into its own table, and an affiliation may be placed on its own line such as

```
\author{Name \\ Affiliation \and Second Name \\ Second Affiliation}
```

After `\maketitle` has completed, `\theauthor` retains the definition of the author, but `\and` is changed to become a comma and a space, intending to print the authors names separated by spaces. This fails when affiliations are included on their own table rows.

A solution, provide here, is to define a macro `\affiliation` which during `\maketitle` starts a new table row and adds the affiliation, but after `\maketitle` is finished `\affiliation` is re-defined to throw away its argument, thus printing only the author names when `\author` is later used inline.

`\affiliation` `{\text}`

Adds the affiliation to the author for use in `\maketitle`. Nullified when later used for inline use of `\theauthor`.

```
for HTML output: 3867 \begin{warpHTML}
3868 \newrobustcmd{\affiliation}[1]{\\ \InlineClass{affiliation}{#1}}
3869 \end{warpHTML}
```

for PRINT output: 3870 `\begin{warpprint}`  
 3871 `\newrobustcmd{affiliation}[1]{\textsc{small#1}}`  
 3872 `\end{warpprint}`

The following are based on the original titling code:

for HTML & PRINT: 3873 `\begin{warpall}`

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

Redefined to nullify `\affiliation`, etc. before printing the authors inline.

`@author` retains the entire author with its `\thanks`, while `\theauthor` will have `\thanks` removed and `\and` simplified.

```
3874 \renewcommand{\author}[1]{%
3875 \gdef\@author{#1}
3876 \begingroup
3877   \renewcommand{\thanks}[1]{ }
3878   \renewcommand{\and}{\unskip, }
3879   \renewcommand{\thanksmark}[1]{ }
3880   \renewcommand{\thanksgap}[1]{ }
3881   \renewcommand{\affiliation}[1]{ }
3882   \protected@xdef\theauthor{#1}
3883 \endgroup}
```

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

```
3884 \newcommand{\published}[1]{%
3885 \gdef\@published{#1}
3886 \begingroup
3887   \renewcommand{\thanks}[1]{ }
3888   \renewcommand{\thanksmark}[1]{ }
3889   \renewcommand{\thanksgap}[1]{ }
3890   \protected@xdef\thepublished{#1}
3891 \endgroup
3892 }
3893 \newcommand{\@published}{ }
3894 \newcommand{\thepublished}{ }
```

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

```
3895 \newcommand{\subtitle}[1]{%
3896 \gdef\@subtitle{#1}
3897 \begingroup
3898   \renewcommand{\thanks}[1]{ }
3899   \renewcommand{\thanksmark}[1]{ }
```

```

3900 \renewcommand{\thanksgap}[1]{}
3901 \protected@xdef\thesubtitle{#1}
3902 \endgroup
3903 }
3904 \newcommand{\@subtitle}{}
3905 \newcommand{\thesubtitle}{}

3906 \end{warpall}

```

## 48.2 Changes to HTML titlepage and titlingpage

for HTML output: 3907 \begin{warpHTML}

Env **titlepage** Sets up a titlepage div with a L<sup>A</sup>T<sub>E</sub>X PDF minipage inside.

```

3908 \renewenvironment*{titlepage}
3909 {
3910 \LWR@forcenewpage
3911 \BlockClass{titlepage}\LWR@subminipage
3912 }
3913 {\LWR@endsubminipage\endBlockClass}

```

Env **titlingpage**

```

3914 \renewenvironment*{titlingpage}
3915 {%

```

Start an HTML titlepage div:

```

3916 \begin{titlepage}

```

Prepare for a custom version of \maketitle inside the titlingpage:

```

3917 \LWR@maketitlesetup
3918 \let\maketitle\LWR@titlingmaketitle
3919 }
3920 {

```

At the end of the environment, end the HTML titlepage div:

```

3921 \end{titlepage}

```

Reset the footnote counter:

```

3922 \@bscontmark
3923 }

```

```
3924 \end{warpHTML}
```

for HTML & PRINT: 3925 \begin{warpall}

\printthanks Forces the \thanks to be printed.

This is necessary in a titlingpage environment when \maketitle was not used.

```
3926 \newcommand*{\printthanks}{\@thanks}
```

Env titlingpage At the end of the titlingpage for both print and HTML, reset footnote markers to zero.

```
3927 \AtEndEnvironment{titlingpage}{\@bscontmark}
```

```
3928 \end{warpall}
```

### 48.3 Printing the title, etc. in HTML

The following are for printing the title, etc. in a titlepage or a titlingpage in HTML:

for HTML output: 3929 \begin{warpHTML}

Patch the pre/post title/author/date to add HTML tags, then initialize:

```
3930 \newcommand{\prepublished}[1]{%
3931 \def\@bspublished{\BlockClass{published}#1}%
3932 }
3933
3934 \newcommand{\postpublished}[1]{%
3935 \def\@bspublished{#1\endBlockClass}%
3936 }
3937
3938 \renewcommand{\prettitle}[1]{%
3939 \def\@bsprettitle{#1\LWR@stoppars\LWR@htmltag{h1}}%
3940 }
3941
3942 \renewcommand{\posttitle}[1]{%
3943 \def\@bsposttitle{\LWR@htmltag{/h1}\LWR@startpars#1}%
3944 }
3945
3946 \newcommand{\presubtitle}[1]{%
3947 \def\@bspresubtitle{\BlockClass{subtitle}#1}%
3948 }
3949
```

```

3950 \newcommand{\postsubtitle}[1]{%
3951 \def\@bspostsubtitle{#1\endBlockClass}%
3952 }
3953
3954 \renewcommand{\preauthor}[1]{%
3955 \def\@bspreauthor{BlockClass{author}#1}%
3956 }
3957
3958 \renewcommand{\postauthor}[1]{%
3959 \def\@bspostauthor{#1\endBlockClass}%
3960 }
3961
3962 \renewcommand{\predate}[1]{%
3963 \def\@bspredate{#1BlockClass{titledate}}%
3964 }
3965
3966 \renewcommand{\postdate}[1]{%
3967 \def\@bspostdate{\endBlockClass#1}%
3968 }
3969
3970 \prepublished{\begin{center}}
3971 \postpublished{\par\end{center}}
3972
3973 \pretitle{\begin{center}}
3974 \posttitle{\par\end{center}}
3975
3976 \presubtitle{\begin{center}}
3977 \postsubtitle{\par\end{center}}
3978
3979 \preauthor{\begin{center}}%
3980 \begin{tabular}[t]{c}%
3981 }
3982 \postauthor{\end{tabular}\par\end{center}}
3983
3984 \predate{\begin{center}}
3985 \postdate{\par\end{center}}

```

\printpublished

```

3986 \newcommand*{\printpublished}{
3987 \ifthenelse{\equal{\thepublished}{}}{
3988 {}
3989 {
3990 \begin{BlockClass}{published}
3991 \@published
3992 \end{BlockClass}
3993 }
3994 }

```

`\printtitle`

```

3995 \newcommand*{\printtitle}
3996 {
3997 \LWR@stoppars
3998 \LWR@htmltag{h1}%
3999 \@title%
4000 \LWR@htmltag{/h1}
4001 \LWR@startpars
4002 }

```

`\LWR@printthetitle` A private version which prints the title without footnotes, used to title each HTML page.

```

4003 \newcommand*{\LWR@printthetitle}
4004 {
4005 \LWR@stoppars
4006 \LWR@htmltag{h1}%
4007 \thetitle%
4008 \LWR@htmltag{/h1}
4009 \LWR@startpars
4010 }

```

`\printsubtitle`

```

4011 \newcommand*{\printsubtitle}{
4012 \ifthenelse{\equal{\thesubtitle}{}}{
4013 {}
4014 {
4015 \begin{BlockClass}{subtitle}
4016 \@subtitle
4017 \end{BlockClass}
4018 }
4019 }

```

`\printauthor`

```

4020 \newcommand*{\printauthor}{
4021 \begin{BlockClass}{author}
4022 \begin{tabular}{c}\@author\end{tabular}
4023 \end{BlockClass}
4024 }

```

`\printdate`

```

4025 \newcommand*{\printdate}{%
4026 \begin{BlockClass}{titledate}

```

```

4027 \@date
4028 \end{BlockClass}
4029 }

4030 \end{warpHTML}

```

## 48.4 Printing the title, etc. in print form

The following are for printing the title, etc. in a `titlepage` or a `titlingpage` in print form:

```

for PRINT output: 4031 \begin{warpprint}

\printpublished

4032 \newcommand*{\printpublished}{\Large\scshape\@published}}

\printtitle

4033 \newcommand*{\printtitle}{\Huge\@title}}

\printsubtitle

4034 \newcommand*{\printsubtitle}{\Large\itshape\@subtitle\bigskip}}

\printauthor

4035 \newcommand*{\printauthor}
4036   {\large\begin{tabular}[t]{c}\@author\end{tabular}}}

\printdate

4037 \newcommand*{\printdate}{\small\textit{\@date}}}

```

## 48.5 \maketitle for print output

`\maketitle` From the titling package, patched to add the publisher and subtitle.

```

4038 \providecommand{\maketitle}{}
4039 \if@titlepage
4040   \renewcommand{\maketitle}{\begin{titlepage}%

```



```

4041 \let\footnotesize\small
4042 \let\footnoterule\relax
4043 \let \footnote \thanks
4044 \@bsmarkseries
4045 \def\@makefnmark{\rlap{\@textsuperscript{%
4046 \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
4047 \long\def\@makefntext##1{\makethanksmark ##1}
4048 \null\vfil
4049 \vskip 60\p@
4050 \vspace*{\droptitle}
4051 \maketitlehooka
4052 \ifcsempy{\@published}
4053 {}
4054 {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
4055 {\@bspretitle \@title \@bsposttitle}
4056 \ifcsempy{\@subtitle}
4057 {}
4058 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
4059 \maketitlehookb
4060 {\@bspreauthor \@author \@bspostauthor}
4061 \maketitlehookc
4062 {\@bspredate \@date \@bspostdate}
4063 \maketitlehookd
4064 \par
4065 \@thanks
4066 \vfil\null
4067 \end{titlepage}%
4068 \@bscontmark % \setcounter{footnote}{0}%
4069 %%% \@bsmtitleempty
4070 } % end titlepage defs
4071 \else
4072 \renewcommand{\maketitle}{\par
4073 \begingroup
4074 \@bsmarkseries
4075 \def\@makefnmark{\rlap{\@textsuperscript{%
4076 \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
4077 \long\def\@makefntext##1{\makethanksmark ##1}
4078 \if@twocolumn
4079 \ifnum \col@number=\@ne
4080 \@maketitle
4081 \else
4082 \twocolumn[\@maketitle]%
4083 \fi
4084 \else
4085 \newpage
4086 \global\@topnum\z@
4087 \@maketitle
4088 \fi
4089 \thispagestyle{plain}\@thanks
4090 \endgroup

```

```

4091 \bscontmark % \setcounter{footnote}{0}%
4092 %% \bsmttitleempty
4093 } % end non-titlepage
4094
4095 \def\@maketitle{%
4096 \newpage
4097 \null
4098 \vskip 2em%
4099 \vspace*\droptitle}
4100 \maketitlehooka
4101 \ifcsempy{published}
4102 {}
4103 {{\bsprepublished \@published \bspostpublished}\maketitlehookaa}
4104 {\bsprettitle \@title \bsposttitle}
4105 \ifcsempy{subtitle}
4106 {}
4107 {\maketitlehookaaa{\bspresubtitle \@subtitle \bspostsubtitle}}
4108 \maketitlehookb
4109 {\bspreauthor \@author \bspostauthor}
4110 \maketitlehookc
4111 {\bspredate \@date \bspostdate}
4112 \maketitlehookd
4113 \par
4114 \vskip 1.5em}
4115 \fi
4116
4117 \providecommand{\maketitlehookaa}{}
4118
4119 \providecommand{\maketitlehookaaa}{}
4120
4121 \newcommand{\prepublished}[1]{%
4122 \def\bsprepublished{#1}%
4123 }
4124
4125 \newcommand{\postpublished}[1]{%
4126 \def\bspostpublished{#1}%
4127 }
4128
4129 \newcommand{\presubtitle}[1]{%
4130 \def\bspresubtitle{#1}%
4131 }

```

\presubtitle Hook after printing the subtitle.

```

4132 \newcommand{\postsubtitle}[1]{%
4133 \def\bspostsubtitle{#1}%
4134 }

```

Initial settings:

```

4135 \if@titlepage
4136 \prepublished{
4137 \vspace*{-\baselineskip}\vspace*{-\medskipamount}\vspace*{-2em}
4138 \begin{center}}
4139 \postpublished{\par\end{center}\vskip 2em}
4140
4141 \presubtitle{\unskip\begin{center}\unskip}
4142 \postsubtitle{\par\end{center}\vskip 2em}
4143 \else
4144 \prepublished{\begin{center}}
4145 \postpublished{\par\end{center}\vskip 0.5em}
4146
4147 \presubtitle{\begin{center}\unskip}
4148 \postsubtitle{\par\end{center}\vskip 0.5em}
4149 \fi
4150 \end{warpprint}

```

## 48.6 \maketitle for HTML output

An HTML div of class `titlepage` is created, inside of which a L<sup>A</sup>T<sub>E</sub>X PDF minipage is generated (without HTML tags), allowing the `\thanks` footnotes to be generated immediately at the end of the title page during HTML output. This is desirable when a large table of contents immediately follows the title.

`\thanks` are a form of footnotes used in the title page. See section 41 for other kinds of footnotes.

See `\thanksmarkseries{series}`, below, to set the style of the footnote marks.

for HTML output: 4151 \begin{warppHTML}

\LWR@maketitlesetup Patches `\thanks` macros to use L<sup>A</sup>T<sub>E</sub>X minipage footnotes.

```
4152 \newcommand*{\LWR@maketitlesetup}{%
```

Select which kind of footnote marks to use:

```

4153 \@bsmarkseries
4154 \@mpbsmarkseries

```

Redefine the footnote mark:

```
4155 \def\@makefnmark{\textsuperscript{\thefootnote}}
```

`\thefootnote`  $\Rightarrow$  `\nameuse{arabic}{footnote}`, or  
`\thefootnote`  $\Rightarrow$  `\nameuse{fnsymbol}{footnote}`

Redefine the footnote text:

```
4156 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
4157 \makethanksmark \LWR@origspace{1in}
```

`\makethanksmark`  $\Rightarrow$  `\thanksfootmark`  $\Rightarrow$  `\tamark`  $\Rightarrow$   
`\@thefnmark`  $\Rightarrow$  `\itshape` a (or similar)

Print the text:

```
4158 ##1%
4159 }%
4160 }
```

`\@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`.

```
4161 \def\@fnsymbol#1{\ensuremath{\ifcase#1\or *\or \dagger\or \ddagger\or
4162 \mathsection\or \mathparagraph\or \text{\HTMLUnicode{2016}}\or
4163 **\or \dagger\dagger \or \ddagger\ddagger \else\@ctrerr\fi}}
```

`\maketitle` Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

```
4164 \renewcommand*\maketitle{%
```

An HTML titlepage div is used for all classes.

```
4165 \begin{titlepage}
```

Set up special patches:

```
4166 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4167 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
4168 \@thanks
```

Close the HTML titlepage div:

```
4169 \end{titlepage}
```

Reset the footnote counter:

```
4170 \@bscontmark
```

```
4171 }
```

`\@maketitle` Typesets the title, etc. for HTML:

```
4172 \DeclareDocumentCommand{\@maketitle}{-}{%
4173 \maketitlehooka
4174 \ifcempty{@published}
4175 {}
4176 {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
4177 {\@bsprettitle \@title \@bsposttitle}
4178 \ifcempty{@subtitle}
4179 {}
4180 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
4181 \maketitlehookb
4182 {\@bspreauthor \@author \@bspostauthor}
4183 \maketitlehookc
4184 {\@bspredate \@date \@bspostdate}
4185 \maketitlehookd
4186 }

4187 \providecommand{\maketitlehookaa}{}
4188 \providecommand{\maketitlehookaaa}{}

```

`\LWR@titlingmaketitle` `\maketitle` for use inside an HTML titlingpage environment.

```
4189 \newcommand*{\LWR@titlingmaketitle}{%
```

Typeset the title, etc:

```
4190 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
4191 \@thanks
```

```
4192 }
```

`\thanksmarkseries`  $\{\langle series \rangle\}$

Sets the type of footnote marks used by `\thanks`, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc. Modified to use the L<sup>A</sup>T<sub>E</sub>X PDF minipage which is included with the title page.

```
4193 \renewcommand{\thanksmarkseries}[1]{%
4194 \def\@mpbsmarkseries{%
4195 \renewcommand*{\thempfootnote}{\@nameuse{#1}{mpfootnote}}}%
4196 \def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
4197 }

4198 \end{warpHTML}
```

## 49 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: 4199 `\begin{warpHTML}`

`\abstractname` User-redefinable title for the abstract.

Also over-written by the `babel` package.

```
4200 \providecommand*{\abstractname}{Abstract}
```

Env `abstract`

```
4201 \DeclareDocumentEnvironment{abstract}{}
4202 {
4203 \LWR@forcenewpage
4204 \BlockClass{abstract}
4205 \BlockClassSingle{abstracttitle}{\abstractname}
4206 }
4207 {
4208 \endBlockClass
4209 }

4210 \end{warpHTML}
```

## 50 Quote and verse

### 50.1 Citations and attributions

`\attribution` for use inside quote, quotation, verse:

ex: `\attribution{author name} --- \citetitle{book name}`

**for HTML output:** 4211 `\begin{warpHTML}`  
 4212 `\newcommand{\attribution}[1]{%`  
 4213 `\InlineClass{attribution}{--\,#1}}% emdash`  
 4214 `\end{warpHTML}`

**for PRINT output:** 4215 `\begin{warpprint}`  
 4216 `\newcommand{\attribution}[1]{\textsc{---\,#1}}`  
 4217 `\end{warpprint}`

`\citetitle` for use inside quote, quotation, verse:

**for HTML output:** 4218 `\begin{warpHTML}`  
 4219 `\newcommand{\citetitle}[1]{%`  
 4220 `\InlineClass{citetitle}{--\,#1}}% emdash`  
 4221 `\end{warpHTML}`

**for PRINT output:** 4222 `\begin{warpprint}`  
 4223 `\newcommand{\citetitle}[1]{\textsl{---\,#1}}`  
 4224 `\end{warpprint}`

### 50.2 Quotes, quotations

**for HTML output:** 4225 `\begin{warpHTML}`

Env `quote`

```
4226 \renewenvironment*{quote}
4227 {
4228 \LWR@forcenewpage
4229 \LWR@htmlblocktag{blockquote}
4230 }
4231 {\LWR@htmlblocktag{/blockquote}}
4232
4233 \renewenvironment*{quotation}
4234 {
4235 \LWR@forcenewpage
4236 \LWR@htmlblocktag{blockquotation}
```

```

4237 }
4238 {\LWR@htmlblocktag{/blockquotation}}

4239 \end{warpHTML}

```

### 50.3 Verse

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

---

```

\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}

```

---

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

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

**for HTML & PRINT:** 4240 `\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.

```

4241 \newlength{\HTMLvleftskip}
4242 \setlength{\HTMLvleftskip}{1em}

```

Len `\TMLleftmargini` Sets `\leftmargini` inside a `verse` environment in HTML.



```

4243 \newlength{\HTMLleftmargini}
4244 \setlength{\HTMLleftmargini}{4.5em}

4245 \end{warpall}

```

## 51 Verbatim

**for HTML output:** 4246 \begin{warpHTML}

Env    **verbatim**

```

4247 \AfterEndPreamble{
4248 \AtBeginEnvironment{verbatim}{%
4249 \LWR@forcenewpage
4250 \LWR@atbeginverbatim{verbatim}\unskip\vspace*{--\baselineskip}%
4251 }
4252 \AfterEndEnvironment{verbatim}{\unskip\vspace*{--\baselineskip}\LWR@afterendverbatim}
4253 }

4254 \end{warpHTML}

```

## 52 Fancyvrb

**for HTML & PRINT:** 4255 \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.

```

4256 \newlength{\VerbatimHTMLWidth}
4257 \setlength{\VerbatimHTMLWidth}{4in}
4258 \end{warpall}

```

**for HTML output:** 4259 \begin{warpHTML}

Bool    **LWR@verbtags**    Used to temporarily turn off verbatim tags while doing VerbatimInput in the HTML head.

```

4260 \newbool{LWR@verbtags}
4261 \booltrue{LWR@verbtags}

```

For `\VerbatimFootnotes`:

```
4262 \renewcommand{\VerbatimFootnotes}{
4263 \PackageError{lwarp}
4264 {Verbatim footnotes are not yet supported by lwarp.}
4265 {This may be improved some day.}
4266 }
```

`\LWR@atbeginverbatim` `{\class}`

Encloses a verbatim environment with the given CSS class.

```
4267 \newcommand*{\LWR@atbeginverbatim}[1]
4268 {%
```

Avoid excessive space between lines:

```
4269 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
4270 \LWR@stoppars%
```

Create a new `pre` of the given class:

```
4271 \ifbool{\LWR@verbtags}{\LWR@htmltag{pre class="#1"{}}}{}%
```

Use a mono-spaced font to preserve horizontal positioning. If horizontal alignment is important for the user, use a mono-spaced font in the CSS for the `verse` class.

```
4272 \LWR@origttfamily%
```

Do not produce HTML tags for `\hspace` inside a verse `par`. Restore plain L<sup>A</sup>T<sub>E</sub>X `\hspace` functionality:

```
4273 \let\hspace\LWR@orighspace%
4274 }
```

`\LWR@afterendverbatim` Finishes enclosing a verbatim environment.

```
4275 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the `pre`:

```
4276 \unskip%
```

At the end of the environment, close the `pre`:

```

4277 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}
4278
4279 }{}%

```

Resume regular paragraph handling:

```

4280 \LWR@startpars%
4281 }

```

`\LWR@Verbatimclass` Holds the class of the following verbatim.

```

4282 \newcommand*{\LWR@Verbatimclass}{fancyvrb}

```

`Env VerbatimClass` `{\langle class \rangle} [\langle Verbatim options \rangle]`

Creates a `Verbatim` enclosed in a `<div>` of the given class.

```

4283 \NewDocumentEnvironment{VerbatimClass}{m 0{}}
4284 {%
4285 \renewcommand*{\LWR@Verbatimclass}{#1}%
4286 \LWR@origVerbatim[#2]%
4287 }
4288 {\endVerbatim}

```

After the preamble is loaded, after any patches to `Verbatim`:

```

4289 \AfterEndPreamble{

```

Remember the original definition of `Verbatim`:

```

4290 \let\LWR@origVerbatim\Verbatim

```

`Env Verbatim` Patched to place the environment in a `fancyvrb` div, and the label in a `fancyvrblabel` div. Also corrects the left margin for line numbers. Also uses `VerbatimHTMLWidth` to control placement of line numbers on the right. Aligning the right margin requires knowing the width.

```

4291 \renewcommand*{\Verbatim}{%
4292 \LWR@forcenewpage
4293 \renewcommand*{\LWR@Verbatimclass}{fancyvrb}%
4294 \LWR@origVerbatim%
4295 }

```

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.

```

4296 \newcommand*{\LWR@fvstartnone}{%
4297 \LWR@traceinfo{fvstartnone}%
4298 \ifbool{LWR@verbtags}{\hbox to\z@{\LWR@htmltagc{div class="\LWR@Verbatimclass"}}}{}%
4299 \hbox to\z@{\LWR@atbeginverbatim{verbatim}}}%
4300 }
4301
4302 \newcommand*{\LWR@fvendnone}{%
4303 \LWR@traceinfo{fvendnone}%
4304 \hbox to\z@{\LWR@afterendverbatim}%
4305 \ifbool{LWR@verbtags}{\hbox to\z@{\LWR@htmltagc{/div}}}{}%
4306 }
4307
4308 \newcommand*{\LWR@fvstartsingle}{%
4309 \LWR@traceinfo{fvstartsingle}%
4310 \LWR@fvstartnone%
4311 \FV@BeginListFrame@Single%
4312 }
4313
4314 \newcommand*{\LWR@fvendsingle}{%
4315 \LWR@traceinfo{fvendsingle}%
4316 \FV@endListFrame@Single%
4317 \LWR@fvendnone%
4318 }
4319
4320 \newcommand*{\LWR@fvstartline}{%
4321 \LWR@traceinfo{fvstartline}%
4322 \LWR@fvstartnone%
4323 \FV@BeginListFrame@Lines%
4324 }
4325
4326 \newcommand*{\LWR@fvendline}{%
4327 \LWR@traceinfo{fvendline}%
4328 \FV@endListFrame@Lines%
4329 \LWR@fvendnone%
4330 }

```

The following patches select the start/left/right/end behaviors depending on **frame**.  
Original code is from the fancyvrb package.

```

4331 \def\FV@Frame@none{%
4332 \let\FV@BeginListFrame\LWR@fvstartnone%
4333 \let\FV@LeftListFrame\relax%
4334 \let\FV@RightListFrame\relax%
4335 \let\FV@endListFrame\LWR@fvendnone%
4336
4337 \def\FV@Frame@single{%
4338 \let\FV@BeginListFrame\LWR@fvstartsingle%
4339 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
4340 \let\FV@RightListFrame\FV@RightListFrame@Single%

```

```

4341 \let\FV@EndListFrame\LWR@fvendsingle}
4342
4343 \def\FV@Frame@lines{%
4344 \let\FV@BeginListFrame\LWR@fvstartline%
4345 \let\FV@LeftListFrame\relax%
4346 \let\FV@RightListFrame\relax%
4347 \let\FV@EndListFrame\LWR@fvendline}
4348
4349 \def\FV@Frame@topline{%
4350 \let\FV@BeginListFrame\LWR@fvstartline%
4351 \let\FV@LeftListFrame\relax%
4352 \let\FV@RightListFrame\relax%
4353 \let\FV@EndListFrame\LWR@fvendnone}
4354
4355 \def\FV@Frame@bottomline{%
4356 \let\FV@BeginListFrame\LWR@fvstartnone%
4357 \let\FV@LeftListFrame\relax%
4358 \let\FV@RightListFrame\relax%
4359 \let\FV@EndListFrame\LWR@fvendline}
4360
4361 \def\FV@Frame@leftline{%
4362 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
4363 \ifx\FancyVerbFillColor\relax%
4364 \let\FV@FrameFillLine\relax%
4365 \else%
4366 \@tempdima\FV@FrameRule\relax%
4367 \multiply\@tempdima-\tw@%
4368 \edef\FV@FrameFillLine{%
4369 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}}%
4370 \kern-\number\@tempdima sp}}%
4371 \fi%
4372 \let\FV@BeginListFrame\LWR@fvstartnone%
4373 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
4374 \let\FV@RightListFrame\relax%
4375 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the `fancyvrb` package.

```

4376 \def\FV@SingleFrameLine#1{%
4377 \hbox to\z@{%
4378 % \kern\leftmargin
4379 \ifnum#1=\z@\relax
4380 \let\FV@Label\FV@LabelBegin
4381 \else
4382 \let\FV@Label\FV@LabelEnd
4383 \fi
4384 \ifx\FV@Label\relax
4385 % \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%

```

```

4386 \else
4387 \ifnum#1=\z@
4388 % \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
4389 \ifx\FV@LabelPositionTopLine\relax
4390 \else
4391 \LWR@htmltagc{div class="fancyvrblabel"}
4392 \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
4393 \LWR@htmltagc{/div}
4394 \fi
4395 \else
4396 % \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
4397 \ifx\FV@LabelPositionBottomLine\relax
4398 \else
4399 \LWR@htmltagc{div class="fancyvrblabel"}
4400 \LWR@origtextrm{\FV@LabelEnd}
4401 \LWR@htmltagc{/div}
4402 \fi
4403 \fi
4404
4405 \fi
4406 \hss
4407 }
4408 }

```

Processes each line, adding optional line numbers. Original code is from the fancyvrb package.

```

4409 \def\FV@ListProcessLine#1{%
4410 \hbox to \hsize{%
4411 % \kern\leftmargin
4412 \hbox to \VerbatimHTMLWidth {%
4413 \ifcvoid\FV@LeftListNumber}{-}{\kern 2.5em}%
4414 \FV@LeftListNumber%
4415 % \FV@LeftListFrame
4416 \FancyVerbFormatLine{#1}%
4417 \hss%
4418 % \FV@RightListFrame
4419 \FV@RightListNumber%
4420 }%
4421 \hss% required to avoid underfull hboxes
4422 }
4423 }

```

Env BVerbatim

```

4424 \AtBeginEnvironment{BVerbatim}
4425 {
4426 \LWR@forcenewpage
4427 \LWR@atbeginverbatim{bverbatim}

```

```

4428
4429 }
4430
4431 \AfterEndEnvironment{BVerbatim}
4432 {
4433 \leavevmode\par\vspace{-\baselineskip}
4434 \LWR@afterendverbatim
4435 }

```

Env **LVerbatim** No changes required.

End of the modifications to make at the end of the preamble:

```

4436 } % \AfterEndPreamble

```

**\UseVerbatim**  $\{\langle text \rangle\}$

No changes required.

```

4437 \end{warpHTML}

```

## 53 Theorems

**\newtheorem**  $\{\langle text \rangle\}$  [ $\langle counter \rangle$ ] -or- [ $\langle oldname \rangle$ ]  $\{\langle text \rangle\}$

A few minor changes are made to supply HTML tags.

- The entire theorem is placed into a div of class **theoremcontents**.
- The label for each theorem is placed inside a span of class **theoremlabel**.
- The contents are placed inside a div of class **theoremcontents**.

for HTML output: 4438 **\begin{warpHTML}**

**\@begintheorem**  $\{\langle name \rangle\}$   $\{\langle number \rangle\}$

```

4439 \renewcommand{\@begintheorem}[2]{%
4440 \LWR@forcenewpage
4441 \BlockClass{theoremcontents}
4442 \InlineClass{theoremlabel}{#1\ #2\ }
4443 }

```

```

\@opargbegintheorem {\langle name\rangle} {\langle number\rangle} {\langle oparg\rangle}

4444 \renewcommand{\@opargbegintheorem}[3]{%
4445 \LWR@forcenewpage
4446 \BlockClass{theoremcontents}
4447 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
4448 }


\@endtheorem

4449 \renewcommand*\@endtheorem{%
4450 \endBlockClass% theoremcontents
4451 }

4452 \end{warpHTML}

```

## 54 Lists

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

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

to preserve the special HTML and enumitem list handling.

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

### 54.1 Itemize

**for HTML output:** 4453 \begin{warpHTML}

```
4454 \let\LWR@origitem\item
```

```
\LWR@itemizeitem [\langle label\rangle]
```

Handles \item inside an itemize or enumerate.

See \LWR@openparagraph where extra \hspace is used to leave room for the label while inside a list during paragraph construction.

```

4455 \newcommand*\LWR@itemizeitem{%
4456 \LWR@stoppars%
4457 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem{}}%
4458 \LWR@htmltag{li}%

```



---

```

4459 \LWR@startpars%
4460 \LWR@origitem%
4461 }

```

To have a blank item, use `\mbox{}`. This forces a new line in print output, matching the new line which will appear in HTML output. Ex:

---

```

begin{itemize}
item \mbox{}
\begin{itemize}
...

```

---

Env `itemize` [*(enumitem options)*]

```

4462 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
4463
4464 \newcommand*{\LWR@itemizestart}{%
4465 \LWR@stoppars%
4466 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4467 \LWR@htmltag{ul style="list-style-type:none">{}}%
4468 \LWR@startpars%
4469 \let\item\LWR@itemizeitem%
4470 }
4471
4472 \AtEndEnvironment{itemize}{\LWR@itemizeend}
4473
4474 \newcommand*{\LWR@itemizeend}{%
4475 \LWR@stoppars%
4476 \LWR@closeprevious{\LWR@depthlistitem}%
4477 \LWR@closeoneprevious{}}%
4478 \LWR@startpars%
4479 }

```

## 54.2 Enumerate

An HTML unordered list is used with customized L<sup>A</sup>T<sub>E</sub>X-generated labels.

Env `enumerate` [*(enumitem options)*]

```

4480 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
4481
4482 \newcommand*{\LWR@enumeratestart}{%
4483 \LWR@stoppars%

```

```

4484 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4485 \LWR@htmltag{ul style="list-style-type:none">{}}%
4486 \LWR@startpars%
4487 \let\item\LWR@itemizeitem%
4488 }
4489
4490
4491 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
4492
4493 \newcommand*{\LWR@enumerateend}{%
4494 \LWR@stoppars%
4495 \LWR@closeprevious{\LWR@depthlistitem}%
4496 \LWR@closeoneprevious{}}%
4497 \LWR@startpars%
4498 }

```

### 54.3 Description

`\LWR@descitem` [*(label)*] Handles an `\item` inside a description.

```

4499 \newcommand*{\LWR@descitem}[1][]%
4500 {%
4501 \LWR@stoppars%
4502 \LWR@setlatestname{#1}%
4503 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem{}}%
4504 \LWR@origitem[]%

```

Be sure the label doesn't print to the left of the rest of the file:

```

4505 \LWR@origspace{1in}
4506 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
4507 \LWR@orignewline%
4508 \LWR@htmltag{dd}%
4509 \LWR@startpars%
4510 }

```

Env `description` [*(enumitem options)*]

```

4511 \AtBeginEnvironment{description}{\LWR@descriptionstart}
4512
4513 \newcommand*{\LWR@descriptionstart}{%
4514 \LWR@stoppars%
4515 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}%
4516 \LWR@htmltag{dl}%
4517 \LWR@startpars%
4518 \let\item\LWR@descitem%

```

```

4519 }
4520
4521 \AtEndEnvironment{description}{\LWR@descriptionend}
4522
4523 \newcommand*{\LWR@descriptionend}{%
4524 \LWR@stoppars%
4525 \LWR@closeprevious{\LWR@depthlistitem}%
4526 \LWR@closeoneprevious{}%
4527 \LWR@startpars%
4528 }

```

```
\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.

```

4529 \let\LWR@orignewlist\newlist
4530
4531 \renewcommand*{\newlist}[3]{%
4532 \LWR@orignewlist{#1}{#2}{#3}%
4533 \AtBeginEnvironment{#1}{\csuse{\LWR@#2start}}%
4534 \AtEndEnvironment{#1}{\csuse{\LWR@#2end}}%
4535 }

4536 \end{warpHTML}

```

## 55 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

Limitations:

### column types

- Vertical rules are not yet supported.
- `*` in a column specification is not used (so far). Repeat the column type the correct number of times.
- Only one each of `@`, `!`, `>`, and `<` may be used at each column, and they are used in that order.
- `\newcolumntype` is ignored; unknown column types are set to `l`.

⚠ `\multirow & \multicolumn`

⚠ `\multirow`

`\multirow` with rules

rule at last row

⚠ paragraphs

`\cmidrule` width, trim

`longtable` headings

⚠ `\warpprintonly`

⚠ `S` columns

- `tabularx` ignores the width, but `X` columns do produce paragraph columns or multicolumns.
- `Multirow` and `multicolumn` cannot be used at the same time. (No rectangular holes wider than one column or taller than one row.)
- For `multirow`, insert `\mrowcell` into any empty multi-row cells. This will be a null function for the print output, and is a placeholder for parsing the table for HTML output.
- If a `multirow` reaches to the bottom of a table, and `\bottomrule` does not go over to that edge, try adding a line of empty cells below the `\bottomrule`. This may be a browser bug.
- If a `\midrule` is desired after the last row, an additional row of blank cells must be used.
- Multiple paragraphs in one cell of a `p`, `b`, `m` column must have `\newline` between paragraphs.
- `\cmidrule` does not support width or trim options due to CSS limitations.
- For `longtable`, place headings and footings which do not apply to HTML inside `\warpprintonly{}`.
- For `\toprule` and `\bottomrule`, when combined with a `warpprint` or `warppHTML` environment, if a “misplaced `\noalign`” error occurs, change
 

```
This & That \endhead
```

 to
 

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

 and likewise with the other `\end` headings. Keep the `\endfirsthead` row unchanged, as it is still relevant to HTML output.
- For `S` columns (from the `siunitx` package), while producing print output, anything non-numeric must be placed inside `{ }` braces, including commands such as `\multirow`. While producing HTML output, though, anything placed inside braces is not seen by `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 \\}
```

## 55.1 Token lookahead

Used by `\LWR@futurenonospacelet` to look at the next token.

for HTML output: 4537 `\begin{warppHTML}`

`\LWR@mynexttoken`

```
4538 \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`

```
4539 \def\LWR@futurenonSPACElet#1{\def\LWR@cs{#1}%
4540 \afterassignment\LWR@fns lone\let\nexttoken= }
4541 \def\LWR@fns lone{\expandafter\futurelet\LWR@cs\LWR@fns ltwo}
4542 \def\LWR@fns ltwo{%
4543 \expandafter\ifx\LWR@cs\@sptoken\let\next=\@BTfns lthree%
4544 \else\let\next=\nexttoken\fi\next}
4545 \def\@BTfns lthree{\afterassignment\LWR@fns lone\let\next= }
```

`\LWR@getmynexttoken` Looks ahead and copies the next token into `\LWR@mynexttoken`.

```
4546 \newcommand*{\LWR@getmynexttoken}{%
4547 % nothing must follow this next line
4548 \LWR@futurenonSPACElet\LWR@mynexttoken\LWR@tabledatacolumn tag
4549 }
```

## 55.2 Booleans

Bool `LWR@startedrow` True if should print a row tag before this column.

```
4550 \newbool{LWR@startedrow}
4551 \boolfalse{LWR@startedrow}
```

Bool `LWR@doinghline` True if the next row will have an hline above it.

```
4552 \newbool{LWR@doinghline}
4553 \boolfalse{LWR@doinghline}
```

Bool `LWR@doingtbrule` True if the next row will have a top/bottom rule above it.

```
4554 \newbool{LWR@doingtbrule}
4555 \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.

```
4556 \newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty multi-row cell, and thus there is no data tag to close.

```
4557 \newbool{LWR@skippingmrowcell}
```

Bool LWR@intabularmetadata True if are in a tabular but not in a data cell. Used to prevent extra HTML breaks if not inside table data.

```
4558 \newbool{LWR@intabularmetadata}
```

```
4559 \boolfalse{LWR@intabularmetadata}
```

### 55.3 Handling & and !

For technical discussion regarding problems redefining \&, See:

[http://tex.stackexchange.com/questions/11638/  
where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860](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.

```
4560 \newcommand*{\LWR@closetabledatacell}{%
4561 \global\booltrue{LWR@intabularmetadata}%
4562 \ifbool{LWR@exitingtabular}{}%
4563 {% not exiting tabular
4564 \ifbool{LWR@skippingmrowcell}{}%
4565 {% not skippingmrowcell
```

Insert any < then any @ and ! column contents:

```
4566 \unskip%
4567 \LWR@getexpparray{LWR@colafterspec}{\theLWR@tablecolspos}%
4568 \LWR@getexpparray{LWR@colatspec}{\theLWR@tablecolspos}%
4569 \LWR@getexpparray{LWR@colbangspec}{\theLWR@tablecolspos}%
```

Close paragraphs:

```
4570 \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
4571 \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell:

```
4572 \unskip\LWR@htmltag{/td}\LWR@orignewline%
4573 }% not skipping mrowcell
```

```

4574 }% not exiting tabular
4575 \global\boolfalse{LWR@skippingmrowcell}%
4576 }

```

LWR@tabulardepth tracks whether & is being used inside a `tabular`.

```

4577 \newcounter{LWR@tabulardepth}
4578 \setcounter{LWR@tabulardepth}{0}
4579

```

When not used inside a `tabular`, & performs its original function as recorded here ( with catcode 4 ).

```

4580 \def\LWR@origampmacro{&}

```

See below for why the group is used.

```

4581 \begingroup

```

& Will behave depending on whether it is being used inside `tabular`.

& is redefined to test whether it is inside a tabular environment, in which case it performs special processing for HTML conversion. If not, it behaves normally.

The `\catcode` allows the & character to be redefined.

```

4582 \catcode'\&=\active
4583
4584 \gdef&{%
4585 \ifthenelse{\cnttest{\value{LWR@tabulardepth}}{>}{0}}%
4586 {%

```

If not skipping a multirow cell, close the current data cell.

```

4587 \unskip%
4588 \LWR@closetabledatacell%

```

Move to the next column.

```

4589 \addtocounter{LWR@tablecolspos}{1}%

```

Look at the next token to decide multi or single column data tag.

```

4590 \LWR@getmynexttoken%
4591 }%

```

If not inside a `tabular`, performs the original action:

```

4592 {\LWR@origampmacro}%
4593 }
4594 \endgroup

```

Outside the group, & is left its original catcode for now.

tikz package seems to require & be left alone until after tikz has been loaded.

\LWR@lwarpStart finally makes & active at the beginning of the HTML conversion.

## 55.4 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendoffline

Throws away options \\[dim] or \\\*

\LWR@tabularendoffline

```

4595 \NewDocumentCommand{\LWR@tabularendoffline}{s o}
4596 {%
4597 \LWR@closetabledatacell%

```

Finish the previous row:

```

4598 \LWR@htmltag{/tr}\LWR@orignewline
4599 \global\booltrue{\LWR@intabularmetadata}

```

Not yet started a table row:

```

4600 \global\boolfalse{\LWR@startedrow}

```

Additional setup:

```

4601 \global\boolfalse{\LWR@doinghline}%
4602 \global\boolfalse{\LWR@doingtbrule}%
4603 \LWR@clearmidrules%

```

Start at first column:

```

4604 \setcounter{\LWR@tablecolspos}{1}

```

Look at the next token to decide between single column data tag or a special case:

```

4605 \LWR@getmynexttoken%
4606 }

```



## 55.5 Variables

```

4607 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
4608 \newcommand*{\LWR@pposition}{%
4609 \newcommand*{\LWR@pleft}{%
4610 \newcommand*{\LWR@pright}{%

```

**\LWR@tablecolspec** Holds the parsed column specification, of total width **LWR@tabletotalcols**.  
 Will contain a string such as `llrrccpc`, exactly one letter per column, without `@`, `>`, `<`, or the vertical pipe.

```

4611 \newcommand*{\LWR@tablecolspec}{%

```

**\LWR@strresult** Holds the result of **Str** functions.

```

4612 \newcommand*{\LWR@strresult}{%

```

**\LWR@origcolspec** Holds the original column specs given to **tabular**.

```

4613 \newcommand*{\LWR@origcolspec}{%

```

**Ctr LWR@tablecolswidth** Holds the width of the table specification.  
 (This is not the total # columns.)

```

4614 \newcounter{LWR@tablecolswidth}

```

**Ctr LWR@tablecolspos** Where are currently looking into the table column specification.

```

4615 \newcounter{LWR@tablecolspos}

```

**Ctr LWR@tabletotalcols** Holds the final number of table columns.

```

4616 \newcounter{LWR@tabletotalcols}

```

**Ctr LWR@tabletotalcolsnext** Holds the next column while parsing. Is one more than **LWR@tabletotalcols**.

```

4617 \newcounter{LWR@tabletotalcolsnext}

```

**LWR@colatspec** A data array of specifications for `@` columns. The leftmost's index is **leftedge**, the others are counter values. See section 28.

**LWR@colbangspec** A data array of specifications for `!` columns. The leftmost's index is **leftedge**, the others are counter values. See section 28.

**LWR@colbeforespec** A data array of specifications for `>` columns.

**LWR@colafterspec** A data array of specifications for `<` columns.

## 55.6 Parsing @, >, <, ! columns

`\LWR@parseatcolumn` Handles @{text} columns.

```
4618 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
4619 \LWR@traceinfo{at column}%
4620 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into `\LWR@strresult`, expanding once:

```
4621 \LWR@traceinfo{about to read the next token:}%
4622 \expandarg%
4623 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]
4624 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
4625 \LWR@traceinfo{have now read the next token}%
4626 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}{
4627 {% left edge of the table:
4628 \LWR@traceinfo{at the left edge}%
4629 \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@strresult}%
4630 }%
4631 {% not at the left edge:
4632 \LWR@traceinfo{not at the left edge}%
4633 \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
4634 \LWR@traceinfo{at \theLWR@tabletotalcols: %
4635 \LWR@getexparray{LWR@colatspec}{\theLWR@tabletotalcols}}!}%
4636 \let\LWR@strresult\relax%
4637 \booltrue{LWR@validtablecol}%
4638 }%
4639 }
```

`\LWR@parsebangcolumn`

```
4640 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
4641 \LWR@traceinfo{bang column}%
4642 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into `\LWR@strresult`, expanding once:

```

4643 \LWR@traceinfo{about to read the next token:}%
4644 \expandarg%
4645 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]
4646 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@strresult:

```

4647 \LWR@traceinfo{have now read the next token}%
4648 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}{
4649 {% left edge of the table:
4650 \LWR@traceinfo{at the left edge}%
4651 \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@strresult}%
4652 }%
4653 {% not at the left edge:
4654 \LWR@traceinfo{not at the left edge}%
4655 \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
4656 \LWR@traceinfo{bang \theLWR@tabletotalcols: \LWR@colbangspec(\theLWR@tabletotalcols)!}%
4657 }%
4658 \let\LWR@strresult\relax%
4659 \booltrue{LWR@validtablecol}%
4660 }

```

\LWR@parsebeforecolumn Handles >{text} columns.

```

4661 \newcommand*{\LWR@parsebeforecolumn}{%

```

Move to the next token after the '>':

```

4662 \addtocounter{LWR@tablecolspos}{1}%

```

Read the next token, expanding once into \LWR@strresult:

```

4663 \expandarg%
4664 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
4665 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@strresult:

```

4666 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\LWR@strresult}%
4667 \let\LWR@strresult\relax%
4668 \booltrue{LWR@validtablecol}%
4669 }

```

\LWR@parseaftercolumn Handles <{text} columns.

```

4670 \newcommand*{\LWR@parseaftercolumn}{%

```

Move to the next token after the '<':

```
4671 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token, expanding once into \LWR@strresult:

```
4672 % \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]
4673 \expandarg%
4674 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
4675 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```
4676 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
4677 \let\LWR@strresult\relax%
4678 \booltrue{LWR@validtablecol}%
4679 }
```

**\LWR@parseskipcolumn** Handles columns to skip, such as the vertical bar.

```
4680 \newcommand*{\LWR@parseskipcolumn}{%
4681 \booltrue{LWR@validtablecol}%
4682 }
```

## 55.7 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.

```
4683 \newcommand*{\LWR@parsenormalcolumn}[1]{%
4684 \appto\LWR@tablecolspec{#1}%
4685 \addtocounter{LWR@tabletotalcols}{1}%
4686 \addtocounter{LWR@tabletotalcolsnext}{1}%
4687 \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcolsnext}{\relax}%
4688 \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcolsnext}{\relax}%
4689 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\relax}%
4690 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcolsnext}{\relax}%
4691 \booltrue{LWR@validtablecol}%
4692 }
```

## 55.8 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn`  $\{ \langle thiscolumn \rangle \}$  The width will be ignored.

```
4693 \newcommand*\LWR@parsepcolumn}[1]{%
```

Converts to the given column type.

```
4694 \LWR@parsenormalcolumn{#1}%
```

skips the following width

```
4695 \addtocounter{LWR@tablecolspos}{1}%
4696 }
```

## 55.9 Parsing ‘D’ columns

From the dcolumn package.

`\LWR@parseDcolumn`  $\{ \langle thiscolumn \rangle \}$  The three parameters will be ignored.

```
4697 \newcommand*\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
4698 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
4699 \addtocounter{LWR@tablecolspos}{3}%
4700 }
```

## 55.10 Parsing the column specifications



HTML CSS cannot exactly match the L<sup>A</sup>T<sub>E</sub>X concept of a baseline for a table row. Table 7 shows the L<sup>A</sup>T<sub>E</sub>X 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.

```
4701 \newbool{LWR@validtablecol}
```

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, L<sup>A</sup>T<sub>E</sub>X places the top line of a parbox aligned with the rest of the text line, so CSS `vertical-align:bottom` is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the CSS may be desired, changing `td.tdp`, `td.tdP`, `td.tdprule`, and `td.tdPrule` to `vertical-align: middle`. Another possibility is to change L,R,C, and P to `vertical-align: top` and not worry about the alignment of B and M cells or trying to approximate L<sup>A</sup>T<sub>E</sub>X 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.

`\LWR@parsetablecols`  $\{\langle colspecs \rangle\}$

Scans the column specification left to right.

Builds `\LWR@tablecolspec` with the final specification, one column per entry. The number of final columns is stored in `LWR@tabletotalcols`.

```
4702 \newcommand*{\LWR@parsetablecols}[1]{%
4703 \LWR@traceinfo{\LWR@parsetablecols started}%
```

Remember the original supplied column spec:

```
4704 \renewcommand*{\LWR@origcolspec}{#1}%
```

Clear the parsed resulting column spec:

```
4705 \renewcommand*{\LWR@tablecolspec}{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
4706 \setcounter{LWR@tabletotalcols}{0}%
4707 \setcounter{LWR@tabletotalcolsnext}{1}%
4708 \LWR@setexparray{LWR@colatspec}{leftedge}{\relax}%
4709 \LWR@setexparray{LWR@colatspec}{1}{\relax}%
4710 \LWR@setexparray{LWR@colatspec}{2}{\relax}%
4711 \LWR@setexparray{LWR@colatspec}{3}{\relax}%
4712 \LWR@setexparray{LWR@colbangspec}{leftedge}{\relax}%
4713 \LWR@setexparray{LWR@colbangspec}{1}{\relax}%
4714 \LWR@setexparray{LWR@colbangspec}{2}{\relax}%
4715 \LWR@setexparray{LWR@colbangspec}{3}{\relax}%
4716 \LWR@setexparray{LWR@colbeforespec}{1}{\relax}%
4717 \LWR@setexparray{LWR@colbeforespec}{2}{\relax}%
4718 \LWR@setexparray{LWR@colbeforespec}{3}{\relax}%
4719 \LWR@setexparray{LWR@colafterspec}{1}{\relax}%
4720 \LWR@setexparray{LWR@colafterspec}{2}{\relax}%
4721 \LWR@setexparray{LWR@colafterspec}{3}{\relax}%
```

Starting at the first column specification:

```
4722 \setcounter{LWR@tablecolspos}{1}%
```

Place the colspecs string length into `\LWR@strresult`, and remember the number of characters in the column specification:

```
4723 \LWR@traceinfo{about to StrLen}%
4724 \noexpandarg%
4725 \StrLen{#1}[\LWR@strresult]%
4726 \fullexpandarg%
```

```
4727 \LWR@traceinfo{finished StrLen}%
4728 \setcounter{LWR@tablecolswidth}{\LWR@strresult}%
```

Scan through the column specifications:


```
4729 \whileof\not\value{LWR@tablecolspos}>\value{LWR@tablecolswidth}}{%
```

Place the next single-character column type into \LWR@strresult:

```
4730 \noexpandarg%
4731 \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]%
4732 \fullexpandarg%
```

Not yet found a valid column type

```
4733 \boolfalse{LWR@validtablecol}%
```

 Note that the parameter for a `p{spec}` column is a token list which will NOT match `l,c,r,p`.

```
4734 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{%
4735 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{%
4736 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{%
4737 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{%
4738 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{%
4739 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{%
4740 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{%
4741 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{%
4742 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
4743 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
4744 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
4745 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
4746 \IfStrEq{\LWR@strresult}{|}{\LWR@parseskipcolumn}{}%
4747 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{%
4748 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{%
4749 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{%
```

From the `dcolumn` package:

```
4750 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{%
```

From the `tabularx` package. `X` column has no parameter, but will be given paragraph tags.

```
4751 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{%
```

---

Many people define centered versions “P”, “M”, and “B”:

```
\newcolumnntype{P}[1]{>\centering\arraybackslash}p{#1}}
```

---



```

4752 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
4753 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
4754 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%

```

If this column was an invalid column type, convert it to a p column:

```

4755 \ifbool{\LWR@validtablecol}{}{}%
4756 \LWR@parsenormalcolumn{1}%
4757 }%
4758 \addtocounter{\LWR@tablecolspos}{1}%
4759 }%
4760 }%

```

### 55.11 Starting a new row

`\LWR@maybenewtablerow` If have not yet started a new table row, begin one now. Creates a new row tag, adding a class for hline or tbrule if necessary.

```

4761 \newcommand*{\LWR@maybenewtablerow}
4762 {%
4763 \ifbool{\LWR@startedrow}%
4764 {}% started the row
4765 {% not started the row

```

Remember that now have started the row:

```

4766 \global\booltrue{\LWR@startedrow}%

```

Create the row tag, with a class if necessary.

```

4767 \global\booltrue{\LWR@intabularmetadata}%
4768 \ifbool{\LWR@doinghline}%
4769 {\LWR@htmltag{tr class="hline"}}{\LWR@orignewline}%
4770 {% not doing hline
4771 \ifbool{\LWR@doingtbrule}%
4772 {\LWR@htmltag{tr class="tbrule"}}{\LWR@orignewline}%
4773 {\LWR@htmltag{tr}}{\LWR@orignewline}%
4774 }% end of not doing hline
4775 }% end of not started the row
4776 }

```

### 55.12 Data opening tag

`\LWR@tabledatasinglecolumn` Print a table data opening tag with style for alignment

```

4777 \newcommand*{\LWR@tabledatasinglecolumnntag}%
4778 {%
4779 \LWR@maybe newtablerow%

```

If have found the end of tabular command, do not create the next data cell:

```

4780 \ifbool{\LWR@exitingtabular}{}%
4781 {% not exiting tabular

```

Fetch the current column's alignment character into \LWR@strresult:

```

4782 \StrChar{\LWR@tablecolspec}{\the\LWR@tablecolspos}[\LWR@strresult]%

```

print the start of a new table data cell:

```

4783 \LWR@htmltag{td class="td%

```

append this column's spec:

```

4784 \LWR@strresult%

```

If this column has a cmidrule, add “rule” to the end of the HTML class tag:

```

4785 \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}{\the\LWR@tablecolspos}}{Y}}{rule}{}%
4786 "{}}%

```

If this is a p, m, b, or X column, allow paragraphs:

```

4787 \ifthenelse{%
4788 \equal{\LWR@strresult}{p}\OR%
4789 \equal{\LWR@strresult}{m}\OR%
4790 \equal{\LWR@strresult}{b}\OR%
4791 \equal{\LWR@strresult}{P}\OR%
4792 \equal{\LWR@strresult}{M}\OR%
4793 \equal{\LWR@strresult}{B}\OR%
4794 \equal{\LWR@strresult}{X}%
4795 }%
4796 {% allow pars
4797 \LWR@startpars%
4798 \global\booltrue{\LWR@tableparcell}%
4799 }% allow pars
4800 {}% no pars

```

Print the @ and ! contents before first column, and then the > contents:

```

4801 \ifthenelse{\cnttest{\value{\LWR@tablecolspos}}=1}%
4802 {%
4803 \LWR@getexparray{\LWR@colatspec}{leftedge}%
4804 \LWR@getexparray{\LWR@colbangspec}{leftedge}%

```

```

4805 }% left edge
4806 {}% not left edge
4807 \LWR@getexparray{LWR@colbeforeSpec}{\theLWR@tablecolspos}%
4808 \global\boolfalse{LWR@intabularmetadata}%
4809 }% not exiting tabular
4810 }%

```

### 55.13 Midrules

**LWR@midrules** LWR@midrules is a data array (section 28) of columns containing Y if a midrule should be created for each column.

**Ctrl LWR@midrulecounter** Indexes across the LWR@midrules data array.

```

4811 \newcounter{LWR@midrulecounter}

```

**\LWR@clearmidrules** Start new midrules. Called at beginning of tabular and also at \.

Clears all LWR@midrules markers for this line.

```

4812 \newcommand*{\LWR@clearmidrules}
4813 {%
4814 \setcounter{LWR@midrulecounter}{1}%
4815 \whiledo{%
4816 \cnttest{\value{LWR@midrulecounter}}{<=}{\value{LWR@tablecolwidth}}%
4817 }%
4818 {%
4819 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{\relax}%
4820 \addtocounter{LWR@midrulecounter}{1}%
4821 }%
4822 }

```

**\LWR@subcmidrule** [*width*] [*trim*] [*leftcolumn*] [*rightcolumn*]

Marks LWR@midrules data array elements to be “Y” from left to right columns.

```

4823 \newcommand*{\LWR@subcmidrule}[4]{%
4824 \setcounter{LWR@midrulecounter}{#3}%
4825 \whiledo{\cnttest{\value{LWR@midrulecounter}}{<=}{#4}}%
4826 {%
4827 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{Y}%
4828 \addtocounter{LWR@midrulecounter}{1}%
4829 }% end of the whiledo
4830 }

```

**\LWR@docmidrule** [*width*] [*trim*] [*leftcolumn-rightcolumn*]

Marks LWR@midrules array elements to be “Y” from left to right columns.

```
4831 \NewDocumentCommand{\LWR@docmidrule}{o d() >{\SplitArgument{1}{-}}m}%
4832 {\LWR@subcmidrule{#1}{#2}#3}
```

## 55.14 Multicolumns

### 55.14.1 Parsing multicolumns

```
4833 \newcounter{LWR@tablemulticolwidth}
4834 \newcounter{LWR@tablemulticolspos}
```

`\LWR@printmccoltype` `{\colspec}` Print any valid column type found. Does not print @, >, or < columns or their associated tokens.

This is printed as part of the table data tag’s class.

```
4835 \newcommand*{\LWR@printmccoltype}[1]{%
4836 \LWR@traceinfo{lwr@printmccoltype -#1-}%
```

Get one token of the column spec:

```
4837 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```
4838 \IfStrEq{\LWR@strresult}{l}{l}{}%
4839 \IfStrEq{\LWR@strresult}{c}{c}{}%
4840 \IfStrEq{\LWR@strresult}{r}{r}{}%
4841 \IfStrEq{\LWR@strresult}{p}{p}{}%
4842 \IfStrEq{\LWR@strresult}{m}{m}{}%
4843 \IfStrEq{\LWR@strresult}{b}{b}{}%
4844 \IfStrEq{\LWR@strresult}{P}{P}{}%
4845 \IfStrEq{\LWR@strresult}{M}{M}{}%
4846 \IfStrEq{\LWR@strresult}{B}{B}{}%
4847 \IfStrEq{\LWR@strresult}{S}{r}{}%
4848 \IfStrEq{\LWR@strresult}{X}{p}{}%
4849 \LWR@traceinfo{lwr@printmccoltype done}%
4850 }
```

`\LWR@multicolpartext` Print the data with paragraph tags:

```
4851 \newcommand*{\LWR@multicolpartext}{%
4852 \LWR@startpars%
4853 \LWR@multicoltext%
4854 \LWR@stoppars%
4855 }
```

`\LWR@multicolother`  $\{\langle colspec \rangle\}$  For @, >, <, print the next token without paragraph tags:

```
4856 \newcommand*{\LWR@multicolother}[1]{%
4857 \addtocounter{\LWR@tablemulticolspos}{1}%
4858 \StrChar{#1}{\the\LWR@tablemulticolspos}[\LWR@strresult]%
4859 \LWR@strresult%
```

A valid column data type was found:

```
4860 \booltrue{\LWR@validtablecol}%
4861 }
```

`\LWR@multicolskip` Nothing to print for this column type.

```
4862 \newcommand*{\LWR@multicolskip}{%
```

A valid column data type was found:

```
4863 \booltrue{\LWR@validtablecol}%
4864 }
```

`\LWR@printmccoldata`  $\{\langle colspec \rangle\}$  Print the data for any valid column type found.

```
4865 \newcommand*{\LWR@printmccoldata}[1]{%
4866 \LWR@traceinfo{\LWR@printmccoldata -#1}%
```

Not yet found a valid column type:

```
4867 \boolfalse{\LWR@validtablecol}%
```

Get one token of the column spec:

```
4868 \StrChar{#1}{\the\LWR@tablemulticolspos}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
4869 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
4870 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
4871 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
4872 \IfStrEq{\LWR@strresult}{D}{}%
4873 \addtocounter{\LWR@tablemulticolspos}{3}% skip parameters
4874 \LWR@multicoltext%
4875 }{}%
4876 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolpartext}{}%
4877 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolpartext}{}%
4878 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolpartext}{}%
```

```

4879 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolpartext}{}%
4880 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolpartext}{}%
4881 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolpartext}{}%
4882 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolpartext}{}%
4883 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolpartext}{}%
4884 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
4885 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@multicolother{#1}}{%
4886 \IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}{%
4887 \IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}{%
4888 \IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}{%

```

If an invalid column type:

```

4889 \ifbool{LWR@validtablecol}{\LWR@multicoltext}%

```

Tracing:

```

4890 \LWR@traceinfo{lwarp@printmccoldata done}%
4891 }

```

`\parsemulticolumnalignment`  $\{\langle 1: \textit{colspec} \rangle\} \{\langle 2: \textit{printresults} \rangle\}$

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.

```

4892 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
4893 \setcounter{LWR@tablemulticolspos}{1}%
4894 \StrLen{#1}[\LWR@strresult]%
4895 \setcounter{LWR@tablemulticolwidth}{\LWR@strresult}%

```

Scan across the tokens in the column spec:

```

4896 \whiledo{%
4897 \not\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolwidth}%
4898 }%
4899 {%

```

Execute the assigned print function for each token in the column spec:

```

4900 #2{#1}%

```

Move to the next token in the column spec:

```

4901 \addtocounter{LWR@tablemulticolspos}{1}%
4902 }%
4903 }

```

### 55.14.2 High-level multicolumn interface

`\LWR@domulticolumn`  $\{\langle 1: numcols \rangle\} \{\langle 2: colspec \rangle\} \{\langle 3: text \rangle\}$

```
4904 \newcommand{\LWR@multicoltext}{%
4905
4906 \NewDocumentCommand{\LWR@domulticolumn}{m m +m}{%
4907 \LWR@traceinfo{lw@domulticolumn -#1- -#2-}%
```

Remember the text to be inserted, and remember that a valid column type was found:

```
4908 \renewcommand{\LWR@multicoltext}{%
4909 #3%
4910 \booltrue{\LWR@validtablecol}%
4911 }%
```

Row processing:

```
4912 \LWR@maybenewtablerow%
```

Begin the opening table data tag:

```
4913 \LWR@htmltag{td colspan="#1"
4914 class="td%
```

Print the column type:

```
4915 \LWR@parsemulticolumnalignment{#2}{\LWR@printmccoltype}%
```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag.

If this position had a “Y” then add “rule”.

```
4916 \ifthenelse{\equal{\LWR@gettexpparray{\LWR@midrules}{\the\LWR@tablecolspos}}{Y}}{rule}{}%
```

Close the class tag’s opening quote:

```
4917 "%
4918 }% end of the opening table data tag
4919 \global\boolfalse{\LWR@intabularmetadata}%
4920 \LWR@parsemulticolumnalignment{#2}{\LWR@printmccoldata}%
4921 }
```

### 55.14.3 Longtable captions

`Bool` `LWR@starredlongtable` Per the caption package, step the counter if `longtable*`.

```
4922 \newbool{LWR@starredlongtable}
4923 \boolfalse{LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
4924 \providecommand*{\LTcapttype}{table}
```

```
\LWR@longtabledatacaptiontag * [(\toc entry)] {\caption}
```

```
4925 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
4926 {%
```

Remember the latest name for \nameref:

```
4927 \IfValueTF{#2}{% optional given?
4928 \ifthenelse{\equal{#2}{}}{% optional empty?
4929 {\LWR@setlatestname{#3}}% empty
4930 {\LWR@setlatestname{#2}}% given and non-empty
4931 }% optional given
4932 {\LWR@setlatestname{#3}}% no optional
```

create a multicolumn across all the columns

```
4933 \LWR@domulticolumn{\theLWR@tabletotalcols}{P}{% \LWR@domulticolumn
4934 % \IfBooleanTF{#1}% star?
4935 % {\IfValueTF{#2}{\LWR@origcaption*{#2}{#3}}{\LWR@origcaption*{#3}}}
4936 % {\IfValueTF{#2}{\LWR@origcaption{#2}{#3}}{\LWR@origcaption{#3}}}
4937 \IfBooleanTF{#1}% star?
```

Star version, show a caption but do not make a LOT entry:

```
4938 {% yes star
4939 \LWR@htmlblocktag{figcaption}%
4940 #3%
4941 \LWR@htmlblocktag{/figcaption}%
4942 }%
4943 {% No star:
```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```
4944 \ifbool{LWR@starredlongtable}%
4945 {%
4946 \ifthenelse{\equal{#2}{}}{% TOC entry
4947 }%
4948 {%
4949 \refstepcounter{\LTcapttype}%
```



```

4950 \protected@edef\@currentlabel{%
4951 \csuse{p@\LTcapttype}\csuse{the\LTcapttype}}%
4952 }%
4953 }{}%

```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```

4954 \LWR@htmlblocktag{figcaption}%
4955 \csuse{fnum@\LTcapttype}\CaptionSeparator#3%
4956 \LWR@htmlblocktag{/figcaption}%

```

See if an optional caption was given:

```

4957 \ifthenelse{\equal{#2}{}}% TOC entry empty

```

if the optional caption was given, but empty, do not form a TOC entry

```

4958 {}%

```

If the optional caption was given, but might only be []:

```

4959 {% TOC entry not empty
4960 \IfNoValueTF{#2}% No TOC entry?

```

The optional caption is []:

```

4961 {% No TOC entry
4962 \addcontentsline%
4963 {\csuse{ext@\LTcapttype}}%
4964 {\LTcapttype}%
4965 {%
4966 \protect\numberline%
4967 {\csuse{p@\LTcapttype}\csuse{the\LTcapttype}}%
4968 {\ignorespaces #3\protect\relax}%
4969 }%
4970 }% end of No TOC entry

```

The optional caption has text enclosed:

```

4971 {% yes TOC entry
4972 \addcontentsline%
4973 {\csuse{ext@\LTcapttype}}%
4974 {\LTcapttype}%
4975 {%
4976 \protect\numberline%
4977 {\csuse{p@\LTcapttype}\csuse{the\LTcapttype}}%
4978 {\ignorespaces #2\protect\relax}%
4979 }%
4980 }% end of yes TOC entry

```

```

4981 }% end of TOC entry not empty
4982 }% end of no star
4983 }% end of \LWR@domulticolumn
4984
4985 \addtocounter{LWR@tablecolspos}{\theLWR@tabletotalcols}
4986 \addtocounter{LWR@tablecolspos}{-1}
4987
4988 }

```

#### 55.14.4 \tabledatamulticolumntag

`\LWR@tabledatamulticolumntag`  $\{\langle numcols \rangle\}$   $\{\langle alignment \rangle\}$   $\{\langle text \rangle\}$

```

4989 \NewDocumentCommand{\LWR@tabledatamulticolumntag}{m m +m}%
4990 {%
4991 \LWR@domulticolumn{#1}{#2}{#3}%
4992 \addtocounter{LWR@tablecolspos}{#1}%
4993 \addtocounter{LWR@tablecolspos}{-1}%
4994 }

```

### 55.15 Multirow

Pkg `multirow`

`\LWR@tabledatamultirowtag`  $\{\langle numrows \rangle\}$   $[\langle bigstruts \rangle]$   $\{\langle width \rangle\}$   $[\langle fixup \rangle]$   $\{\langle text \rangle\}$

```

4995 \NewDocumentCommand{\LWR@tabledatamultirowtag}{m o m o m}%
4996 {%
4997 \LWR@maybenewtablerow%

```

Print the start of a new table data cell:

```

4998 \LWR@htmltag{td rowspan="#1" class="td%

```

Append this column's spec:

```

4999 \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}%

```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag:

```

5000 \ifthenelse{\equal{\LWR@getexparray{LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
5001 "{}}

```

While printing the text, redefine `\` to generate a new line

```

5002 \begingroup \let\\\LWR@endofline #5 \endgroup
5003 \LWR@stoppars%
5004 \global\boolfalse{\LWR@intabularmetadata}%
5005 }%

```

## 55.16 Utility macros inside a table

```

5006 \newcommand*{\LWR@donothing}{}
5007 \newcommand*{\LWR@domidrule}{\booltrue{\LWR@doinghline}}
5008 \newcommand*{\LWR@dotbrule}{\booltrue{\LWR@doingtbrule}}

```

## 55.17 Checking for a new table cell

`\LWR@tabledatacolumnntag` Open a new HTML table cell unless the next token is for a macro which does not create data, such as `\hline`, `\toprule`, etc:

```

5009 \newbool{\LWR@exitingtabular}
5010 \newcommand*{\LWR@tabledatacolumnntag}%
5011 {%

```

`\show\LWR@mynexttoken` to see what tokens to look for

If not any of the below, start a new table cell:

```

5012 \let\mynext\LWR@tabledatasinglecolumnntag%

```

If exiting the tabular:

```

5013 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\end}}{%
5014 {\booltrue{\LWR@exitingtabular}}}%

```

`longtable` can have a caption in a cell

```

5015 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\caption}}{%
5016 {\let\mynext\LWR@donothing}}%

```

Look for other things which would not start a table cell:

```

5017 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multicolumn}}{%
5018 {\let\mynext\LWR@donothing}}%
5019 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multirow}}{%
5020 {\let\mynext\LWR@donothing}}%

```

if come to an `\mrowcell`, this is a cell to be skipped over

```

5021 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\mrowcell}}{%
5022 {\let\mynext\LWR@donothing}{}}%
5023 %
5024 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\hline}}{%
5025 {\let\mynext\LWR@donothing}{}}%
5026 %
5027 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\toprule}}{%
5028 {%
5029 \let\mynext\LWR@donothing}{}}%
5030 %
5031 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\midrule}}{%
5032 {\let\mynext\LWR@donothing}{}}%
5033 %
5034 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cmidrule}}{%
5035 {\let\mynext\LWR@donothing}{}}%
5036 %
5037 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cline}}{%
5038 {\let\mynext\LWR@donothing}{}}%
5039 %
5040 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\bottomrule}}{%
5041 {\let\mynext\LWR@donothing}{}}%
5042 %
5043 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpprintonly}}{%
5044 {\let\mynext\LWR@donothing}{}}%
5045 %
5046 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpHTMLonly}}{%
5047 {\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:


```

5048 \mynext%
5049 }

5050 \end{warpHTML}

```

## 55.18 `\mrowcell`

`\mrowcell` The user must insert `\mrowcell` into any multirow cells which must be skipped.  
 This command has no action during print output.

**for HTML & PRINT:** 5051 `\begin{warpall}`  
5052 `\newcommand*{\mrowcell}{}%`  
5053 `\end{warpall}`

## 55.19 New `\tabular` definition

for HTML output: 5054 `\begin{warpHTML}`

Env `LWR@tabular` [*⟨verticalposition⟩*] {*⟨colspecs⟩*}

The new tabular environment will be `\let` in `\LWR@LwarpStart`, since `siunitx` might redefine `tabular` in the user's document.

```
5055 \newenvironment*{LWR@tabular}[2] []
5056 {%
5057 \LWR@traceinfo{tabular started}%
5058 \begingroup%
5059 \addtocounter{LWR@tabulardepth}{1}%
```

Not yet started a table row:

```
5060 \global\boolfalse{LWR@startedrow}%
```

Not yet doing an hline:

```
5061 \global\boolfalse{LWR@doinghline}%
```

Not yet doing a top/bottom rule:

```
5062 \global\boolfalse{LWR@doingtbrule}%
```

Have not yet found the end of tabular command:

```
5063 \boolfalse{LWR@exitingtabular}%
```

Create the `table` tag:

```
5064 \global\booltrue{LWR@intabularmetadata}%
5065 \LWR@forcenewpage
5066 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
5067 \LWR@parsetablecols{#2}%
```

Table col spec is: `\LWR@tablecolspec` which is a string of `llccrr`, etc.

Do not place the table inside a paragraph:

```
5068 \LWR@stoppars%
```

Track column # for setting text-align:

```
5069 \setcounter{LWR@tablecolspos}{1}%
```

Start looking for midrules:

```
5070 \LWR@clearmidrules%
```

\\ becomes a macro to end the table row:

```
5071 \let\\LWR@tabularendoffline%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with `\LWR@getmynexttoken` to see if the next token might create a new data cell:

```
5072 \renewcommand*{\hline}{\LWR@domidrule\LWR@getmynexttoken}%
5073 \newcommand*{\midrule}{\LWR@domidrule\LWR@getmynexttoken}%
5074 \NewDocumentCommand{\cmidrule}{o d() m}%
5075 {\LWR@docmidrule{##1}(##2){##3}\LWR@getmynexttoken}%
5076 \RenewDocumentCommand{\cline}{m}%
5077 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%
5078 \newcommand*{\toprule}{\LWR@dotbrule\LWR@getmynexttoken}%
5079 \newcommand*{\bottomrule}{\LWR@dotbrule\LWR@getmynexttoken}%
```

The following create data cells and will have no more data in this cell, so we do not want to look ahead for a possible data cell, so do not want to use `\LWR@getmynexttoken`.

```
5080 \let\multicolumnLWR@tabledatamulticolumntag%
5081 \let\multirowLWR@tabledatamultirowtag%
5082 \renewcommand*{\mrowcell}{\global\booltrue{LWR@skippingmrowcell}}%
5083 \let\captionLWR@longtabledatacaptiontag%
```

Reset for new processing:

```
5084 \global\boolfalse{LWR@tableparcell}%
5085 \global\boolfalse{LWR@skippingmrowcell}%
```

Look ahead for a possible table data cell:

```
5086 \LWR@getmynexttoken%
5087 }%
```

Ending the environment:

```
5088 {%
5089 \LWR@closetabledatacell%
```

```

5090 \LWR@htmlblocktag{/tr}%
5091 \LWR@htmlblocktag{/table}%
5092 \global\boolfalse{LWR@intabularmetadata}%

5093 \addtocounter{LWR@tabulardepth}{-1}%
5094 \endgroup%
5095 }

5096 \end{warpHTML}

```

## 55.20 Array

Pkg `array`

`array` is also automatically loaded by `siunitx`.

## 56 Cross-references

Sectioning commands have been emulated from scratch, so the cross-referencing commands are custom-written for them. Emulating both avoids several layers of patches.

The `zref` package is used to remember section name, file, and `lateximage` depth and number for each label.

Table 9 shows the data structures related to cross-referencing.

for HTML output: 5097 \begin{warpHTML}

### 56.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

```
5098 \newcommand*{\@currentlabelname}{}

```

`\LWR@stripperperiod`  $\{\langle text \rangle\} [ \langle . \rangle ]$

Removes a trailing period.

```
5099 \def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%

```

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 56.3), plus \slabel data (section 56.2): <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 56.4.	
<b>\ref and \nameref:</b> Adds HTML tags. See section 56.4.	
<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 41.4 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 65.	
<b>\label inside math:</b> See section 60.4.1.	
<b>Footnotes:</b> See \noteentry in section 41.4.	



`\LWR@setlatestname`  $\{\langle object\ name\rangle\}$

Removes `\label`, strips any final period, and remembers the result.

```
5100 \newcommand*{\LWR@setlatestname}[1]{%
```

Remove `\label` and other commands from the name, the strip any final period.  
See `zref-titleref` and `getttitlestring`.

```
5101 \GetTitleStringExpand{#1}%
5102 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
5103 \edef\@currentlabelname{%
5104 \expandafter\LWR@stripperperiod\@currentlabelname%
5105 \ltx@empty.\ltx@empty\@nil%
5106 }%
5107 }
```

## 56.2 Zref setup

See:

[http://tex.stackexchange.com/questions/57194/  
extract-section-number-from-equation-reference](http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference)

Create a new property list called `special`:

```
5108 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
5109 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
5110 \zref@newprop{zLWR@htmlfilenumber}{%
5111 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilenumber}%
5112 }%
```

Additional properties for `lateximages`:

```
5113 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
5114 \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`:

```

5115 \zref@addprop{special}{zLWR@name}
5116 \zref@addprop{special}{zLWR@htmlfilenumber}
5117 \zref@addprop{special}{zLWR@lateximagedepth}
5118 \zref@addprop{special}{zLWR@lateximagenumber}

```

Returns the selected field:

```

5119 \newcommand*{\LWR@spref}[2]{%
5120 \zref@extractdefault{#1}{#2}{??}}

```

**\LWR@nameref**  $\{\langle label \rangle\}$  Returns the section name for this label:

```

5121 \newcommand*{\LWR@nameref}[1]{%
5122 \LWR@spref{#1}{zLWR@name}%
5123 }

```

**\LWR@htmlfileref**  $\{\langle label \rangle\}$  Returns the file number for this label:

```

5124 \newcommand*{\LWR@htmlfileref}[1]{%
5125 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
5126 \LWR@spref{#1}{zLWR@htmlfilenumber}%
5127 }

```

**\LWR@lateximagedepthref**  $\{\langle label \rangle\}$  Returns the lateximagedepth for this label:

```

5128 \newcommand*{\LWR@lateximagedepthref}[1]{%
5129 \LWR@spref{#1}{zLWR@lateximagedepth}%
5130 }

```

**\LWR@lateximagenumberref**  $\{\langle label \rangle\}$  Returns the lateximagenumber for this label:

```

5131 \newcommand*{\LWR@lateximagenumberref}[1]{%
5132 \LWR@spref{#1}{zLWR@lateximagenumber}%
5133 }

```

**\LWR@splabel**  $\{\langle label \rangle\}$  Sanitize the name and then creates the label:

```

5134 \newcommand*{\LWR@splabel}[1]{%
5135 \LWR@setlatestname{\@currentlabelname}%
5136 \zref@labelbylist{#1}{special}}

```

## 56.3 Labels

**\LWR@subsublabel**  $\{\langle label \rangle\}$  Creates an HTML id tag.

```
5137 \newcommand*{\LWR@subsublabel}[1]{%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```
5138 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}{>}{0}}{%
5139 }{%
5140 {% not lateximage
```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```
5141 \ifbool{\LWR@doingstartpars}{%
5142 {% pars allowed
5143 \ifbool{\LWR@doingapar}{
5144 {% par started
5145 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5146 }% par started
5147 {% par not started
5148 \LWR@stoppars%
5149 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5150 \LWR@startpars%
5151 }% par not started
5152 }% pars allowed
5153 {% pars not allowed
5154 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5155 }% pars not allowed
5156 }% not lateximage
5157 }
```

`\LWR@newlabel`  $\langle label \rangle$  [ $\langle type \rangle$ ]

`\label` during HTML output when not in math mode, removing extra spaces around the label, as done by regular L<sup>A</sup>T<sub>E</sub>X `\label`.

`cleveref` later encases this to add its own cross-referencing.

The optional  $\langle type \rangle$  is per the `ntheorem` package, and is ignored.

```
5158 \NewDocumentCommand{\LWR@newlabel}{m o}{%
5159 \LWR@traceinfo{\LWR@newlabel: starting}%
5160 \LWR@traceinfo{\LWR@newlabel: !#1!}%
5161 % \@bsphack%
```

Create a traditional LaTeX label, as modified by `cleveref`:

```
5162 \LWR@origlabel{#1}%
```

Create a special label which holds the section number, `LWR@htmlfilename`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```

5163 \LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
5164 \LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
5165 \LWR@traceinfo{LWR@newlabel: LWR@htmlfilename is \theLWR@htmlfilename}%
5166 \LWR@splabel{#1}%
5167 \LWR@subsublabel{#1}%
5168 % \@esphack%
5169 \LWR@traceinfo{LWR@newlabel: done}%
5170 }
```

## 56.4 References

`\LWR@startref` `{\label{}}` (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, `#` character, and a label.

```

5171 \newcommand*{\LWR@startref}[1]
5172 {%
5173 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
5174 \LWR@traceinfo{LWR@startref A: !#1!}%
```

Create the filename part of the link:

```

5175 \LWR@htmltag{a href="%
5176 \LWR@traceinfo{LWR@startref B}%
5177 \LWR@htmlrefsectionfilename{#1}%
5178 \LWR@traceinfo{LWR@startref C}%
5179 \#%
```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```

5180 \LWR@traceinfo{LWR@startref D: !#1!}%
5181 \ifthenelse{\equal{\LWR@lidref}{??}}%
```

“??” if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:

```

5182 {%
5183 \LWR@traceinfo{LWR@startref D0: ??}%
5184 ??}%
```

If `LWR@lateximagedepth` is known. Use a `lateximage` if the depth is greater than zero, or a regular link otherwise:

```

5185 {%
5186 \LWR@traceinfo{\LWR@startref D1: \LWR@lidref}%
5187 \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}%
5188 {%
5189 \LWR@traceinfo{\LWR@startref D2: \LWR@lidref}%
5190 lateximage\LWR@lateximagenumberref{#1}%
5191 }%
5192 {%
5193 \LWR@traceinfo{\LWR@startref D3}%
5194 #1%
5195 }%
5196 }%
5197 \LWR@traceinfo{\LWR@startref E}%

```

Closing quote:

```

5198 "}}}%
5199 \LWR@traceinfo{\LWR@startref F}%
5200 }

```

`\LWR@subnewref`  $\{\langle label \rangle\}$   $\{\langle label \text{ or } sub@label \rangle\}$

Factored for the `subfig` package. Uses the original label for the hyper-reference, but prints its own text, such as “1(b)”.

```

5201 \NewDocumentCommand{\LWR@subnewref}{m m}{%
5202 \LWR@traceinfo{\LWR@subnewref #1 #2}%
5203 \LWR@startref{#1}%
5204 \LWR@origref{#2}%
5205 \LWR@htmltag{/a}%
5206 }

```

`\ref` \*  $\{\langle label \rangle\}$  `\ref` is `\let` to `\LWR@newref`

`\LWR@newref` \*  $\{\langle label \rangle\}$  Create an internal document reference link, or without a link if starred per `hyperref`.

```

5207 \NewDocumentCommand{\LWR@newref}{s m}{%
5208 \LWR@traceinfo{\LWR@newref #2}%
5209 \IfBooleanTF{#1}%
5210 {\LWR@origref{#2}}%
5211 {\LWR@subnewref{#2}{#2}}%
5212 }

```

`\pagerefPageFor` Text for starred page references.

```
5213 \newcommand*{\pagerefPageFor}{see }
```

`\pageref` \*  $\langle label \rangle$  Create an internal document reference, or just the unlinked number if starred, per `hyperref`.

```
5214 \NewDocumentCommand{\LWR@newpageref}{s m}{%
5215 \IfBooleanTF{#1}%
5216 {(\pagerefPageFor\LWR@origref{#2})}%
5217 {(\cpageref{#2})}%
5218 }
```

`\nameref`  $\langle label \rangle$

```
5219 \newcommand*{\nameref}[1]{%
5220 \LWR@traceinfo{nameref A}%
5221 \LWR@startref{#1}%
5222 \LWR@traceinfo{nameref B}%
5223 \LWR@nameref{#1}%
5224 \LWR@traceinfo{nameref C}%
5225 \LWR@htmltag{/a}%
5226 \LWR@traceinfo{nameref D}%
5227 }
```

`\Nameref`  $\langle label \rangle$  In print, adds the page number. In HTML, does not.

```
5228 \let\Nameref\nameref
```

## 56.5 Hyper-references



Note that the code currently only sanitizes the underscore character. Additional characters should be rendered inert as well. See the `hyperref.sty` definition of `\gdef\hyper@normalise` for an example.

Pkg `hyperref`



Do not tell other packages that `hyperref` is emulated. Some packages patch various commands if `hyperref` is present, which will probably break something, and the emulation already handles whatever may be emulated anyhow.



Any reference to `\usepackage{hyperref}` must be placed inside a `warpprint` environment.

```
5229 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF:
5230 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Create a link with a text name:

`\LWR@subhyperref {<URL>} {<text>}`

```
5231 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
5232 \LWR@htmltag{a href="#1" target="_{blank}"\LWR@orignewline}#2\LWR@htmltag{/a}%
5233 \LWR@ensuredoingapar%
5234 }
```

`\LWR@subhyperrefclass {<URL>} {<text>} {<htmlclass>}`

```
5235 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
5236 \LWR@htmltag{a href="#1"
5237 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
5238 \LWR@ensuredoingapar%
5239 }
```

`\href [options] {<URL>} {<text>}`

Create a link with accompanying text:

```
5240 \NewDocumentCommand{\LWR@hrefb}{0{} m +m}{%
5241 \LWR@subhyperref{#2}{#3}%
5242 \endgroup%
5243 \LWR@ensuredoingapar%
5244 }
5245
5246 \newcommand{\href}{%
5247 \LWR@ensuredoingapar%
5248 \begingroup%
5249 \catcode'\_ =12
5250 \LWR@hrefb%
5251 }
```

`\nolinkurl {<URL>}`

Print the name of the link without creating the link:

```
5252 \newcommand*{\LWR@nolinkurlb}[1]{#1\endgroup\LWR@ensuredoingapar}
5253
5254 \newcommand{\nolinkurl}{%
5255 \LWR@ensuredoingapar%
5256 \begingroup\catcode'\_ =12
5257 \LWR@nolinkurlb%
5258 }
```

`\url {<URL>}`

Create a link whose text name is the address of the link:

```

5259 \newcommand*{\LWR@urlb}[1]{%
5260 \href{#1}{#1}%
5261 \endgroup%
5262 \LWR@ensuredoingapar%
5263 }
5264
5265 \newcommand{\url}{%
5266 \LWR@ensuredoingapar%
5267 \begingroup\catcode'\_ =12
5268 \LWR@urlb%
5269 }

```

`\LWR@subinlineimage` [*⟨alttag⟩*] [*⟨class⟩*] [*⟨filename⟩*] [*⟨extension⟩*] [*⟨style⟩*]

```

5270 \newcommand*{\LWR@subinlineimage}[5][]{%
5271 \ifthenelse{\equal{#1}{}}{%
5272 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"{}{}}%
5273 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{}{}}%
5274 }

5275 \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.

---

## 57 Floats

Floats are supported, although partially through emulation.

Table 10 shows the data structure associated with each `<type>` of float.

`\@makecaption` is redefined to print the float number and caption text, separated by `\CaptionSeparator`, which works with the `babel` package to adjust the caption separator according to the language. French, for example, uses an en-dash instead of a colon: “Figure 123 – Caption text”.

## 57.1 Float captions

for HTML output: 5276 \begin{warpHTML}

\LWR@floatbegin {<type>} [<placement>]

Begins a \newfloat environment.

```
5277 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
5278 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
5279
5280 === #1 begin
5281
5282 }{}%
5283 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
5284 \addtocounter{LWR@thisfloat}{1}%
5285 \booltrue{LWR@freezethisfloat}%
```

```
5286 \begingroup
```

Settings while inside the environment:

```
5287 \LWR@origraggedright
```

Open an HTML figure tag:

```
5288 \LWR@htmltag{figure id="autofloat-\arabic{LWR@thisfloat}" class="#1"}

5289 \renewcommand*{\@capttype}{#1}
5290 \caption@settype{#1}
5291 \LWR@startpars
5292 }
```

\@float Support packages which create floats directly.  
\@dblfloat

```
5293 \let\@float\LWR@floatbegin
5294 \let\@dblfloat\LWR@floatbegin
```

\LWR@floatend Ends a \newfloat environment.

```
5295 \newcommand*{\LWR@floatend}{%
5296 \LWR@stoppars%
5297 \LWR@htmllementend{figure}%
```

```

5298 \endgroup%
5299 \boolfalse{LWR@freezethisfloat}%
5300 \LWR@startpars%
5301 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
5302
5303 === end
5304
5305 }{}%
5306 }

```

`\end@float` Support packages which create floats directly.  
`\end@dblfloat`

```

5307 \let\end@float\LWR@floatend
5308 \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.

```

5309 \newcounter{LWR@thisfloat}

```

**Bool** `LWR@freezethisfloat` Prevents multiple increments of `\LWR@thisfloat` inside a float.

```

5310 \newbool{LWR@freezethisfloat}
5311 \boolfalse{LWR@freezethisfloat}

```

`\LWR@maybeinthisfloat`

```

5312 \newcommand*{\LWR@maybeinthisfloat}{%
5313 \ifbool{LWR@freezethisfloat}{\addtocounter{LWR@thisfloat}{1}}%
5314 }

```

`\@capttype` Remembers which float type is in use.

```

5315 \newcommand*{\@capttype}{}

```

### 57.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```

5316 \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.

```

5317 \AtBeginDocument{\renewcommand{\@makecaption}[2]{\#1\CaptionSeparator\#2}

```

### 57.1.2 Caption and LOF linking and tracking

When a new HTML file is marked in the L<sup>A</sup>T<sub>E</sub>X PDF file, the L<sup>A</sup>T<sub>E</sub>X page number at that point is stored in `LWR@latestautopage`, (and the associated filename is remembered by the special L<sup>A</sup>T<sub>E</sub>X 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.

```
5318 \newcounter{LWR@nextautofloat}
5319 \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.

```
5320 \newcommand*{\LWRsetnextfloat}[2]{%
5321 \setcounter{LWR@nextautopage}{#1}%
5322 \setcounter{LWR@nextautofloat}{#2}%
5323 }
```

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.

```
5324 \newcounter{LWR@latestautopage}
5325 \setcounter{LWR@latestautopage}{1}

5326 \let\LWR@origcaption@begin\caption@begin
5327 \let\LWR@origcaption@end\caption@end
5328 \let\LWR@orig@@par\@@par
```

`\LWR@caption@begin` Low-level patches to create HTML tags for captions.

```
5329 \newcommand{\LWR@caption@begin}
5330 {
5331 \LWR@traceinfo{LWR@caption@begin}%
```

Keep par and minipage changes local:

```
5332 \begingroup%
```

The caption code was not allowing the closing par tag:

```
5333 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a minipage or \parbox inside the caption:

```
5334 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{-}{-}%
```

```
5335 \RenewDocumentCommand{\parbox}{0{t} o 0{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
5336 \LWR@htmlblocktag{figcaption}%
```

```
5337 \LWR@origcaption@begin%
```

```
5338 }
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
5339 \newcommand{\LWR@caption@end}
```

```
5340 {%
```

```
5341 \LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
5342 \vspace*{\baselineskip}%
```

Closing tag:

```
5343 \LWR@htmlblocktag{/figcaption}%
```

```
5344 \endgroup%
```

```
5345 % \leavevmode% avoid bad space factor (0) error
```

```
5346 \LWR@traceinfo{\LWR@caption@end: done}%
```

```
5347 }
```

`\caption@begin` Low-level patches to create HTML tags for captions.

`\caption@end`

```
5348 \AtBeginDocument{
```

```
5349 \let\caption@begin\LWR@caption@begin
```

```
5350 \let\caption@end\LWR@caption@end
```

```
5351 }
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```

5352 \let\LWR@origcaptionlistentry\captionlistentry
5353
5354 \renewcommand*{\captionlistentry}{%
5355 \LWR@maybeinthisfloat%
5356 \LWR@ensuredoingapar%
5357 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5358 \LWR@origcaptionlistentry%
5359 }
5360
5361 \def\LWR@LTcaptionlistentry{%
5362 \LWR@ensuredoingapar%
5363 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5364 \bgroup
5365 \@ifstar{\egroup\LWR@LT@captionlistentry}% gobble *
5366 {\egroup\LWR@LT@captionlistentry}}%
5367 \def\LWR@LT@captionlistentry#1{%
5368 \caption@listentry\@firstoftwo[LTcaptype]{#1}}%

```

`\addcontentsline` Patched to write the autopage and autofloat before each float's entry. No changes if writing .toc For a theorem, automatically defines `\ext@<type>` as needed, to mimic and reuse the float mechanism.

```

5369 \let\LWR@origaddcontentsline\addcontentsline
5370
5371 \renewcommand*{\addcontentsline}[3]{%
5372 \ifthenelse{\equal{#1}{toc}}{}{%
5373 \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{}
5374 \addtocontents{\@nameuse{ext@#2}}{%
5375 \protect\LWRsetnextfloat%
5376 {\arabic{LWR@latestautopage}}%
5377 {\arabic{LWR@thisfloat}}%
5378 }% addtocontents
5379 }% not toc
5380 \LWR@origaddcontentsline{#1}{#2}{#3}%
5381 }

```

`\captionof` Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

5382 \AtBeginDocument{
5383 \let\LWR@origcaptionof\captionof
5384
5385 \renewcommand*{\captionof}{%
5386 \LWR@maybeinthisfloat%
5387 \LWR@stoppars
5388 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5389 \LWR@origcaptionof%
5390 }

```

```
5391 }
5392 \end{warpHTML}
```

## 58 Table of Contents, LOF, LOT

This section controls the generation of the TOC, LOF, LOT.

The `.toc`, `.lof`, and `.lot` files are named by the source code `\jobname`.

In HTML, the printed tables are placed inside a div of class `.toc`, `.lof`, or `.lot`.

A “sidetoc” is provided which prints a subset of the TOC on the side of each page other than the homepage.

The regular  $\text{\LaTeX}$  infrastructure is used for TOC, along with some patches to generate HTML output.

**for HTML output:** 5393 `\begin{warpHTML}`

### 58.1 Reading and printing the TOC

```
\LWR@myshorttoc {\toc/lof/lot}
```

Reads in and prints the TOC/LOF/LOT at the current position. While doing so, makes the `@` character into a normal letter to allow formatting commands in the section names.

Unlike in regular  $\text{\LaTeX}$ , 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.

```
5394 \newcommand*{\LWR@myshorttoc}[1]{
5395 \LWR@ensuredoingapar
```

Only if the file exists:

```
5396 \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`.

```

5397 \begingroup
5398 % \ifxetexorluatex%
5399 % \else
5400 % \inputencoding{latin1}% currently disabled
5401 % \fi
5402 \makeatletter

```

Read in the TOC file:

```

5403 \@input{\jobname.#1}
5404 % \makeatother
5405 \endgroup
5406 }%
5407 {}%
5408 }

```

`\LWR@subtableofcontents`  $\{\langle toc/lof/lot \rangle\} \{\langle sectionstarname \rangle\}$

Places a TOC/LOF/LOT at the current position.

```

5409 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%

```

Closes previous levels:

```

5410 \@ifundefined{chapter}
5411 {\LWR@closeprevious{\LWR@depthsection}}
5412 {\LWR@closeprevious{\LWR@depthchapter}}

```

Prints any pending footnotes so that they appear above the potentially large TOC:

```

5413 \LWR@printpendingfootnotes

```

Place the list into its own chapter (if defined) or section:

```

5414 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}

```

Create a new HTML nav containing the TOC/LOF/LOT:

```

5415 \LWR@htmlclass{nav}{#1}

```

Create the actual list:

```

5416 \LWR@myshorttoc{#1}

```

Close the nav:

```

5417 \LWR@htmlclassend{nav}{#1}
5418 }

```



Patch `\@starttoc` to encapsulate the TOC inside HTML tags:

```
5419 \let\LWR@orig@starttoc\@starttoc
5420
5421 \renewcommand{\@starttoc}[1]{
5422 \LWR@htmlelementclass{nav}-{#1}
5423 \LWR@orig@starttoc{#1}
5424 \LWR@htmlelementclassend{nav}-{#1}
5425 }
```

Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.

```
5426 \let\LWR@origtableofcontents\tableofcontents
5427 \let\LWR@origlistoffigures\listoffigures
5428 \let\LWR@origlistoftables\listoftables
5429
5430 \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:

```
5431 \ifbool{FormatWordProcessor}{-}{%
```

Copy the `.toc` file to `.sidetoc` for printing the sideTOC. The original `.toc` file is renewed when `\tableofcontents` is finished.

```
5432 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
5433 \LWR@printpendingfootnotes
5434 \LWR@origtableofcontents
5435 }
5436 }
5437 \renewcommand*{\listoffigures}{
5438 \ifbool{FormatWordProcessor}{-}{
5439 \LWR@printpendingfootnotes
5440 \LWR@origlistoffigures
5441 }
5442 }
5443
5444 \renewcommand*{\listoftables}{
5445 \ifbool{FormatWordProcessor}{-}{
5446 \LWR@printpendingfootnotes
5447 \LWR@origlistoftables
5448 }
5449 }
```

## 58.2 High-level TOC commands

`\listof`  $\{\langle type \rangle\}$   $\{\langle title \rangle\}$

Emulate the `\listof` command from the `float` package (section 95). Used to create lists of custom float types. Also used to redefine the standard L<sup>A</sup>T<sub>E</sub>X `\listoffigures` and `\listoftables` commands.

```
5450 \NewDocumentCommand{\listof}{m +m}{%
5451 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
5452 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
5453 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
5454     \jobname.\csuse{ext@#1}\relax
5455 }
```

## 58.3 Side TOC

The “side TOC” is a table-of-contents positioned to the side.

It may be renamed by redefining `\sidetocname`, and may contain paragraphs.

CSS may be used to format the sideTOC:

*CSS related to sideTOC:*

---

**nav.sidetoc:** The entire sideTOC.

**div.sidetoctitle:** The title.

**div.sidetoccontents:** The table of contents.

---

```
5456 \end{warpHTML}
```

**for HTML & PRINT:** 5457 `\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`.

```
5458 \newcounter{SideTOCDepth}
5459 \setcounter{SideTOCDepth}{1}
```

`\sidetocname` Holds the default name for the sideTOC.

```
5460 \newcommand{\sidetocname}{Contents}
```

```
5461 \end{warpall}
```

for HTML output: 5462 \begin{warpHTML}

\LWR@sidetoc Creates the actual side-TOC.

```
5463 \newcommand*{\LWR@sidetoc}{
5464 \LWR@forcenewpage
5465 \LWR@stoppars
5466
```

The entire sideTOC is placed into a nav of class `sidetoc`.

```
5467 \LWR@htmlclass{nav}{sidetoc}
5468
5469 \setcounter{tocdepth}{\value{SideTOCDepth}}
5470
```

The title is placed into a div of class `sidetoctitle`, and may contain paragraphs.

```
5471 \begin{BlockClass}{sidetoctitle}
5472 \sidetocname
5473 \end{BlockClass}
```

The table of contents is placed into a div of class `sidetoccontents`.

```
5474 \begin{BlockClass}{sidetoccontents}
5475 \LinkHome
5476
5477 \LWR@myshorttoc{sidetoc}
5478 \end{BlockClass}
5479 \LWR@htmlclassend{nav}{sidetoc}
5480 }
```

## 58.4 Low-level TOC line formatting

\numberline  $\{\langle number \rangle\}$

(Called from each line in the `.aux`, `.lof` files.)

Record this section number for further use:

```
5481 \renewcommand*{\numberline}[1]{%
5482 \LWR@sectionnumber{#1}%
5483 }
```

`\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

```
5484 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
5485 \ifthenelse{\cnttest{#1}{<=}}{\value{tocdepth}}{%
```

```
5486 \LWR@startpars%
```

Create an HTML link to `filename#autosec-(page)`, with text of the caption, of the given HTML class.

```
5487 \LWR@subhyperrefclass{%
```

```
5488 \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
```

```
5489 \LWR@stoppars%
```

```
5490 }
```

```
5491 {}
```

```
5492 }
```

**Ctrl** `lofdepth` TOC depth for figures.

```
5493 \newcounter{lofdepth}
```

```
5494 \setcounter{lofdepth}{1}
```

**Ctrl** `lotdepth` TOC depth for tables.

```
5495 \newcounter{lotdepth}
```

```
5496 \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

```
5497 \newcommand{\hypertocfloat}[5]{%
5498 \LWR@startpars
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
5499 \@ifundefined{c@#3depth}{%
5500 \newcounter{#3depth}%
5501 \setcounter{#3depth}{1}%
5502 }{}%
```

Respond to `lofdepth`, etc.:

```
5503 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
5504 \ifthenelse{\cnttest{#1}{<=}}{\arabic{#3depth}}{}%
5505 \LWR@startpars%
```

Create an HTML link to `filename#autofloat-(float number)`, with text of the caption, of the given HTML class.

```
5506 \LWR@subhyperrefclass{%
5507 \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
5508 \#autofloat-\arabic{LWR@nextautofloat}}%
5509 {#4}{toc#2}%
5510 \LWR@stoppars%
5511 }{}%
5512 }
```

Automatically called by `\contentsline`:

```
5513 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
5514 \DeclareDocumentCommand{\l@chapter}{m m}
5515 {\hypertoc{0}{chapter}{#1}{#2}}
5516 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
5517 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
5518 \renewcommand{\l@subsubsection}[2]
5519 {\hypertoc{3}{subsubsection}{#1}{#2}}
5520 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
5521 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
5522 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
5523 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

5524 \end{warpHTML}
```

## 59 Index and glossary

See:

[http://tex.stackexchange.com/questions/187038/  
how-to-mention-section-number-in-index-created-by-imakeidx](http://tex.stackexchange.com/questions/187038/how-to-mention-section-number-in-index-created-by-imakeidx)

Index links are tracked by the counter `LWR@autoindex`. This counter is used to create a label for each index entry, and a reference to this label for each entry in the index listing. This method allows each index entry to link directly to its exact position in the document.

for HTML output: 5525 `\begin{warpHTML}`

```
5526 \newcounter{LWR@autoindex}
5527 \setcounter{LWR@autoindex}{0}
5528
5529 \newcounter{LWR@autoglossary}
5530 \setcounter{LWR@autoglossary}{0}
```

`\LWR@indexsection` Controls whether the index will be in a section or a chapter, depending on the documentclass.

```
5531 \@ifundefined{chapter}
5532 {\newcommand*{\LWR@indexsection}{\section{\indexname}}}
5533 {\newcommand*{\LWR@indexsection}{\chapter{\indexname}}}
```

`\printindex`

```
5534 \let\LWR@origprintindex\printindex
5535
5536 \renewcommand*{\printindex}
5537 {
5538 \LWR@indexsection
5539 \LWR@startpars
5540 \LWR@origprintindex
5541 }
```

Env `theindex`

```
5542 \renewenvironment*{theindex}{%
5543 \let\item\LWR@indexitem%
5544 \let\subitem\LWR@indexsubitem%
5545 \let\subsubitem\LWR@indexsubsubitem%
5546 }{}
```

`\LWR@indexitem`

```

5547 \newcommand{\LWR@indexitem}{
5548
5549 \InlineClass@indexitem}{}
5550 }

```

`\LWR@indexitem`

```

5551 \newcommand{\LWR@indexsubitem}{
5552
5553 \InlineClass@indexsubitem}{}
5554 }

```

`\LWR@indexitem`

```

5555 \newcommand{\LWR@indexsubsubitem}{
5556
5557 \InlineClass@indexsubsubitem}{}
5558 }

```

`\@wrindex {<term>}`      Redefined to write the `LWR@latestautopage` counter instead of `page`

```

5559 \def\@wrindex#1{%
5560 \addtocounter{LWR@autoindex}{1}%
5561 \LWR@newlabel{LWRindex-\theLWR@autoindex}%
5562 \protected@write\@indexfile{%
5563 {\string\indexentry{#1}{\theLWR@autoindex}}%
5564 \endgroup
5565 \@esphack}

```

`\@wrglossary {<term>}`      Redefined to write the `LWR@latestautopage` counter instead of `page`

```

5566 \def\@wrglossary#1{%
5567 \addtocounter{LWR@autoglossary}{1}%
5568 \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
5569 \protected@write\@glossaryfile{%
5570 {\string\glossaryentry{#1}{\theLWR@autoglossary}}%
5571 \endgroup
5572 \@esphack}

```

`\hyperindexref {<autosecnumber>}`

`\hyperindexref{web address}` is inserted into `*.ind` by the `xindy` style file `lwarp.xdy`

```

5573 \newcommand*{\hyperindexref}[1]{\nameref{LWRindex-#1}}

```

```
5574 \end{warpHTML}
```

**for PRINT output:** A null command for print mode, in case `hyperref` was not used:

```
5575 \begin{warpprint}
5576 \newcommand{\hyperindexref}[1]{#1}
5577 \end{warpprint}
```

## 60 Math

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

**SVG math option** For SVG math, math is rendered as usual by L<sup>A</sup>T<sub>E</sub>X 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 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 tag 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 7 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**    <sup>9</sup>See section 165 regarding fonts and fractions.



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, `\tagged` equations have the chapter number prepended in HTML output, unlike  $\text{\LaTeX}$ . `\tag*` equations (correctly) do not. This may be improved with future versions of the MathJax support script.

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

#### subequations

- MathJax itself does not support subequations. This may be improved by parsing the  $\text{\LaTeX}$  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 extension. While inside a math expression, do not use `\SI` or `\si` inside `\text`, where it will be rendered as normal text.

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

 `siunitx` inside an equation

#### $\text{\LaTeX}$ macros

- MathJax does not automatically support custom  $\text{\LaTeX}$  macros, but they may be set up by the user.

#### custom MathJax macros

As an example of using custom  $\text{\LaTeX}$  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: 5578 `\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.

```

5579 \let\LWR@origtextdollar\$
5580
5581 \renewcommand*{\$}{%
5582 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{
5583 {\LWR@origtextdollar}%
5584 {\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
5585 }

```

Ctr LWR@externalfilecnt Counter for the external files which are generated and then referenced from the HTML:

```

5586 \newcounter{LWR@externalfilecnt}

```

## 60.1 Inline and display math

```

5587 \let\LWR@origdollar=$
5588 \let\LWR@secondorigdollar=$% balance for editor syntax highlighting

5589 \let\LWR@origopenparen\(
5590 \let\LWR@origcloseparen\)
5591 \let\LWR@origopenbracket\[
5592 \let\LWR@origclosebracket\]

```

`$` Redefine the dollar sign to place math inside a `lateximage`, or use MathJax:  
`$$`

```

5593 \begingroup
5594 \catcode'\$=\active%
5595 \protected\gdef$ {\ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

`\LWR@doubledollar` Redefine the double dollar sign to place math inside a `lateximage`, or use MathJax:

```
5596 \gdef\LWR@doubledollar$#1$${
5597 \ifbool{mathjax}%
```

For MathJax, print the math between `\[` and `\]`:

```
5598 {\textbackslash[\LWR@HTMLsanitize{#1}\textbackslash]}
```

For SVG, print the math inside a `lateximage`, with an ALT tag of the  $\LaTeX$  code:

```
5599 {% not mathjax
5600
5601 \begin{lateximage}%
5602 [\textbackslash{[] \LWR@HTMLsanitize{#1} \textbackslash{[]}]%
5603 \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
5604 \end{lateximage}
5605
5606 }
5607 }%
```

`\LWR@singledollar` Redefine the single dollar sign to place math inside a `lateximage`, or use MathJax:

```
5608 \gdef\LWR@singledollar#1${%
5609 \ifbool{mathjax}%
```

For MathJax, print the math between `\(` and `\)`:

```
5610 {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}
```

For SVG, print the math inside a `lateximage`, with an ALT tag of the  $\LaTeX$  code:

```
5611 {% not mathjax
5612 \begin{lateximage}%
5613 [\textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)]%
5614 \LWR@origdollar#1\LWR@origdollar%
5615 \end{lateximage}%
5616 }%
5617 }%
```

`\(` Redefine to the above dollar macros.

`\)`

```
5618 \gdef\(#1\){$#1$}
5619 \gdef\[#1\]{$$#1$$}
5620
5621 \endgroup
```

Remove the old `math` and `displaymath` environments:

```
5622 \let\math\relax
5623 \let\endmath\relax
5624 \let\displaymath\relax
5625 \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`.

```
5626 \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`.

```
5627 \NewEnviron{displaymath}{\expandafter\[\BODY\]\@ignoretrue}
```

When the document begins, the dollar sign must be made active to trigger the new math macros:

```
5628 \AtBeginDocument{\catcode'\$=\active}
```

## 60.2 MathJax support

Ctr `LWR@nextequation` Used to add one to compute the next equation number.

```
5629 \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.

```
5630 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
5631 \ifcsdef{thechapter}{
5632 \BlockClassSingle{hidden}{
5633 \textbackslash(
5634 \textbackslash{}seteqsection \{\thechapter\}
5635 \textbackslash)
5636 }
```

```

5637 }
5638 {}% not using chapters
5639

```

MathJax doesn't allow setting the equation number to 1:

```

5640 \ifthenelse{\cnttest{\value{equation}}>0}
5641 {

```

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.

```

5642 \setcounter{LWR@nextequation}{\value{equation}}
5643 \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.

```

5644 \BlockClassSingle{hidden}{
5645 \textbackslash(
5646 \textbackslash)seteqnumber \{\arabic{LWR@nextequation}\}
5647 \textbackslash)
5648 }
5649 }{}% not eq > 1
5650 }

```

`\LWR@restoremathlatexformatting` While producing math, use regular L<sup>A</sup>T<sub>E</sub>X formatting instead of HTML tags.

```

5651 \newcommand*{\LWR@restoremathlatexformatting}{%
5652 \let\hspace\LWR@origspace%
5653 \let\rule\LWR@origrule%
5654 \let\,\LWR@origcomma% disable HTML short unbreakable space
5655 \let\textit\LWR@origtextit%
5656 \let\textbf\LWR@origtextbf%
5657 \let\texttt\LWR@origtexttt%
5658 \let\textsc\LWR@origtextsc%
5659 \let\textsf\LWR@origtextsf%
5660 \let\textrm\LWR@origtextrm%
5661 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
5662 \let\textsuperscript\LWR@origtextsuperscript%
5663 \let\textsubscript\LWR@origtextsubscript%
5664 \let~\LWR@origtilde%
5665 \let\enskip\LWR@origenskip%
5666 \let\quad\LWR@origquad%
5667 \let\qquad\LWR@origqquad%
5668 }

```

```

\LWR@hidelatexequation {\(environment)\} {\(contents)\}

```

Creates the L<sup>A</sup>T<sub>E</sub>X version of the equation inside an HTML comment.

```
5669 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```
5670 \LWR@stoppars
5671 \LWR@htmlopencomment
5672
```

Start the L<sup>A</sup>T<sub>E</sub>X math environment inside the HTML comment:

```
5673 \begingroup
5674 \csuse{\LWR@orig#1}
```

While in the math environment, restore various commands to their L<sup>A</sup>T<sub>E</sub>X meanings.

```
5675 \LWR@restoremathlatexformatting
```

See `\LWR@htmlmathlabel` in section [60.4.1](#).

Print the contents of the equation:

```
5676 #2
```

End the L<sup>A</sup>T<sub>E</sub>X math environment inside the HTML comment:

```
5677 \csuse{\LWR@origend#1}
5678 \endgroup
5679
```

Close the HTML comment and resume HTML paragraph handling:

```
5680 \LWR@htmlclosecomment
5681 \LWR@startpars
5682 }
```

```
\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.

```
5683 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
5684
```

Enclose the MathJax environment inside printed “\(" and “\)” characters.

```
5685 \textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
5686 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
5687 \textbackslash{}end\{#1\}
5688
5689 }
```

### 60.3 Equation environment

Remember existing `equation` environment:

```
5690 \let\LWR@origequation\equation
5691 \let\LWR@origendequation\endequation
```

Remove existing `equation` environment:

```
5692 \let\equation\relax
5693 \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.

```
5694 \NewEnviron{equation}{%
5695
5696 \ifbool{mathjax}
```

MathJax output:

```
5697 {
```

Print commands to synchronize MathJax's equation number and format to the current  $\LaTeX$  chapter/section and equation number:

```
5698 \LWR@syncmathjax
```

Print the L<sup>A</sup>T<sub>E</sub>X math inside an HTML comment:

```
5699 \LWR@hidelatexequation{equation}{\BODY}
5700 }
```

SVG output: Create the `lateximage` along with an HTML ALT tag having an equation number, the L<sup>A</sup>T<sub>E</sub>X equation environment commands, and the contents of the environment's `\BODY`.

```
5701 {% not mathjax
```

Begin the `lateximage` with an ALT tag containing the math source:

```
5702 \begin{lateximage}[(\theequation) \textbackslash{begin}\{equation\}}{%
5703 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}}}%
5704 \textbackslash{end}\{equation\}}}% alt tag
```

Create the actual L<sup>A</sup>T<sub>E</sub>X-formatted equation inside the `lateximage` using the contents of the environment.

```
5705 \LWR@origequation
5706 \BODY% contents collected by NewEnviron
5707 \LWR@origendequation
5708 \end{lateximage}%
5709 }
5710
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
5711 }[\ifbool{mathjax}{\LWR@addmathjax{equation}{\BODY}}{}}]
```

## 60.4 AMS Math environments

### 60.4.1 Support macros

Bool `LWR@amsmultline` True if processing a multiline environment.

To compensate for `multiline`-specific code, `LWR@amsmultline` is used to add extra horizontal space in `\LWR@htmlmathlabel` if is used in an `amsmath` environment which is not a `multiline` environment and not an `equation`.

```
5712 \newbool{LWR@amsmultline}
5713 \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`.

```
5714 \newcommand*\LWR@htmlmathlabel}[1]{%
5715 \LWR@traceinfo{\LWR@htmlmathlabel #1}%
5716 \ifbool{mathjax}{%
```

The combined  $\text{\LaTeX}$  & HTML label is printed in a `\text` field:

```
5717 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
5718 \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.)

```
5719 \LWR@htmlclosecomment%
5720 \LWR@origltx@label{#1}%
5721 \LWR@htmlopencomment%
5722 }% text
5723 }% mathjax
5724 {%
5725 \LWR@origltx@label{#1}%
5726 }%
5727 }
```

`\LWR@beginhideamsmath` Starts hiding  $\text{\LaTeX}$  math inside an HTML comment.

```
5728 \newcommand*\LWR@beginhideamsmath){
5729 \LWR@stoppars
5730
5731 \LWR@htmlopencomment
5732
5733 \begingroup
5734 \LWR@restoremathlatexformatting
5735 }
```

`\LWR@endhideamsmath` Ends hiding  $\text{\LaTeX}$  math inside an HTML comment.

```

5736 \newcommand*{\LWR@endhideamsmath}{
5737 \endgroup
5738
5739 \LWR@htmlclosecomment
5740
5741 \LWR@startpars
5742 }

```

### 60.4.2 Environment patches

The following `amsmath` environments already collect their contents in `\@envbody` for further processing.

For SVG math: Each environment is encapsulated inside a `lateximage` environment, along with a special `LWRAMSMATHBODY` argument telling `lateximage` to use as the HTML ALT tag the environment's contents which were automatically captured by the  $\mathcal{AMS}$  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 `multline`

```

5743 \BeforeBeginEnvironment{multline}{
5744 \ifbool{mathjax}
5745 {
5746 \LWR@syncmathjax
5747 \booltrue{LWR@amsmultline}
5748 \LWR@beginhideamsmath
5749 }
5750 {
5751 \lateximage[LWRAMSMATHBODY]
5752 }
5753 }
5754
5755 \AfterEndEnvironment{multline}{
5756
5757 \ifbool{mathjax}
5758 {
5759 \LWR@endhideamsmath
5760 \boolfalse{LWR@amsmultline}
5761 \LWR@addmathjax{multline}{\the\@envbody}
5762 }
5763 {\endlateximage}
5764
5765 }

```

Env **multline\***

```

5766 \BeforeBeginEnvironment{multline*}{
5767 \ifbool{mathjax}
5768 {
5769 \LWR@syncmathjax
5770 \booltrue{LWR@amsmultline}
5771 \LWR@beginhideamsmath
5772 }
5773 {
5774 \lateximage[LWRMSMATHBODY]
5775 }
5776 }
5777
5778 \AfterEndEnvironment{multline*}{
5779
5780 \ifbool{mathjax}
5781 {
5782 \LWR@endhideamsmath
5783 \boolfalse{LWR@amsmultline}
5784 \LWR@addmathjax{multline*}{\the\@envbody}
5785 }
5786 {\endlateximage}
5787
5788 }
5789

```

Env **gather**

```

5790 \BeforeBeginEnvironment{gather}{
5791 \ifbool{mathjax}
5792 {
5793 \LWR@syncmathjax
5794 \boolfalse{LWR@amsmultline}
5795 \LWR@beginhideamsmath
5796 }
5797 {
5798 \lateximage[LWRMSMATHBODY]
5799 }
5800 }
5801
5802 \AfterEndEnvironment{gather}{
5803
5804 \ifbool{mathjax}
5805 {
5806 \LWR@endhideamsmath
5807 \LWR@addmathjax{gather}{\the\@envbody}
5808 }
5809 {\endlateximage}

```

```
5810
5811 }
```

Env **gather\***

```
5812 \BeforeBeginEnvironment{gather*}{
5813 \ifbool{mathjax}
5814 {
5815 \LWR@syncmathjax
5816 \boolfalse{LWR@amsmultline}
5817 \LWR@beginhideamsmath
5818 }
5819 {
5820 \lateximage[LWR@SMATHBODY]
5821 }
5822 }
5823
5824 \AfterEndEnvironment{gather*}{
5825
5826 \ifbool{mathjax}
5827 {
5828 \LWR@endhideamsmath
5829 \LWR@addmathjax{gather*}{\the\@envbody}
5830 }
5831 {\endlateximage}
5832
5833 }
```

Env **align**

```
5834 \BeforeBeginEnvironment{align}{
5835 \ifbool{mathjax}
5836 {
5837 \LWR@syncmathjax
5838 \boolfalse{LWR@amsmultline}
5839 \LWR@beginhideamsmath
5840 }
5841 {
5842 \lateximage[LWR@SMATHBODY]
5843 }
5844 }
5845
5846 \AfterEndEnvironment{align}{
5847
5848 \ifbool{mathjax}
5849 {
5850 \LWR@endhideamsmath
5851 \LWR@addmathjax{align}{\the\@envbody}
5852 }
```

```

5853 {\endlateximage}
5854
5855 }

```

Env **align\***

```

5856 \BeforeBeginEnvironment{align*}{
5857 \ifbool{mathjax}
5858 {
5859 \LWR@syncmathjax
5860 \boolfalse{LWR@amsmultline}
5861 \LWR@beginhideamsmath
5862 }
5863 {
5864 \lateximage[LWR@SMATHBODY]
5865 }
5866 }
5867
5868 \AfterEndEnvironment{align*}{
5869
5870 \ifbool{mathjax}
5871 {
5872 \LWR@endhideamsmath
5873 \LWR@addmathjax{align*}{\the\@envbody}
5874 }
5875 {\endlateximage}
5876
5877 }

```

Env **flalign**

```

5878 \BeforeBeginEnvironment{flalign}{
5879 \ifbool{mathjax}
5880 {
5881 \LWR@syncmathjax
5882 \boolfalse{LWR@amsmultline}
5883 \LWR@beginhideamsmath
5884 }
5885 {
5886 \lateximage[LWR@SMATHBODY]
5887 }
5888 }
5889
5890 \AfterEndEnvironment{flalign}{
5891
5892 \ifbool{mathjax}
5893 {
5894 \LWR@endhideamsmath
5895 \LWR@addmathjax{flalign}{\the\@envbody}

```

```

5896 }
5897 {\endlateximage}
5898
5899 }

```

Env `flalign*`

```

5900 \BeforeBeginEnvironment{flalign*}{
5901 \ifbool{mathjax}
5902 {
5903 \LWR@syncmathjax
5904 \boolfalse{LWR@amsmultline}
5905 \LWR@beginhideamsmath
5906 }
5907 {
5908 \lateximage[LWR@SMATHBODY]
5909 }
5910 }
5911
5912 \AfterEndEnvironment{flalign*}{
5913
5914 \ifbool{mathjax}
5915 {
5916 \LWR@endhideamsmath
5917 \LWR@addmathjax{flalign*}{\the\@envbody}
5918 }
5919 {\endlateximage}
5920
5921 }

5922 \end{warpHTML}

```

## 61 Lateximages

A `\lateximage` is typeset on its own PDF page inside an HTML comment which starts on the preceeding page and ends on following page, and instructions are written to `lateximage.txt` for `lwarpmk` to extract the `\lateximage` from the page of the PDF file then generate an accompanying `.svg` file image file. Meanwhile, instructions to show this image are placed into the HTML file after the comment.

An HTML span is created to hold both the HTML comment, which will have the `pdftotext` conversion, and also the link to the final `.svg` image.

A  $\LaTeX$  label is used to remember which PDF page has the image. A label is used because footnotes, endnotes, and pagenotes may cause the image to appear

at a later time. The label is declared along with the image, and so it correctly remembers where the image finally ended up.

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

**for HTML output:** 5923 `\begin{warpHTML}`

Ctr `LWR@lateximagenumber` Sequence the images.

```
5924 \newcounter{LWR@lateximagenumber}
5925 \setcounter{LWR@lateximagenumber}{0}
```

Ctr `LWR@lateximagedepth` Do not create `\lateximage` inside of `\lateximage`.

```
5926 \newcounter{LWR@lateximagedepth}
5927 \setcounter{LWR@lateximagedepth}{0}
```

Declare the `\LWR@file` for writing to generate file `lateximages.txt`:

```
5928 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
```

A few utility macros to write special characters:

```
5929 \edef\LWR@hashmark{\string#} % for use in \write
5930 \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`.

```
5931 \newcounter{LWR@Lipage}
```

```
5932 \end{warpHTML}
```

**for HTML & PRINT:** 5933 `\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.*

```
5934 \newcommand*{\LateximageFontSizeName}{large}
```

```
5935 \end{warpall}
```

**for HTML output:** 5936 `\begin{warpHTML}`

`\LWR@HTMLsanitize`  $\{\langle text \rangle\}$

Math expressions are converted to `lateximages`, and some math environments may contain “&”, “<”, or “>”, which should not be allowed inside an HTML ALT tag, so must convert them to HTML entities.

Two versions follow, depending on expansion needs. There may be a better way...

```

5937 \newcommand{\LWR@HTMLsanitize}[1]{%
5938 \protect\StrSubstitute{\detokenize{#1}}{%
5939 {\detokenize{&}}{%
5940 {\detokenize{&}}}{\LWR@strresult}%
5941 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
5942 {\detokenize{<}}{%
5943 {\detokenize{<}}}{\LWR@strresult}%
5944 [\LWR@strresult]%
5945 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
5946 {\detokenize{>}}{%
5947 {\detokenize{>}}}{\LWR@strresult}%
5948 [\LWR@strresult]%
5949 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
5950 {\detokenize{##}}{%
5951 {\#}}{\LWR@strresult}%
5952 [\LWR@strresult]%
5953 \LWR@strresult%
5954 }

```

`\LWR@HTMLsanitizeexpand`  $\{\langle text \rangle\}$

This version expands the argument before sanitizing it.

```

5955 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
5956 \protect\StrSubstitute{\detokenize\expandafter{#1}}{%
5957 {\detokenize{&}}{%
5958 {\detokenize{&}}}{\LWR@strresult}%
5959 [\LWR@strresult]%
5960 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
5961 {\detokenize{<}}{%
5962 {\detokenize{<}}}{\LWR@strresult}%
5963 [\LWR@strresult]%
5964 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
5965 {\detokenize{>}}{%
5966 {\detokenize{>}}}{\LWR@strresult}%
5967 [\LWR@strresult]%
5968 \LWR@strresult%
5969 }

```

Env `lateximage`  $[\langle alttag \rangle]$



```

5970 \NewDocumentEnvironment{lateximage}{0{image}}{%
5971 \LWR@traceinfo{lateximage: starting on page \arabic{page}}%
5972 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%

```

If nesting inside an already-existing lateximage, simply record one more level:

```

5973 {%
5974 \addtocounter{LWR@lateximagedepth}{1}%
5975 }%

```

Otherwise, this is the outer-most lateximage:

```

5976 {% start of outer-most lateximage

```

Starting a new lateximage:

```

5977 \addtocounter{LWR@lateximagenumber}{1}%
5978 \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%

```

While inside a lateximage, do not use mathjax:

```

5979 \boolfalse{mathjax}

```

Be sure that are doing a paragraph:

```

5980 \LWR@ensuredoingapar%

```

Next file:

```

5981 \addtocounter{LWR@externalfilecnt}{1}%
5982 \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%

```

Figure out what the next page number will be:

```

5983 \setcounter{pageref}{LWR@Lipage}{LWR@lateximage\theLWR@lateximagenumber}%
5984 \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:

```

5985 \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
5986 class="lateximagesource">{}} \LWR@orignewline

```

Write instructions to the `lateximages.txt` file:

```

5987 \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.

```
5988 \LWR@htmlopencomment%
5989 \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
5990 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
5991 \LWR@origminipage{6in}%
5992 \csuse{LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for \hspace, etc. inside a lateximage.

```
5993 \let\hspace\LWR@orighspace%
5994 \let\rule\LWR@origrule%
5995 \let\,\LWR@origcomma% disable HTML short unbreakable space
5996 \let\textit\LWR@origtextit%
5997 \let\textbf\LWR@origtextbf%
5998 \let\texttt\LWR@origtexttt%
5999 \let\textsc\LWR@origtextsc%
6000 \let\textsf\LWR@origtextsf%
6001 \let\textrm\LWR@origtextrm%
6002 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
6003 \let\textsuperscript\LWR@origtextsuperscript%
6004 \let\textsubscript\LWR@origtextsubscript%
6005 \let~\LWR@origtilde%
6006 \let\enskip\LWR@origenskip%
6007 \let\quad\LWR@origquad%
6008 \let\qquad\LWR@origqquad%
6009 \let\tabular\LWR@origtabular%
6010 \let\endtabular\LWR@origendtabular%
6011 \let\newline\LWR@orignewline%
6012 \LWR@origlabel{LWR@lateximage\arabic{LWR@lateximagenumber}}}%
6013 }% end of outer-most lateximage
6014 }% end of \begin{lateximage}
6015 {% start of \end{lateximage}
6016 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
6017 {%
6018 \addtocounter{LWR@lateximagedepth}{-1}%
6019 }%
```

if this is the outer-most lateximage:

```
6020 {% end of outer-most lateximage
```

Finish the lateximage minipage and start a new PDF page:

```
6021 \LWR@origendminipage%
6022 \LWR@orignewpage%
6023 \LWR@origscriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by `pdftotext`:

```
6024 \LWR@htmlclosecomment{}\LWR@orignewline%
6025 \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.

```
6026 \ifthenelse{\equal{#1}{LWRAMSMATHBODY}}{%
6027 {%
6028 \LWR@subinlineimage[%
6029 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}}%
6030 ]%
6031 {lateximage}%
6032 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}%
6033 {svg}%
6034 {}%
6035 }%
6036 {%
6037 \LWR@subinlineimage[#1]{lateximage}%
6038 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}{svg}{}%
6039 }%
6040 % \LWR@orignewline% Removed to prevent extra space.
```

Be sure that are doing a paragraph:

```
6041 \LWR@ensuredoingapar%
```

Close the HTML span which has the `pdftotext` comment and also the link to the `.svg` image:

```
6042 \LWR@htmltag{/span}%
6043 \ifbool{HTMLDebugComments}{%
6044 \LWR@htmlcomment{End of lateximage}%
6045 }{}%
6046 % \LWR@orignewline% Removed to prevent extra space.
```

Undo one lateximage level:

```
6047 \addtocounter{LWR@lateximagedepth}{-1}%
6048 }% end of outer-most lateximage
6049 \LWR@traceinfo{lateximage: done}
6050 }%
6051 \end{warpHTML}
```

for PRINT output: 

```
6052 \begin{warpprint}
6053 \newenvironment{lateximage}[1][\minipage{\linewidth}]{\endminipage}
6054 \end{warpprint}
```

## 62 center, flushleft, flushright

for HTML output: 

```
6055 \begin{warpHTML}
```

Env **center** Replace center functionality with CSS tags:

```
6056 \renewenvironment*{center}
6057 {
6058 \LWR@forcenewpage
6059 \BlockClass{center}
6060 }
6061 {\endBlockClass}
```

Env **flushright**

```
6062 \renewenvironment*{flushright}
6063 {
6064 \LWR@forcenewpage
6065 \BlockClass{flushright}
6066 }
6067 {\endBlockClass}
```

Env **flushleft**

```
6068 \renewenvironment*{flushleft}
6069 {
6070 \LWR@forcenewpage
6071 \BlockClass{flushleft}
6072 }
6073 {\endBlockClass}
```

```
6074 \end{warpHTML}
```

## 63 Siunitx

Pkg **siunitx**

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

for HTML output: 6075 `\begin{warpHTML}`

Options for siunitx:

```
6076 \PassOptionsToPackage{
6077 detect-mode=true,
6078 per-mode=symbol,% fraction is not seen by pdftotext
6079 text-celsius = {\HTMLentity{deg}C},
6080 text-degree = {\HTMLentity{deg}},
6081 }{siunitx}

6082 \end{warpHTML}
```

## 64 Graphics

Pkg **graphics**

Pkg **graphicx**

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

⚠ **viewports**

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

**units** For `\includegraphics`, avoid `px` and `%` units for width and height, or enclose them inside `warpHTML` environments. For font-proportional image sizes, use `ex` or `em`. For fixed-sized images, use `cm`, `mm`, `in`, `pt`, or `pc`. Using the keys `width=.5\linewidth`, or similar for `\textwidth` or `\textheight` to give fixed-sized images proportional to a 6 by 9 inch text area.

**options** `\includegraphics` accepts `width` and `height`, `origin`, `rotate` and `scale`, plus a new `class` key.

**HTML class** With HTML output, `\includegraphics` accepts an optional `class=xyz` keyval combination, and if this is given then the HTML output will include that class for the image. The class is ignored for print output.

⚠ image file types

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

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

⚠ browser support

`\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The CSS3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike L<sup>A</sup>T<sub>E</sub>X, so expect some ugly results for scaling and rotating.

for HTML output: 6083 `\begin{warpHTML}`

## 64.1 `\graphicspath`

`\graphicspath` `{\path}`

6084 `\newcommand*\thisgraphicspath{\{}}`

6085 `\renewcommand*\graphicspath[1]{\renewcommand*\thisgraphicspath{\#1}}`

`\DeclareGraphicsExtensions` `{\list}`

`\DeclareGraphicsRule` `{\langle}` `{\rangle}` `{\langle}` `{\rangle}`

6086 `\renewcommand*\DeclareGraphicsExtensions[1]{}`

6087 `\renewcommand*\DeclareGraphicsRule[4]{}`

## 64.2 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.

6088 `\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.

6089 `\newlength{\LWR@igwidth}`

6090 `\newlength{\LWR@igheight}`

```

6091 \newcommand*{\LWR@igwidthstyle}{%
6092 \newcommand*{\LWR@igheightstyle}{%
6093 \newcommand*{\LWR@igorigin}{%
6094 \newcommand*{\LWR@igangle}{%
6095 \newcommand*{\LWR@igxscale}{1}
6096 \newcommand*{\LWR@igyscale}{1}
6097 \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:

```

6098 \define@key{igraph}{width}{%
6099 \setlength{\LWR@igwidth}{#1}%
6100 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}%
6101 {%

```

Default to use the converted fixed length given:

```

6102 \uselengthunit{PT}%
6103 \renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%

```

If ex or em dimensions were given, use those instead:

```

6104 \IfEndWith{#1}{ex}%
6105 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes ex
6106 {}% not ex
6107 \IfEndWith{#1}{em}%
6108 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes em
6109 {}% not em
6110 \IfEndWith{#1}{\}%
6111 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes percent
6112 {}% not percent
6113 \IfEndWith{#1}{px}%
6114 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
6115 {}% not px
6116 }{}% end of length > 0pt
6117 }

```

If an optional height was given, set an HTML style:

```

6118 \define@key{igraph}{height}{%
6119 \setlength{\LWR@igheight}{#1}%
6120 \ifthenelse{\lengthtest{\LWR@igheight > 0pt}}%
6121 {%

```

Default to use the converted fixed length given:

```

6122 \uselengthunit{PT}%
6123 \renewcommand*{\LWR@igheightstyle}{%
6124 height:\rndprintlength{\LWR@igheight} %
6125 }%

```

If ex or em dimensions were given, use those instead:

```

6126 \IfEndWith{#1}{ex}%
6127 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
6128 {}% not ex
6129 \IfEndWith{#1}{em}%
6130 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
6131 {}% not em
6132 \IfEndWith{#1}{\}%
6133 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
6134 {}% not percent
6135 \IfEndWith{#1}{px}%
6136 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
6137 {}% not px
6138 }{}% end of length > 0pt
6139 }

```

Handle origin key:

```

6140 \define@key{igraph}{origin}{%
6141 \renewcommand*{\LWR@igorigin}{#1}%
6142 }

```

Handle angle key:

```

6143 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}

```

Handle class key:

```

6144 \define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}
6145

```

It appears that graphicx does not have separate keys for xscale and yscale. scale adjusts both at the same time.

```

6146 \define@key{igraph}{scale}{%
6147 \renewcommand*{\LWR@igxscale}{#1}%
6148 \renewcommand*{\LWR@igyyscale}{#1}}

```

Numerous ignored keys:

```

6149 \define@key{igraph}{bb}{}
6150 \define@key{igraph}{bblx}{}
6151 \define@key{igraph}{bbly}{}

```



```

6152 \define@key{igraph}{bburx}{}
6153 \define@key{igraph}{bbury}{}
6154 \define@key{igraph}{natwidth}{}
6155 \define@key{igraph}{natheight}{}
6156 \define@key{igraph}{hiresbb}{}
6157 \define@key{igraph}{viewport}{}
6158 \define@key{igraph}{trim}{}
6159 \define@key{igraph}{totalheight}{}
6160 \define@key{igraph}{keepaspectratio}{}
6161 \define@key{igraph}{clip}{}
6162 \define@key{igraph}{draft}{}
6163 \define@key{igraph}{type}{}
6164 \define@key{igraph}{ext}{}
6165 \define@key{igraph}{read}{}
6166 \define@key{igraph}{command}{}

```

`\LWR@rotstyle`  $\{\langle prefix \rangle\} \{\langle degrees \rangle\}$

Prints the rotate style with the given prefix.

`prefix` is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:rotate` style.

```

6167 \newcommand*{\LWR@rotstyle}[2]{%
6168   #1transform:rotate(-#2deg);
6169 }

```

`\LWR@scalestyle`  $\{\langle prefix \rangle\} \{\langle xscale \rangle\} \{\langle yscale \rangle\}$

Prints the scale style with the given prefix.

`prefix` is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:scale` style.

```

6170 \newcommand*{\LWR@scalestyle}[3]{%
6171   #1transform:scale(#2,#3);
6172 }

```

## 64.3 \includegraphics

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```

6173 \newbool{LWR@infloatrow}
6174 \boolfalse{LWR@infloatrow}

6175 \newcommand*{\LWR@imageextension}{}
6176 \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.

```
6177 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
6178 {%
```

Start the image tag on a new line, allow PDF output word wrap:

```
6179 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
6180 \ifthenelse{\cnttest{\value{\LWR@minipagedepth}}{=}{0}}{%
6181 \ifbool{\LWR@infloatrow}%
6182 {}
6183 {% not in a minipage or a floatrow:
6184 \setlength{\linewidth}{6in}%
6185 \setlength{\textwidth}{6in}%
6186 \setlength{\textheight}{9in}%
6187 }%
6188 }{}}%
```

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.

```
6189 \edef\LWR@expgraphicsfilename{#4}
6190 \renewcommand*{\LWR@imageextension}{}%
6191 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.jpg}%
6192 {\renewcommand*{\LWR@imageextension}{.jpg}}{}%
6193 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.JPG}%
6194 {\renewcommand*{\LWR@imageextension}{.JPG}}{}%
6195 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.png}%
6196 {\renewcommand*{\LWR@imageextension}{.png}}{}%
6197 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.PNG}%
6198 {\renewcommand*{\LWR@imageextension}{.PNG}}{}%
6199 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.gif}%
6200 {\renewcommand*{\LWR@imageextension}{.gif}}{}%
6201 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.GIF}%
6202 {\renewcommand*{\LWR@imageextension}{.GIF}}{}%
6203 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svg}%
6204 {\renewcommand*{\LWR@imageextension}{.svg}}{}%
```

```

6204 {\renewcommand*{\LWR@imageextension}{.svg}}{}%
6205 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVG}%
6206 {\renewcommand*{\LWR@imageextension}{.SVG}}{}%
6207 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svgz}%
6208 {\renewcommand*{\LWR@imageextension}{.svgz}}{}%
6209 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVGZ}%
6210 {\renewcommand*{\LWR@imageextension}{.SVGZ}}{}%

```

Convert a PDF extension to SVG, leave the result in \LWR@strresult:

Must also \detokenize .pdf and .svg comparison strings.

```

6211 \StrSubstitute{\detokenize\expandafter{\LWR@expgraphicsfilename}}%
6212 {\detokenize{.pdf}}{\detokenize{.svg}}[\LWR@strresult]%
6213 %
6214 \StrSubstitute{\LWR@strresult}%
6215 {\detokenize{.PDF}}{\detokenize{.SVG}}[\LWR@strresult]%

```

For correct em sizing during the width and height conversions:

```

6216 \large%

```

Reset some defaults, possibly will be changed below if options were given:

```

6217 \setlength{\LWR@igwidth}{0pt}%
6218 \setlength{\LWR@igheight}{0pt}%
6219 \renewcommand*{\LWR@igwidthstyle}{}%
6220 \renewcommand*{\LWR@igheightstyle}{}%
6221 \renewcommand*{\LWR@igorigin}{}%
6222 \renewcommand*{\LWR@igangle}{}%
6223 \renewcommand*{\LWR@igxscale}{1}%
6224 \renewcommand*{\LWR@igyscale}{1}%
6225 \renewcommand*{\LWR@igclass}{inlineimage}%

```

If #3 is empty, only one optional argument was given, thus graphicx syntax:

```

6226 \IfValueTF{#3}{}{}%
6227 \IfValueTF{#2}%
6228 {\setkeys{igraph}{#2}}%
6229 {\setkeys{igraph}{}}%
6230 }%

```

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.

```

6231 \href{\thisgraphicspath\LWR@strresult\LWR@imageextension}%

```

```

6232 {% start of href
6233 \LWR@htmltag{% start of image tags
6234 img src="\thisgraphicspath\LWR@strresult\LWR@imageextension" \LWR@orignewline
6235 \LWR@origtilde{} alt="\LWR@strresult" \LWR@orignewline

```

Only include a style tag if a width, height, angle, or scale was given:

```

6236 \ifthenelse{
6237 \NOT\equal{\LWR@igwidthstyle}{}} \OR
6238 \NOT\equal{\LWR@igheightstyle}{}} \OR
6239 \NOT\equal{\LWR@igorigin{}} \OR
6240 \NOT\equal{\LWR@igangle{}} \OR
6241 \NOT\equal{\LWR@igxscale}{1}} \OR
6242 \NOT\equal{\LWR@igyscale}{1}}
6243 }%
6244 {\LWR@origtilde{} style="%
6245 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{}}%
6246 {\LWR@igwidthstyle;}}}%
6247 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{}}%
6248 {\LWR@igheightstyle;}}}%
6249 \ifthenelse{\NOT\equal{\LWR@igorigin{}}}%
6250 {\LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{}}%
6251 \ifthenelse{\NOT\equal{\LWR@igangle{}}}%
6252 {%
6253 \LWR@rotstyle{-ms-}{\LWR@igangle}%
6254 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
6255 \LWR@rotstyle{}{\LWR@igangle}%
6256 }}}%
6257 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}}\OR%
6258 \NOT\equal{\LWR@igyscale}{1}}%
6259 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}%
6260 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
6261 \LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyscale}}}%
6262 " \LWR@orignewline}{}}%

```

Set the class:

```

6263 \LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%
6264 }% end of image tags
6265 }% end of href
6266 \endgroup

```

Return to small-sized output:

```

6267 \LWR@origscriptsize
6268 }

```

`\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.

```
6269 \renewcommand*{\includegraphics}
6270 {%
```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```
6271 \LWR@ensuredoingapar%
6272 \begin{group}%
6273 \LWR@includegraphicsb%
6274 }
```

```
6275 \end{warpHTML}
```

**for PRINT output:** For print output, accept and then discard the new class key:

```
6276 \begin{warpprint}
6277 \define@key{Gin}{class}{}
6278 \end{warpprint}
```

## 64.4 \rotatebox, \scalebox, \reflectbox

**for HTML output:** 6279 \begin{warpHTML}

\LWR@rotboxorigin Holds the origin key letters.

```
6280 \newcommand*{\LWR@rotboxorigin}{}%
```

\LWR@originname {\letter}

Given one L<sup>A</sup>T<sub>E</sub>X origin key value, translate into an HTML origin word:

```
6281 \newcommand*{\LWR@originname}[1]{%
6282 \ifthenelse{\equal{#1}{t}}{top}{}%
6283 \ifthenelse{\equal{#1}{b}}{bottom}{}%
6284 \ifthenelse{\equal{#1}{c}}{center}{}%
6285 \ifthenelse{\equal{#1}{l}}{left}{}%
6286 \ifthenelse{\equal{#1}{r}}{right}{}%
6287 }
```

`\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:

```
6288 \newcommand*\LWR@originnames[1]{%
6289 \StrChar{#1}{1}[\LWR@strresult]%
6290 \LWR@originname{\LWR@strresult}
6291 \StrChar{#1}{2}[\LWR@strresult]%
6292 \LWR@originname{\LWR@strresult}
6293 }
```

Handle the origin key for `\rotatebox`:

```
6294 \define@key{krotbox}{origin}{%
6295 \renewcommand*\LWR@rotboxorigin{#1}%
6296 }
```

These keys are ignored:

```
6297 \define@key{krotbox}{x}{}
6298 \define@key{krotbox}{y}{}
6299 \define@key{krotbox}{units}{}

```

`\rotatebox`  $[\langle keyval list \rangle] \{\langle angle \rangle\} \{\langle text \rangle\}$

Will `\let\rotatebox\LWR@rotatebox` at `\LWR@LwarpStart`, in case `\rotatebox` was over-written by a later package load.

```
6300 \NewDocumentCommand{\LWR@rotatebox}{0{ m +m}}{%
```

Reset the origin to “none-given”:

```
6301 \renewcommand*\LWR@rotboxorigin{}
```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
6302 \setkeys{krotbox}{#1}%
```

Select `inline-block` so that HTML will transform this span:

```
6303 \LWR@htmltagc{span style="display: inline-block; %
```

If an origin was given, translate and print the origin information:

```
6304 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}{%
6305 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}}%
```

Print the rotation information:

```
6306 \LWR@rotstyle{-ms-}{#2} %
6307 \LWR@rotstyle{-webkit-}{#2} %
6308 \LWR@rotstyle{}{#2} %
6309 "{}\LWR@orignewline%
```

Print the text to be rotated:

```
6310 \begin{LWR@nestspan}%
6311 #3%
```

Close the span:

```
6312 \LWR@htmltagc{/span}%
6313 \end{LWR@nestspan}%
6314 }
```

`\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.

```
6315 \NewDocumentCommand{\LWR@scalebox}{m o m}{%
```

Select `inline-block` so that HTML will transform this span:

```
6316 \LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
6317 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6318 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6319 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6320 "{}}%
```

Print the text to be scaled:

```
6321 \begin{LWR@nestspan}%
6322 #3%
```

Close the span:

```
6323 \LWR@htmltagc{/span}%
6324 \end{LWR@nestspan}%
6325 }
```

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

Will `\let\reflectbox\LWR@reflectbox` at `\LWR@LwarpStart`, in case `\reflectbox` was over-written by a later package load.

```
6326 \newcommand{\LWR@reflectbox}[1]{\LWR@scalebox{-1}[1]{#1}}
6327 \end{warpHTML}
```

## 64.5 Null functions

These functions are not supported by `lwarp`'s HTML conversion.

for HTML output: 6328 `\begin{warpHTML}`

```
\resizebox {<h-length>} {<v-length>} {<text>}
```

Simply prints its text argument.

```
6329 \renewcommand{\resizebox}[3]{#3}
6330 \end{warpHTML}
```

## 65 Cleverref

Pkg `cleveref` `cleveref` package is used as-is with minor patches.

loading order `cleveref` and the following associated macro patches are automatically preloaded at the end of the preamble via `\AtEndPreamble` and `\AfterEndPreamble`. This is done because the HTML conversion requires `cleveref`. The user's document may not require `cleveref`, thus the user may never explicitly load it, so during HTML output `lwarp` loads it last. If the user's document preamble uses `cleveref` options, or functions such as `\crefname`, then `cleveref` may be loaded in the user's preamble near the end, and `lwarp`'s additional loading of `cleveref` will have no effect.

Table 9 on 264 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: 6331 `\begin{warpHTML}`

```
\AtEndPreamble forces cleveref to be loaded last:
```

The following patches are applied after `cleveref` has loaded, and after `\AtBeginDocument`:



```

6332 \AfterEndPreamble{

\@@setcref  {\<kindofref\>} {\<label\>}

6333 \renewcommand*{\@@setcref}[2]{\#1{\ref{\#2}}{\}\{\}}

\@@setcrefrange  {\<text\>} {\<label\>} {\<label\>}

6334 \renewcommand{\@@setcrefrange}[3]{%
6335 \#1{\ref{\#2}}{\ref{\#3}}{\}\{\}\{\}}

\cpagerefFor  Redefinable word between “page(s)” and the page numbers.

6336 \newcommand*{\cpagerefFor}{for}

\@@setcpageref  {\<typeofref\>} {\<label\>}, where typeofref is “page” or “pages”

6337 \renewcommand*{\@@setcpageref}[2]{%
6338 \#1{\cpagerefFor\ \cref{\#2}}{\}\{\}}%
6339 }

6340 \renewcommand{\@@setcpagerefrange}[3]{%
6341 \#1{\cpagerefFor\ \cref{\#2}}{\cref{\#3}}{\}\{\}\{\}}
6342 }% AfterEndPreamble

```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

```

6343 \let\LWR@origlabel\label
6344 \let\label\LWR@newlabel
6345 \let\LWR@origref\ref
6346 \let\ref\LWR@newref% \end{ syntax highlighting
6347 \let\LWR@origpageref\pageref
6348 \let\pageref\LWR@newpageref
6349
6350
6351
6352 \end{warpHTML}

```

## 66 Picture

**Env** `picture` The `picture` environment is enclosed inside a `\lateximage`.

**for HTML output:** 6353 `\begin{warpHTML}`

Env `picture`

```

6354 \BeforeBeginEnvironment{picture}{%
6355 \lateximage%
6356 \let\makebox\LWR@origmakebox%
6357 }
6358
6359 \AfterEndEnvironment{picture}{\endlateximage}


6360 \end{wrapHTML}

```

## 67 Boxes and Minipages

A CSS flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

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

 **inline** A line of text with an inline minipage or parbox will have the minipage or parbox placed onto its own line, because a paragraph is a block element and cannot be made `inline-block`.


**side-by-side** Side-by-side minipages may be separated by `\quad`, `\qquad`, `\enskip`, `\hspace`, `\hfill`, or a `\rule`. When inside a `center` environment, the result is similar in print and HTML. Paragraph tags are suppressed between side-by-side minipages and these spacing commands, but not at the start or end of the paragraph.

**in a span** There is limited support for minipages inside an HTML `<span>`. An HTML `<div>` cannot appear inside a `<span>`. While in a `<span>`, minipages and parboxes are ignored. Use `\newline` or `\par` for an HTML break.

**size** When using `\linewidth`, `\textwidth`, and `\textheight`, widths and heights are scaled proportionally to a 6×9 inch text area.

**no-width minipages** A minipage of width exactly `\linewidth` is automatically given no HTML width.

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

 **text alignment** Nested minipages adopt their parent's text alignment in HTML, whereas in regular L<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:** `6361 \begin{wrapHTML}`

## 67.1 Counters and lengths

**Ctrl** `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
6362 \newcounter{LWR@minipagedepth}
6363 \setcounter{LWR@minipagedepth}{0}
```

**Len** `\WR@minipagewidth` Used to convert the width into printable units.

```
6364 \newlength{\LWR@minipagewidth}
```

**Len** `\WR@minipageheight` Used to convert the height into printable units.

```
6365 \newlength{\LWR@minipageheight}
```

Remember the original definitions:

```
6366 \let\LWR@origminipage\minipage
6367 \let\LWR@origendminipage\endminipage
```

## 67.2 Footnote handling

Also see section [41](#) for other forms of footnotes.

## 67.3 Minipage handling

`\LWR@endminipage` Used to close a minipage.

Copied the L<sup>A</sup>T<sub>E</sub>X definition and modified to create a `mpfootnotes` div class:

```
6368 \def\LWR@endminipage{%
6369   \par
6370   \unskip
6371   \ifvoid\@mpfootins\else
6372     \vskip\skip\@mpfootins
6373     \normalcolor
6374   \LWR@htmldivclass{mpfootnotes}
6375   \LWR@origmedskip
6376   \unvbox\@mpfootins
6377   \LWR@htmldivclassend{mpfootnotes}
6378   \fi
6379   \@minipagefalse
6380   \color@endgroup
6381   \egroup
6382   \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.

```
6383 \newcommand*{\LWR@subminipage}{%
6384 \LWR@stoppars
6385 \LWR@origminipage{6in}
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
6386 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
6387 \LWR@startpars%
6388 }
```

`\LWR@endsubminipage` Closes the subminipage.

```
6389 \newcommand*{\LWR@endsubminipage}{%
6390 \LWR@stoppars%
6391 \LWR@endminipage% The following empty line is required:
6392
6393 }
```

Bool `LWR@minipagefullwidth` Should the next minipage have no HTML width?

```
6394 \newbool{LWR@minipagefullwidth}
6395 \boolfalse{LWR@minipagefullwidth}
```

`\minipagefullwidth` Requests that the next minipage have no width tag in HTML:

**for HTML output:**

```
6396 \newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}
6397 \end{warpHTML}
```

**for PRINT output:**

```
6398 \begin{warpprint}
6399 \newcommand*{\minipagefullwidth}{}
6400 \end{warpprint}
```

**for HTML output:**

```
6401 \begin{warpHTML}
```

Bool `LWR@minipagethispar` Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
6402 \newbool{LWR@minipagethispar}
6403 \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.

```
6404 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}
6405 {%
```

Pre-compute the given width and height:

Reset the text area if are starting the outer-most minipage:

```
6406 \LWR@traceinfo{starting minipage of width #4}%
6407 \uselengthunit{in}%
6408 \setlength{\LWR@minipagewidth}{#4}%
6409 \ifthenelse{\cnttest{\value{\LWR@minipagedepth}}{=}{0}}{%
6410 \addtolength{\LWR@minipagewidth}{3em}% room for frames
6411 \setlength{\linewidth}{6in}%
6412 \setlength{\textwidth}{6in}%
6413 \setlength{\textheight}{9in}%
6414 }{}%
6415 \LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}
6416 \addtocounter{\LWR@minipagedepth}{1}%
6417 \setlength{\LWR@minipageheight}{\textheight}% default unless specified
6418 \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.

```
6419 \LWR@stoppars%
```

Create the `<div>` tag with optional alignment style:

```
6420 \LWR@traceinfo{minipage: creating div class}%
6421 \LWR@orignewpage%
6422 \LWR@htmltag{div class="minipage" style="%
6423 \ifthenelse{\equal{#1}{t}}{vertical-align: bottom ; }{}%
6424 \ifthenelse{\equal{#1}{c}}{vertical-align: middle ; }{}%
6425 \ifthenelse{\equal{#1}{b}}{vertical-align: top ; }{}%
6426 \ifthenelse{\equal{#3}{t}}{justify-content: flex-start ; }{}%
6427 \ifthenelse{\equal{#3}{c}}{justify-content: center ; }{}%
6428 \ifthenelse{\equal{#3}{b}}{justify-content: flex-end ; }{}%
6429 \ifthenelse{\equal{#3}{s}}{justify-content: space-between ; }{}%
```

Print the width and optional height styles:

```

6430 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
6431 \uselengthunit{PT}%
6432 \ifbool{\LWR@minipagefullwidth}%
6433 {\boolfalse{\LWR@minipagefullwidth}}%
6434 {%
6435 \ifthenelse{\lengthtest{#4}=\linewidth}%
6436 {}%
6437 {width:\rndprintlength{\LWR@minipagewidth} ; }%
6438 }%
6439 \LWR@traceinfo{minipage: about to print the height}%
6440 \IfValueTF{#2}{height:\rndprintlength{\LWR@minipageheight} ; }{}%
6441 "{}%

```

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.

```

6442
6443 \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.

```

6444 \setlength{\linewidth}{#4}% the original width
6445 \setlength{\textwidth}{6in}%
6446 \setlength{\textheight}{9in}%

```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```

6447 \LWR@origraggedright%

```

Resume paragraph tag handling for the contents of the minipage:

```

6448 \LWR@startpars%
6449 \LWR@traceinfo{minipage: finished starting the minipage}%
6450 }

```

End the environment with L<sup>A</sup>T<sub>E</sub>X processing and closing tag:

```

6451 {%
6452 \LWR@stoppars%
6453 \LWR@endminipage% The following empty line is required:
6454
6455 \LWR@htmldivclassend{minipage}%
6456 \vspace{1\baselineskip}% required for subcaption
6457 \addtocounter{\LWR@minipagedepth}{-1}%

```

```
6458 \LWR@startpars%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
6459 \global\booltrue{LWR@minipagethispar}%
6460 }
```

## 67.4 Parbox, makebox, framebox, fbox, raisebox

```
\parbox  [<pos>] [<height>] [<inner-pos>] {<width>} {<text>}
```

A parbox uses the minipage code:

```
6461 \RenewDocumentCommand{\parbox}{0{t} o 0{t} m +m}
6462 {
6463 \LWR@traceinfo{parbox of width #4}%
6464 \begin{minipage}[#1][#2][#3]{#4}
6465 #5
6466 \end{minipage}
6467 }
```

```
\makebox  [<width>] [<pos>] {<text>}
```

Width and position are ignored.

```
6468 \let\LWR@origmakebox\makebox
6469
6470 \RenewDocumentCommand{\makebox}{o o m}{%
6471 \mbox{#3}
6472 }
```

```
\framebox  [<width>] [<pos>] {<text>}
```

Width and position are ignored.

```
6473 \RenewDocumentCommand{\framebox}{o o m}{%
6474 \fbox{#3}
6475 }
```

```
\fbox  {<text>}
```

```
6476 \let\LWR@origfbox\fbox
6477 %
6478 \renewcommand*{\fbox}[1]{%
```

```
6479 \InlineClass{framebox}{#1}%
6480 }
```

```
\raisebox {<raiselen>} [<height>] [<depth>] {<text>}
```

```
6481 \RenewDocumentCommand{\raisebox}{m o o m}{%
6482 #4%
6483 }
```

```
6484 \end{warpHTML}
```

## 68 Direct formatting

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

For high-level block and inline custom CSS classes, see section [35.7](#).

for HTML output: 6485 `\begin{warpHTML}`

```
\emph {<text>}
```

```
6486 \renewcommand{\emph}[1]{\LWR@htmlspan{em}{#1}}
```

```
\textmd {<text>}
```

```
6487 \renewcommand{\textmd}[1]{\LWR@htmlspan{textmd}{#1}}
```

```
\textbf {<text>}
```

```
6488 \renewcommand{\textbf}[1]{\LWR@htmlspan{b}{#1}}
```

```
\textrm {<text>}
```

```
6489 \renewcommand{\textrm}[1]{\InlineClass{textrm}{#1}}
```

```
\textsf {<text>}
```

```
6490 \renewcommand{\textsf}[1]{\InlineClass{textsf}{#1}}
```

```
\texttt {<text>}
```

```
6491 \renewcommand{\texttt}[1]{\LWR@htmlspan{kbd}{#1}}
```



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

6492 `\renewcommand{\textup}[1]{\InlineClass{textup}{#1}}`

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

6493 `\renewcommand{\textit}[1]{\LWR@htmlspan{i}{#1}}`

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

6494 `\renewcommand{\textsc}[1]{\InlineClass{textsc}{#1}}`

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

6495 `\renewcommand{\textnormal}[1]{\textmd{\textrm{\textup{#1}}}}`

`\mdseries`

6496 `\renewcommand*{\mdseries}{}`

`\bfseries`

6497 `\renewcommand*{\bfseries}{}`

`\rmfamily`

6498 `\renewcommand*{\rmfamily}{}`

`\sffamily`

6499 `\renewcommand*{\sffamily}{}`

`\ttfamily`

6500 `\renewcommand*{\ttfamily}{}`

`\upshape`

6501 `\renewcommand*{\upshape}{}`

`\itshape`

6502 `\renewcommand*{\itshape}{}`

`\scshape`

```
6503 \renewcommand*\scshape{}
```

`\scshape`

```
6504 \renewcommand*\normalfont{}
```

`\sp`  $\langle text \rangle$

For siunitx. Must work in math mode.

```
6505 \renewcommand{\sp}[1]{\text{<sup>#1</sup>}}
```

`\sb`  $\langle text \rangle$

For siunitx. Must work in math mode.

```
6506 \renewcommand{\sb}[1]{\text{<sub>#1</sub>}}
```

`\textsuperscript`  $\langle text \rangle$

```
6507 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

`\textsubscript`  $\langle text \rangle$

```
6508 \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.

```
6509 \AtBeginDocument{\let\up\textsuperscript}
```

`\fup`  $\langle text \rangle$  Prints superscript.

Supports `fmtcount` package.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
6510 \AtBeginDocument{\let\fup\textsuperscript}
```

`\hfill`

```
6511 \renewcommand*{\hfill}{\quad}
```

`\hrulefill`

```
6512 \renewcommand*{\hrulefill}{\rule{1in}{1pt}}
```

`\dotfill`

```
6513 \renewcommand*{\dotfill}{\dots}
```

```
6514 \end{warpHTML}
```

## 69 Skips, spaces, font sizes

for HTML output: 6515 `\begin{warpHTML}`

`\`, must be redefined after `\RequirePackage{printlen}`

```
6516 \let\LWR@origcomma\,
6517 \let\LWR@origtilde~
6518 \let\LWR@origenskip\enskip
6519 \let\LWR@origquad\quad
6520 \let\LWR@origqqquad\qqquad
6521 \let\LWR@orighspace\hspace
6522 \let\LWR@origrule\rule
6523 \let\LWR@origmedskip\medskip
```

Direct-formatting space commands become HTML entities:

```
6524 \renewcommand*{\,}{\HTMLUnicode{202f}} % HTML thin non-breakable space
```

```
6525 \renewcommand*{~}{\HTMLentity{nbsp}}
```

```
6526 \renewcommand*{\textellipsis}{\HTMLUnicode{2026}}
```

Direct-formatting font sizes are ignored:

```
6527 \let\LWR@orignormalsize\normalsize
6528 \let\LWR@origsmall\small
6529 \let\LWR@origfootnotesize\footnotesize
6530 \let\LWR@origscriptsize\scriptsize
6531 \let\LWR@origtiny\tiny
6532 \let\LWR@origlarge\large
```

```

6533 \let\LWR@origLarge\Large
6534 \let\LWR@origLARGE\LARGE
6535 \let\LWR@orighuge\huge
6536 \let\LWR@origHuge\Huge
6537 \renewcommand*\normalsize{}
6538 \renewcommand*\small{}
6539 \renewcommand*\footnotesize{}
6540 \renewcommand*\scriptsize{}
6541 \renewcommand*\tiny{}
6542 \renewcommand*\large{}
6543 \renewcommand*\Large{}
6544 \renewcommand*\LARGE{}
6545 \renewcommand*\huge{}
6546 \renewcommand*\Huge{}
6547
6548 \renewcommand*\onecolumn{}
6549
6550 \renewcommand{\twocolumn}[1][]{
6551
6552 #1
6553
6554 }

```

`\newline` Uses HTML `<br />` tag

```

6555 \newcommand*\LWR@newlinebr{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
6556 \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`.

```

6557 \let\LWR@origendoffline\\
6558 \NewDocumentCommand{\LWR@endoffline}{s o}
6559 {%
6560 \newline%
6561 }

```

`\LWR@minipagestartpars` Minipages are often placed side-by-side inside figures, with a bit of horizontal space to separate them. Since HTML does not allow a `<div>` to be inside a `p`, paragraphs must be turned off during the generation of the minipage, then turned on after the minipage is complete. When this occurs between side-by-side minipages, `lwarp` correctly suppresses the paragraph tags between the minipages, unless some other

text is between the minipages. Such text forms its own paragraph, resulting in text after a minipage to be on its own line. Since people often place small horizontal space between minipages, it is desirable to maintain this space if possible. `lwarp` tries to do this by remembering that a minipage has been seen, in which case paragraph tags are suppressed around `\hspace`, `\enskip`, `\quad`, and `\qquad` until the end of the paragraph, when the closing `p` tag is created.

When a minipage is seen, the boolean `LWR@minipagethispar` is set, telling the following horizontal whitespace commands to try to suppress their surrounding paragraph tags. `LWR@minipagethispar` is cleared at the next end of paragraph, when the HTML paragraph closing tag is generated.

Placed just before `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
6562 \newcommand*{\LWR@minipagestartpars}{%
6563 \ifbool{LWR@minipagethispar}%
6564 {%
6565 \LWR@startpars%
6566 }{}%
6567 }
```

`\LWR@minipagestoppars` Placed just after `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
6568 \newcommand*{\LWR@minipagestoppars}{%
6569 \ifbool{LWR@minipagethispar}%
6570 {%
6571 \LWR@stoppars%
6572 }{}%
6573 }
```

`\quad` Handles special minipage & horizontal space interactions.

```
6574 \renewcommand*{\quad}{%
6575 \LWR@minipagestoppars%
6576 \HTMLUnicode{2001}%
6577 \LWR@minipagestartpars%
6578 }
```

`\qquad` Handles special minipage & horizontal space interactions.

```
6579 \renewcommand*{\qquad}{\quad\quad}
```

`\enskip` Handles special minipage & horizontal space interactions.

```
6580 \renewcommand*{\enskip}{%
6581 \LWR@minipagestoppars%
6582 \HTMLUnicode{2000}%
```

```
6583 \LWR@minipagestartpars%
6584 }
```

Len \WR@tempwidth Used to compute span width, height, raise for \hspace and \rule:

```
Len \WR@tempheight 6585 \newlength{\LWR@tempwidth}
Len \WR@tempraise 6586 \newlength{\LWR@tempheight}
6587 \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.

```
6588 \NewDocumentCommand{\LWR@hspace}{s m}{%
6589 \setlength{\LWR@tempwidth}{#2}%
```

If \fill, change to \qquad:

```
6590 \ifnum\gluestretchorder\LWR@tempwidth>0%
6591 \setlength{\LWR@tempwidth}{2em}%
6592 \fi%
```

Only if the width is not zero:

```
6593 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{Opt}}{ }{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
6594 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
6595 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{.16667em}}
6596 {%
6597 \HTMLUnicode{2009}% thin breakable space
6598 }%
```

Print the span with the converted width. Not rounded.

```
6599 {%
6600 \uselengthunit{PT}%
6601 \LWR@htmltagc{%
6602 span style="width:\printlength{\LWR@tempwidth}; display:inline-block;"%
```

```

6603 }%
6604 \LWR@htmltagc{/span}%
6605 }%

```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

6606 \LWR@minipagestartpars%
6607 }%
6608 }

```

`\hspace` \*  $\langle length \rangle$

Handles special minipage & horizontal space interactions.

```

6609 \let\hspace\LWR@hspace

```

`\linebreak` [ $\langle num \rangle$ ] Inserts an HTML br tag.

```

6610 \renewcommand*\linebreak[1][]{\newline}

```

`\nolinebreak` [ $\langle num \rangle$ ]

```

6611 \renewcommand*\nolinebreak[1][]{\}

```

`\pagebreak` [ $\langle num \rangle$ ] Starts a new paragraph.

```

6612 \renewcommand*\pagebreak[1][]{
6613
6614 }

```

`\nopagebreak` [ $\langle num \rangle$ ]

```

6615 \renewcommand*\nopagebreak[1][]{\}

```

`\enlargethispage` \*  $\langle len \rangle$

```

6616 \RenewDocumentCommand{\enlargethispage}{s m}{\}

```

`\LWR@rule` [ $\langle raise \rangle$ ]  $\langle width \rangle$   $\langle height \rangle$

Handles special minipage & horizontal space interactions.

Creates a span of a given width and height. Ignores the optional star.

`\fill` is zero-width, so `\hspace{\fill}` is ignored.

```
6617 \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:

```
6618 \setlength{\LWR@tempwidth}{#2}%
```

If it's zero-width then skip the entire rule:

```
6619 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}{
6620 {}% zero- width
6621 {% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
6622 \ifthenelse{\lengthtest{\LWR@tempwidth>0pt}\AND%
6623 \lengthtest{\LWR@tempwidth<1pt}}{%
6624 {\setlength{\LWR@tempwidth}{1pt}}{}}%
```

Likewise with height:

```
6625 \setlength{\LWR@tempheight}{#3}%
6626 \ifthenelse{\lengthtest{\LWR@tempheight>0pt}\AND%
6627 \lengthtest{\LWR@tempheight<1pt}}{%
6628 {\setlength{\LWR@tempheight}{1pt}}{}}%
```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
6629 \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.

```
6630 \uselengthunit{PT}%
6631 \LWR@htmltagc{%
6632 span
6633 style=" %
```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```
6634 background:\LWR@currenttextcolor; %
```

The width and height are printed, converted to PT:



```
6635 width:\printlength{\LWR@tempwidth}; %
6636 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  $\text{\LaTeX}$  length allows a typical  $\text{\LaTeX}$  expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a  $\text{\LaTeX}$  length limits the allowable syntax. To do: A superior method would compute a ratio of  $\text{\LaTeX}$  ex height, then print that to HTML with an ex unit.

```
6637 \IfValueTF{#1}%
6638 {%
6639 \setlength{\LWR@tempraise}{Opt-#1}%
6640 \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
6641 \LWR@orignewline%
6642 -ms-transform: translate(Opt,\printlength{\LWR@tempraise}); %
6643 \LWR@orignewline%
6644 -webkit-transform: translate(Opt,\printlength{\LWR@tempraise}); %
6645 \LWR@orignewline%
6646 transform: translate(Opt,\printlength{\LWR@tempraise}); %
6647 \LWR@orignewline%
6648 }{}%
```

Display inline-block to place the span inline with the text:

```
6649 display:inline-block;"%
6650 }%
6651 \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
6652 \LWR@minipagestartpars%
6653 }% non-zero width
6654 }
```

```
\rule [raise] {width} {height}
```

Handles special minipage & horizontal space interactions.

```
6655 \let\rule\LWR@rule

6656 \end{warpHTML}
```

## 70 `\phantomsection`

for HTML output: 6657 `\begin{warpHTML}`

`\phantomsection` Emulate the hyperref `\phantomsection` command, often used to insert the bibliography into table of contents:

```
6658 \newcommand*{\phantomsection}{\section*{}}
```

```
6659 \end{warpHTML}
```

## 71 `\LaTeX` and other logos

Logos for HTML and print modes:

Some of these logos may be redefined in a later package, so after loading other packages, and at the beginning of the document, their definitions are finally `\let` in `\LWR@LwarpStart`.

For CSS conversions, see:

<http://edward.oconnor.cx/2007/08/tex-poshlet>

<http://nitens.org/taraborelli/texlogo>

### 71.1 HTML logos

for HTML output: 6660 `\begin{warpHTML}`

`\TeX`  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$

`latexlogo` is a CSS class used to properly typeset the E and A in  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$  and friends.

`latexlogofont` is a CSS class used to select the font for the rest of the logo in  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{LuaT}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{ConT}_{\mathrm{E}}\mathrm{Xt}$ , etc.

```
6661 \newcommand*{\LWR@TeX}
```

```
6662 {\InlineClass{latexlogofont}}%
```

```
6663 {\InlineClass{latexlogo}{T\textsubscript{e}X}}}
```

`\LaTeX`  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X} 2_{\varepsilon}$

`\LaTeXe`

```
6664 \newcommand*{\LWR@LaTeX}
```

```
6665 {\InlineClass{latexlogofont}}%
```

```

6666 {\InlineClass{latexlogo}%
6667 {L\textsuperscript{a}T\textsubscript{e}X}}
6668
6669 \renewcommand*{\LaTeXe}
6670 {\LaTeX\InlineClass{latexlogofont}%
6671 {\,2\textsubscript{\textit{\HTMLUnicode{3B5}}}}}

```

`\LuaTeX`     $\text{LuaT}_{\text{E}}\text{X}$ ,  $\text{LuaL}^{\text{A}}\text{T}_{\text{E}}\text{X}$   
`\LuaLaTeX`

```

6672 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}\Lua\TeX}
6673 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}\Lua\LaTeX}

```

`\XeTeX`     $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$ ,  $\text{X}_{\text{E}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$   
`\XeLaTeX`

`xetexlogo` is a CSS class which aligns the backwards E in  $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$  and spaces  $\text{T}_{\text{E}}\text{X}$  appropriately.

`xelatexlogo` is a CSS class which aligns the backwards E in  $\text{X}_{\text{E}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  and spaces  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  appropriately.

```

6674 \newcommand*{\Xe}
6675   {X\textsubscript{\HTMLUnicode{18e}}}
6676 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}\Xe\TeX}
6677 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}\Xe\LaTeX}

```

`\ConTeXt`     $\text{ConT}_{\text{E}}\text{Xt}$

```

6678 \newcommand*{\LWR@ConTeXt}
6679 {\InlineClass{latexlogofont}\Con\TeX{}}%
6680 \InlineClass{latexlogofont}\t}

```

`\BibTeX`     $\text{BIBT}_{\text{E}}\text{X}$ , *MakeIndex*  
`\MakeIndex`

```

6681 \providecommand*{\BibTeX}
6682 {\InlineClass{latexlogofont}\B\textsc{ib}}\TeX}
6683
6684 \newcommand*{\MakeIndex}
6685 {\InlineClass{latexlogofont}\textit{MakeIndex}}

```

`\AmS`     $\mathcal{A}\mathcal{M}\mathcal{S}$

`amslogo` is a CSS class used for the  $\mathcal{A}\mathcal{M}\mathcal{S}$ logo.

```

6686 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}}
6687 {\InlineClass{amslogo}\textit{A\textsubscript{M}S}}}

```

`\MiKTeX` `MiKTeX`

```
6688 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}
```

`\LyX` `LyX`

`lyxlogo` is a CSS class used for the `LyX` logo.

```
6689 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
```

```
6690 \end{warpHTML}
```

## 71.2 Print logos

for PRINT output:

```
6691 \begin{warpprint}
6692 \newcommand*{\XeTeXrevE}
6693   {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
6694 \providecommand*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
6695 \providecommand*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
6696 \providecommand*{\AmS}{\%}
6697 \leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex\%}
6698 \hbox{$\mathcal M\kern-.2em\mathcal S$}
6699 \newcommand*{\LyX}{\textsf{LyX}}
6700 \providecommand*{\LuaTeX}{\mbox{Lua\TeX}}
6701 \providecommand*{\LuaLaTeX}{\mbox{Lua\LaTeX}}
6702 \providecommand*{\BibTeX}{\mbox{B\textsc{ib}\TeX}}
6703 \providecommand*{\MakeIndex}{\mbox{\textit{MakeIndex}}}
6704 \providecommand*{\ConTeXt}{\mbox{Con\TeX{t}}}
6705 \providecommand*{\MiKTeX}{\mbox{MiK\TeX}}
6706 \end{warpprint}
```

## 72 `\AtBeginDocument`, `\AtEndDocument`

for HTML output:

```
6707 \begin{warpHTML}
```

`\LWR@LwarpStart` Automatically sets up the HTML-related actions for the start and end of the document.  
`\LWR@LwarpEnd`

```
6708 \AfterEndPreamble{\LWR@LwarpStart}
6709 \AtEndDocument{\LWR@LwarpEnd}
```

```
6710 \end{warpHTML}
```

## 73 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

## 74 Abstract

*(Based on original code by PETER WILSON.)*

`Pkg abstract` abstract is supported and patched by lwarp.

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

**for HTML output:** Accept all options for lwarp-abstract:

```

1 \LWR@ProvidesPackagePass{abstract}

2 \AtBeginDocument{
3 \BeforeBeginEnvironment{abstract}{
4 \LWR@forcenewpage
5 \BlockClass{abstract}
6 }
7 \AfterEndEnvironment{abstract}{\endBlockClass}
8 }
9
10 \renewcommand{\@bsrunintitle}{%
11 \hspace*\abstitlestitle}%
12 {\abstractnamefont%
13 \InlineClass{abstractrunintitle}{\abstractname}%
14 \@bslabeldelim}%
15 }
16
17 \if@titlepage
18 \renewenvironment{abstract}{%
19 % \titlepage
20 \null\vfil
21 \@beginparpenalty\@lowpenalty
22 \if@bsrunin
23 \else
24 \if@bsstyle
25 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
26 \else
27 \ifnumber@bs
28 \num@bs

```

---

```

29         \else
30         \begin{\absnamepos}%
31 \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
32         \endparpenalty\@M
33         \end\absnamepos%
34 %%         \vspace{\abstitleskip}%
35         \fi
36     \fi
37     \vspace{\abstitleskip}%
38 \fi
39 \put@bsintoc%
40 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
41 {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
42 }
43 \else
44 \renewenvironment{abstract}{%
45 \if@bsrunin
46 \else
47 \if@bsstyle
48 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
49 \else
50 \ifnumber@bs
51 \num@bs
52 \else
53 \begin{\absnamepos}%
54 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
55 \end\absnamepos%
56 %%         \vspace{\abstitleskip}%
57         \fi
58     \fi
59     \vspace{\abstitleskip}%
60 \fi
61 \put@bsintoc%
62 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
63 {\par\end{@bstr@ctlist}}
64 \fi
65

```

## Package 3

# lwarp-afterpage.sty

## 75 Afterpage

Pkg `afterpage` Not used.

for **HTML output**: Discard all options for `lwarp-afterpage`:

```
1 \LWR@ProvidesPackageDrop{afterpage}
```

```
2 \newcommand{\afterpage}[1]{#1}
```



## Package 4

# lwarp-algorithmicx.sty

## 76 Algorithmicx

Pkg algorithmicx algorithmicx is supported with minor adjustments.

for HTML output: 1 \LWR@ProvidesPackagePass{algorithmicx}

Inside the `algorithmic` environment, level indenting is converted to a `<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 159.1.

for HTML output: 2 \begin{warpHTML}

```

3 \AtBeginEnvironment{algorithmic}{%
4 %
5 \let\origALG@doentity\ALG@doentity%
6 %
7 \renewcommand*{\ALG@doentity}{%
8 \origALG@doentity%
9 \uselengthunit{PT}%
10 \LWR@htmltagc{%
11 span style="width:\rndprintlength{\ALG@thistlm}; display:inline-block;"%
12 }%
13 \LWR@htmltagc{/span}%
14 }%
15 %
16 \let\origComment\Comment%
17 %
18 \renewcommand{\Comment}[1]{\InlineClass{floatright}{\origComment{#1}}}%
19 }

20 \end{warpHTML}
```

## Package 5

# lwarp-alltt.sty

## 77 Alltt

Pkg alltt alltt is patched for use by lwarp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{alltt}

2 \AfterEndPreamble{
3 \AtBeginEnvironment{alltt}{%
4 \LWR@forcenewpage
5 \LWR@atbeginverbatim{alltt}\unskip\vspace*{-\baselineskip}%
6 }
7 \AfterEndEnvironment{alltt}{\unskip\vspace*{-\baselineskip}\LWR@afterendverbatim}
8 }
```

## Package 6

# lwarp-amsthm.sty

## 78 AMSthm

(Based on original code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)

Pkg amsthm amsthm is patched for use by lwarp.

### CSS styling of theorems and proofs:

**Theorem:** <div> of class amsthmbody<theoremstyle>  
**Theorem Name:** <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   \@xp\@ifdefinable\csname #2\endcsname{%
17     \global\@xp\let\csname end#2\endcsname\@endtheorem
18     \ifx *#1% unnumbered, need to get one more mandatory arg
19       \edef\@tempa##1{%
20         \gdef\@xp\@nx\csname#2\endcsname{%
21           \@nx\@thm{\@xp\@nx\csname th@\the\thm@style\endcsname}%
22           }\{##1\}}%
23       \else % numbered theorem, need to check for optional arg
24         \def\@tempa{\@oparg{\@ynthm{#2}}{}}%
25       \fi
26 \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#2}}}% new
27 }%
28 \@tempa
29 }

```

Patched to enclose with CSS:

```

30 \newcommand{\LWR@haveamsthmname}{
31 \renewcommand{\thmname}[1]{\InlineClass{amsthmname\LWR@thisthmstyle}{##1}}
32 }
33
34 \newcommand{\LWR@haveamsthmnumber}{
35 \renewcommand{\thmnumber}[1]{\InlineClass{amsthmnumber\LWR@thisthmstyle}{##1}}
36 }
37
38 \newcommand{\LWR@haveamsthmnote}{
39 \renewcommand{\thmnote}[1]{\InlineClass{amsthmnote\LWR@thisthmstyle}{##1}}
40 }
41
42 \LWR@haveamsthmname
43 \LWR@haveamsthmnumber
44 \LWR@haveamsthmnote

```

Patches for CSS:

```

45 \def\@begintheorem#1#2[#3]{%
46 \LWR@forcenewpage% new
47 \BlockClass{amsthmbody\LWR@thisthmstyle}% new
48 \deferred@thm@head{
49 \the\thm@headfont \thm@indent
50   \@ifempty{#1}{\let\thmname\@gobble}{\LWR@haveamsthmname}% new
51   \@ifempty{#2}{\let\thmnumber\@gobble}{\LWR@haveamsthmnumber}% new
52   \@ifempty{#3}{\let\thmnote\@gobble}{\LWR@haveamsthmnote}% new
53   \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
54   \the\thm@headpunct~
55   \thmheadnl % possibly a newline.
56   \hskip\thm@headsep

```

```

57 }%
58 \ignorespaces}

```

Patched for CSS:

```

59 \def\@endtheorem{\endBlockClass\endtrivlist\@endpefalse }

```

Proof QED symbol:

```

60 \AtBeginDocument{
61 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
62 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
63 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
64 }

```

Patched for CSS:

```

65 \renewenvironment{proof}[1][\proofname]{\par
66 \LWR@forcenewpage% new
67 \BlockClass{amsthmproof}% new
68 \pushQED{\qed}%
69 \normalfont \topsep6\p@\@plus6\p@\relax
70 \trivlist
71 \item[\hskip\labelsep
72 \InlineClass{amsthmproofname}{#1\@addpunct{.}}]\ignorespaces% changes
73 }{%
74 \InlineClass{theoremendmark}{\popQED}\endtrivlist%
75 \endBlockClass% new
76 \@endpefalse
77 }

```

## Package 7

# lwarp-bookmark.sty

## 79 Bookmark

`\Pkg bookmark` bookmark is emulated during HTML output, and the bookmark package is ignored.

**for HTML output:** Discard all options for lwarp-bookmark:

```

1 \LWR@ProvidesPackageDrop{bookmark}

2 \newcommand*{\bookmarksetup}[1]{ }

```

---

```
3 \newcommand*{\bookmarksetupnext}[1]{  
4 \newcommand*{\bookmark}[2][ ]{  
5 \newcommand*{\bookmarkdefinestyle}[2]{  
6 \newcommand*{\bookmarkget}[1]{  
7 \newcommand{\BookmarkAtEnd}[1]{
```

## Package 8

# lwarp-booktabs.sty

## 80 Booktabs

Pkg booktabs booktabs is emulated during HTML output, and the booktabs package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{booktabs}

Booktabs emulation is spread among the tabular code.

Emulated for source compatibility.

2 \newcommand\*{\addlinespace}[1]{}

3 \newcommand\*{\morecmidrules}{}%

4 \newcommand\*{\specialrule}[3]{}%

## Package 9

# lwarp-ccaption.sty

## 81 Ccaption

Pkg ccaption ccaption is not used. The user is recommended to use caption instead.

for HTML output: 1 \LWR@loadnever{ccaption}{caption}

## Package 10

# lwarp-changepage.sty

## 82 Changepage

Pkg changepage changepage is ignored.

for HTML output: Discard all options for lwarp-changepage:

```
1 \LWR@ProvidesPackageDrop{changepage}

2 \newif\ifoddpag
3 \DeclareRobustCommand{\checkoddpag}{\oddpagetrue}
4 \DeclareRobustCommand{\changetext}[5]{ }
5 \DeclareRobustCommand{\changepag}[9]{ }
6 \newenvironment{adjustwidth}[2]{ }{ }
7 \newenvironment{adjustwidth*}[2]{ }{ }
```



## Package 11

# lwarp-cutwin.sty

## 83 Cutwin

Pkg cutwin Emulated.

for HTML output: Discard all options for lwarp-cutwin:

```

1 \LWR@ProvidesPackageDrop{cutwin}

2 \newcommand*{\opencutleft}{}
3 \newcommand*{\opencutright}{}
4 \newcommand*{\opencutcenter}{}
5 \newcommand*{\cutfuzz}{}
6
7 \newenvironment{cutout}[4]
8 {\marginpar{\windowpagestuff}}
9 {}
10
11 \newcommand*{\windowpagestuff}{}
12
13 \newcommand*{\pageinwindow}{%
14 % \begin{minipage}{.3\linewidth}
15 \windowpagestuff
16 % \end{minipage}
17 }
18
19 \newenvironment{shapedcutout}[3]
20 {\marginpar{\picinwindow}}
21 {}
22
23 \newcommand*{\putstuffinpic}{}
24
25 \newcommand*{\picinwindow}{%
26 \begin{picture}(0,0)
27 \putstuffinpic
28 \end{picture}}
```

## Package 12

# lwarp-dcolumn.sty

## 84 Dcolumn

Pkg dcolumn dcolumn is emulated during HTML output, and the dcolumn package is ignored.

```
1 \LWR@ProvidesPackageDrop{dcolumn}
```

## Package 13

# lwarp-draftwatermark.sty

## 85 Draftwatermark

Pkg `draftwatermark` `draftwatermark` is emulated during HTML output, and the `draftwatermark` package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{draftwatermark}

2 \newcommand{\SetWatermarkAngle}[1]{}
3 \newcommand{\SetWatermarkColor}[1]{}
4 \newcommand{\SetWatermarkLightness}[1]{}
5 \newcommand{\SetWatermarkFontSize}[1]{}
6 \newcommand{\SetWatermarkScale}[1]{}
7 \newcommand{\SetWatermarkHorCenter}[1]{}
8 \newcommand{\SetWatermarkVertCenter}[1]{}
9 \newcommand{\SetWatermarkText}[1]{}

```

## Package 14

# lwarp-ellipsis.sty

## 86 Ellipsis

Pkg `ellipsis` `ellipsis` is emulated during HTML output, and the `ellipsis` package is ignored.

```
1 \LWR@ProvidesPackageDrop{ellipsis}
2
3 \newcommand{\ellipsisgap}{0.1em}

```

## Package 15

# lwarp-emptypage.sty

## 87 Emptypage

Pkg emptypage emptypage is ignored.

for **HTML output**: Discard all options for lwarp-emptypage:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

## Package 16

# lwarp-endnotes.sty

## 88 Endnotes

*(Based on original code by JOHN LAVAGNINO.)*

Pkg endnotes

Discard all options for lwarp-endnotes:

```
for HTML output: 1 \LWR@ProvidesPackagePass{endnotes}

2 \def\noteformat{%
3 % \rightskip\z@ \leftskip\z@ \parindent=1.8em
4 \leavevmode
5 % \llap{
6 \makeenmark
7 % }
8 }
9
10 \def\@makeenmark{\hbox{\textsuperscript{\normalfont\theenmark}}}
11 \def\makeenmark{\@makeenmark}
```

## Package 17

# lwarp-enumerate.sty

## 89 Enumerate

Pkg `enumerate` `enumerate` is ignored. `enumitem` is then modified per the `shortlabels` option.

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

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

for HTML output: Discard all options for `lwarp-enumerate`:

```

1 \LWR@ProvidesPackageDrop{enumerate}

2 % \DeclareOption{shortlabels}
3 % {
4 \def\enit@shl#1{%
5     \ifnum\enit@type=\tw@
6         \enit@toks{#1}%
7     \else
8         \def\enit@c{#1}%
9         \enit@first#1,\@nil\@nil % Returns enit@toks
10    \fi}
11 % }
```

## Package 18

# lwarp-epigraph.sty

## 90 Epigraph

Pkg epigraph epigraph is emulated during HTML output, and the epigraph package is ignored.

```
for HTML output: 1 \LWR@ProvidesPackageDrop{epigraph}

2 \newcommand{\qitem}[2]
3 {
4 \begin{BlockClass}{qitem}
5 #1
6 \begin{BlockClass}{epigraphsource}
7 #2
8 \end{BlockClass}
9 \end{BlockClass}
10 }

11 \newcommand{\epigraph}[2]
12 {
13 \begin{BlockClass}{epigraph}
14 \qitem{#1}{#2}
15 \end{BlockClass}
16 }
17
18 \newenvironment*{epigraphs}
19 {\BlockClass{epigraph}}
20 {\endBlockClass}
```

Use CSS to format epigraphs.

The following are null commands for source compatibility:

```
21 \newlength{\epigraphwidth}
22 \setlength{\epigraphwidth}{.5\linewidth}
23 \newenvironment*{flushepinormal}{}{}
24 \newcommand{\textflush}[1]{flushepinormal}
25 \newcommand{\epigraphflush}[1]{flushright}
26 \newcommand{\sourceflush}[1]{flushright}
27 \newcommand*{\epigraphsize}{\small}
28 \newlength{\epigraphrule}
29 \newlength{\beforeepigraphskip}
30 \newlength{\afterepigraphskip}
31 \newcommand{\epigraphhead}[2][0]{#2}
```

```

32 \newcommand{\dropchapter}[1]{}
33 \newcommand*{\undodrop}{}
34 \newcommand{\cleartoevenpage}[1] [] {}

```

## Package 19

# lwarp-eso-pic.sty

## 91 Eso-pic

Pkg eso-pic eso-pic is emulated during HTML output, and the eso-pic package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{eso-pic}

2 \newcommand*{\LenToUnit}{}
3 \newcommand{\AtPageUpperLeft}[1]{}
4 \newcommand{\AtPageLowerLeft}[1]{}
5 \newcommand{\AtPageCenter}[1]{}
6 \newcommand{\AtStockLowerLeft}[1]{}
7 \newcommand{\AtStockUpperLeft}[1]{}
8 \newcommand{\AtStockCenter}[1]{}
9 \newcommand{\AtTextUpperLeft}[1]{}
10 \newcommand{\AtTextLowerLeft}[1]{}
11 \newcommand{\AtTextCenter}[1]{}
12 \NewDocumentCommand{\AddToShipoutPictureBG}{s +m}{}
13 \let\AddToShipoutPicture\AddToShipoutPictureBG
14 \NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
15 \newcommand*{\ClearShipoutPictureBG}{}
16 \newcommand*{\ClearShipoutPicture}{}
17 \newcommand*{\ClearShipoutPictureFG}{}
18 \newcommand{\gridSetup}[6] [] {}

```



## Package 20

# lwarp-everypage.sty

## 92 Everypage

Pkg `everypage` `everypage` is emulated during HTML output, and the `everypage` package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{everypage}

2 \newcommand*{\AddEverypageHook}[1]{}
3 \newcommand*{\AddThispageHook}[1]{}

```

## Package 21

# lwarp-extramarks.sty

## 93 Extramarks

Pkg `extramarks` `extramarks` is not used.

for HTML output: Discard all options for `lwarp-extramarks`:

```

1 \LWR@ProvidesPackageDrop{extramarks}

2 \newcommand*{\extramarks}[2]{}
3 \newcommand*{\firstleftxmark}{}
4 \newcommand*{\lastleftxmark}{}
5 \newcommand*{\firstrightxmark}{}
6 \newcommand*{\lastrightxmark}{}
7 \newcommand*{\firstxmark}{}
8 \newcommand*{\lastxmark}{}
9 \newcommand*{\topxmark}{}
10 \newcommand*{\topleftxmark}{}
11 \newcommand*{\firstleftmark}{}
12 \newcommand*{\lastrightmark}{}

```

## Package 22

# lwarp-fancyhdr.sty

## 94 Fancyhdr

Pkg fancyhdr fancyhdr is nullified.

for HTML output: Discard all options for lwarp-fancyhdr:

```

1 \LWR@ProvidesPackageDrop{fancyhdr}

2 \newcommand*{\fancyhead}[2] [] {}
3 \newcommand*{\fancyfoot}[2] [] {}
4 \newcommand*{\fancyhf}[2] [] {}
5 \newcommand*{\fancypagestyle}[2] {}
6 \newcommand*{\lhead}[1] {}
7 \newcommand*{\chead}[1] {}
8 \newcommand*{\rhead}[1] {}
9 \newcommand*{\lfoot}[1] {}
10 \newcommand*{\cfoot}[1] {}
11 \newcommand*{\rfoot}[1] {}
12 \newcommand*{\headrulewidth}{}
13 \newcommand*{\footrulewidth}{}
14 \newcommand*{\fancyheadoffset}[2] [] {}
15 \newcommand*{\fancyfootoffset}[2] [] {}
16 \newcommand*{\fancyhfoffset}[2] [] {}
17 \newcommand*{\iffloatpage}[2] {#2}
18 \newcommand*{\ifftopfloat}[2] {#2}
19 \newcommand*{\iffbotfloat}[2] {#2}

```

## Package 23

# lwarp-float.sty

## 95 Float and \newfloat

`Pkg float` float is emulated during HTML output, and the float package is ignored.

**for HTML output:** `1 \LWR@ProvidesPackageDrop{float}[2016/03/04]`

See section 58.2 for the `\listof` command.

`\newfloat` `{\langle type \rangle}{\langle 2: placement \rangle}{\langle 3: ext \rangle} [\langle 4: within \rangle]`

Emulates the `\newfloat` command from the float package.

“placement” is ignored.

```
2 \NewDocumentCommand{\newfloat}{m m m o}{%
3 \IfValueTF{#4}
4 {
5 \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}
6 }
7 {\DeclareFloatingEnvironment[fileext=#3]{#1}}
```

newfloat package automatically creates the `\listof` command for new floats, but float does not, so remove `\listof` here in case it is manually created later.

```
8 \cslet{listof#1s}\relax
9 \cslet{listof#1es}\relax
10 }
```

`\floatname` `{\langle type \rangle}{\langle name \rangle}`

Sets the text name of the float, such as “Figure”.

```
11 \NewDocumentCommand{\floatname}{m +m}{%
12 \SetupFloatingEnvironment{#1}{name=#2}%
13 }
```

`\floatplacement` `{\langle type \rangle}{\langle placement \rangle}`

Float placement is ignored.

```
14 \newcommand*{\floatplacement}[2]{%  
15 \SetupFloatingEnvironment{#1}{placement=#2}%  
16 }
```

`\floatstyle`  $\{\langle style \rangle\}$

Float styles are ignored.

```
17 \newcommand{\floatstyle}[1]{%  
18 }
```

`\restylefloat`  $* \{\langle style \rangle\}$

Float styles are ignored.

```
19 \NewDocumentCommand{\restylefloat}{s m}{%  
20 }
```

## Package 24

# lwarp-floatflt.sty

## 96 Floatflt

Pkg floatflt Emulated.

for HTML output: Discard all options for lwarp-floatflt:

```
1 \LWR@ProvidesPackageDrop{floatflt}
```

Borrowed from the lwarp version of keyfloat:

```
2 \NewDocumentEnvironment{KFLTfloatflt@marginfloat}{0{-1.2ex} m}
3 {% start
4 \LWR@maybeinthisfloat%
5 \LWR@forcenewpage
6 \LWR@stoppars%
7 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
8 \LWR@startpars%
9 \captionsetup{type=#2}%
10 }
11 {
12 \LWR@htmldivclassend{div}
13 }
14
15 \DeclareDocumentEnvironment{floatingfigure}{o m}
16 {\begin{KFLTfloatflt@marginfloat}{figure}}
17 {\end{KFLTfloatflt@marginfloat}}
18
19 \DeclareDocumentEnvironment{floatingtable}{o +m}
20 {\begin{KFLTfloatflt@marginfloat}{table}#2}
21 {\end{KFLTfloatflt@marginfloat}}
```


## Package 25


# lwarp-floatrow.sty

## 97 Floatrow

`Pkg floatrow` floatrow is emulated during HTML output, and the floatrow package is ignored.

**for HTML output:** `1 \LWR@ProvidesPackageDrop{floatrow}`

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

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

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

After everything has loaded, remember whether subcaption was loaded. If not, it is assumed that subfig is used instead:

```
2 \newbool{LWR@subcaptionloaded}
3
4 \AtBeginDocument{
5 \@ifpackageloaded{subcaption}
6 {\booltrue{LWR@subcaptionloaded}}
7 {\boolfalse{LWR@subcaptionloaded}}
8 }
```

```
\floatbox [1 preamble] {2 captype} [3 width] [4 height] [5 vert pos]
{6 caption} {7 object}
```

Only parameters for captype, width, caption, and object are used.

LWR@insubfloatrow is true if inside a subfloatrow environment.

There are two actions, depending on the use of `subcaption` or `subfig`.

```

9 \NewDocumentCommand{\floatbox}{o m o o o +m +m}{%
10 \ifbool{LWR@subcaptionloaded}%
11 {% subcaption

```

For `subcaption`:

```

12 \ifbool{LWR@insubfloatrow}%
13 {% subcaption in a subfloatrow

```

`subfigure` and `subtable` environments take width as an argument.

```

14 \IfValueTF{#3}%
15 {\@nameuse{sub#2}{#3}}%
16 {\@nameuse{sub#2}{\linewidth}}%
17 }% subcaption in a subfloatrow
18 {% subcaption not in subfloatrow

```

`figure` and `table` environments do not take a width argument.

```

19 \@nameuse{#2}%
20 }% subcaption not in subfloatrow
21 #6
22
23 #7

```

End the environments:

```

24 \ifbool{LWR@insubfloatrow}%
25 {\@nameuse{endsub#2}}%
26 {\@nameuse{end#2}}%
27 }% subcaption
28 {% assume subfig

```

For `subfig`:

```

29 \ifbool{LWR@insubfloatrow}%
30 {% subfig in a subfloatrow

```

`\subfloat` is a macro, not an environment.

Package `subfig`'s `\subfloat` command takes an optional argument which is the caption, but `\floatbox` argument `#6` contains commands to create the caption and label, not the caption itself. Thus, `\caption` is temporarily disabled to return its own argument without braces.

```

31 \begingroup
32 \let\caption\@firstofone

```

```

33 \subfloat[#6]{#7}
34 \endgroup
35 }% subfig in a subfloatrow
36 {% subfig package, but not a subfig

```

figure and table are environments:

```

37 \@nameuse{#2}
38 #6
39
40 #7
41 \@nameuse{end#2}
42 }% subfig package, but not a subfig
43 }% assume subfig
44 }

```

Not used:

```

45 \newcommand*{\nocapbeside}{}
46 \newcommand*{\capbeside}{}
47 \newcommand*{\captop}{}
48 \newlength{\FBwidth}
49 \setlength{\FBwidth}{.3\linewidth}
50 \newlength{\FBheight}
51 \setlength{\FBheight}{2in}
52 \newcommand*{\useFCwidth}{}
53 \newcommand{\floatsetup}[2][{}]{
54 \newcommand{\thisfloatsetup}[1]{
55 \newcommand{\clearfloatsetup}[1]{
56 \newcommand*{\killfloatstyle}{}

```

Preamble and default width are ignored.

```

57 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
58 \@namedef{#1}{
59 \floatbox{#2}
60 }
61 }

```

Preamble and default width are ignored.

```

62 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
63 \@namedef{#1}{%
64 \floatbox{#2}
65 }
66 }

67 \newfloatcommand{ffigbox}{figure}[\nocapbeside] []

```



```
68 \newfloatcommand{ttabbox}{table}[\captop][\FBwidth]
```

```
69 \newfloatcommand{fcapside}{figure}[\capbeside][]
```

The row of floats is placed into a <div> of class floatrow.

```
70 \newenvironment*{floatrow}[1][2]
```

```
71 {
```

```
72 \LWR@forcenewpage
```

```
73 \BlockClass{floatrow}
```

While inside the floatrow, divide the \linewidth by the number of floats.

```
74 \booltrue{LWR@infloatrow}
```

```
75 \setlength{\linewidth}{6in/#1}
```

```
76 }
```

```
77 {
```

```
78 \boolfalse{LWR@infloatrow}
```

```
79 \endBlockClass
```

```
80 }
```

Keys for \DeclareNewFloatType:

```
81 \newcommand*{\LWR@frowkeyplacement}{}%
```

```
82 \newcommand*{\LWR@frowkeyname}{}%
```

```
83 \newcommand*{\LWR@frowkeyfileext}{}%
```

```
84 \newcommand*{\LWR@frowkeywithin}{}%
```

```
85 \newcommand*{\LWR@frowkeycapstyle}{}%
```

```
86
```

```
87 \define@key{frowkeys}{placement}{}%
```

```
88 \define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}%
```

```
89 \define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}%
```

```
90 \define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}%
```

```
91 \define@key{frowkeys}{relatedcapstyle}{}%
```

Use \listof{type}{Title} to print a list of the floats.

```
92 \newcommand*{\DeclareNewFloatType}[2]{%
```

Reset key values:

```
93 \renewcommand*{\LWR@frowkeyplacement}{}%
```

```
94 \renewcommand*{\LWR@frowkeyname}{}%
```

```
95 \renewcommand*{\LWR@frowkeyfileext}{}%
```

```
96 \renewcommand*{\LWR@frowkeywithin}{}%
```

```
97 \renewcommand*{\LWR@frowkeycapstyle}{}%
```

Read new key values:

```

98 \LWR@traceinfo{about to setkeys frowkeys}%
99 \setkeys{frowkeys}{#2}%
100 \LWR@traceinfo{finished setkeys frowkeys}%

```

Create a new float with optional [within]:

```

101 \ifthenelse{\equal{\LWR@frowkeywithin}{}}{
102 {
103 \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\
104 \LWR@frowkeyfileext}%
105 \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}
106 }%
107 {%
108 \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\
109 \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
110 \newfloat{#1}{\LWR@frowkeyplacement}%
111 {\LWR@frowkeyfileext}[\LWR@frowkeywithin}%
112 \LWR@traceinfo{finished newfloat #1}
113 }%

```

Rename the float if a name was given:

```

114 \ifthenelse{\equal{\LWR@frowkeyname}{}}{
115 {}
116 {\floatname{#1}{\LWR@frowkeyname}}%
117 }

```

Not used:

```

118 \newcommand{\buildFBBBOX}[2]{}
119 \newcommand*{\CenterFloatBoxes}{}
120 \newcommand*{\TopFloatBoxes}{}
121 \newcommand*{\BottomFloatBoxes}{}
122 \newcommand*{\PlainFloatBoxes}{}
123
124 \newcommand{\capsubrowsettings}{}
125
126 \NewDocumentCommand{\RawFloats}{o o}{}

```

To be used inside a minipage or parbox.

```

127 \newcommand{\RawCaption}[1]{#1}

```

Places additional text inside a float, inside a CSS <div> of class floatfoot.

```

128 \NewDocumentCommand{\floatfoot}{s +m}{%
129 \begin{BlockClass}{floatfoot}
130 #2
131 \end{BlockClass}

```

```
132 }
```

Used to compute `\linewidth`.

```
133 \newbool{LWR@insubfloatrow}  
134 \boolfalse{LWR@insubfloatrow}
```

```
135 \newenvironment*{subfloatrow}[1][2]  
136 {
```

The row of floats is placed into a `<div>` of class `floatrow`:

```
137 \LWR@forcenewpage  
138 \BlockClass{floatrow}
```

While inside the `floatrow`, `LWR@insubfloatrow` is set true, which tells `\floatbox` to use `\subfigure` or `\subtable`.

```
139 \begingroup  
140 \booltrue{LWR@insubfloatrow}  
141 }  
142 {  
143 \endgroup  
144 \endBlockClass  
145 \boolfalse{LWR@insubfloatrow}  
146 }
```

## Package 26

# lwarp-fontenc.sty

## 98 Fontenc

Pkg fontenc Error if fontenc is loaded after lwarp.

Discard all options for lwarp-fontenc:

for HTML output: 1 \LWR@ProvidesPackageDrop{fontenc}  
2 \LWR@loadbefore{fontenc}

## Package 27

# lwarp-fontspec.sty

## 99 Fontspec

Pkg fontspec Error if fontspec is loaded after lwarp.

Discard all options for lwarp-fontspec:

for HTML output: 1 \LWR@ProvidesPackageDrop{fontspec}  
2 \LWR@loadbefore{fontspec}

## Package 28

# lwarp-footmisc.sty

## 100 Footmisc

*(Based on original code by ROBIN FAIRBAIRNS.)*

Pkg `footmisc` `footmisc` is emulated during HTML output, and the `footmisc` package is ignored.

```
1 \LWR@ProvidesPackageDrop{footmisc}
```

Some nullified commands:

```
2 \newcommand{\footnotelayout}{}
3 \newcommand{\setfnsymbol}[1]{}
4 \NewDocumentCommand{\DefineFNSymbols}{s m o m}{}
5
6 \newdimen\footnotemargin
7 \footnotemargin1.8em\relax
8
9 \newcommand*\hangfootparskip{0.5\baselineskip}
10 \newcommand*\hangfootparindent{0em}%
11
12 \let\pagefootnoterule\footnoterule
13 \let\mpfootnoterule\footnoterule
14 \def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}
15
16 \providecommand*\multiplefootnotemarker}{3sp}
17 \providecommand*\multfootsep}{,}
```

Using `cleveref`:

```
18 \providecommand*\footref}[1]{\labelcref{#1}}
```

The following work as-is:

```
19 \newcommand\mpfootnotemark{%
20   \@ifnextchar[%
21     \@xmpfootnotemark
22     {%
23       \stepcounter\@mpfn
24       \protected@xdef\@thefnmark{\thempfn}%
25       \@footnotemark
26     }%
27 }
28 \def\@xmpfootnotemark[#1]{%
29   \begingroup
30     \csname c@\@mpfn\endcsname #1\relax
31     \unrestored@protected@xdef\@thefnmark{\thempfn}%
32   \endgroup
33   \@footnotemark
34 }
```

## Package 29

# lwarp-footnote.sty

## 101 Footnote

Pkg footnote footnote is used with minor patches.

for HTML output: 1 \LWR@ProvidesPackagePass{footnote}

Removed print-version formatting:

```
2 \def\fn@startnote{%
3 %   \@parboxrestore%
4   \protected@edef\@currentlabel{\csname p@\@mpfn\endcsname\@thefnmark}%
5 %   \color@begingroup% *** conflicts with lwarp
6 }
7
8 % \let\fn@endnote\color@endgroup% *** conflicts with lwarp
9 \newcommand*{\fn@endnote}{\LWR@htmltagc{/LWR@tagregularparagraph}}
```

Removed print-version formatting:

```
10 \def\fn@startfntext{%
11   \setbox\z@\vbox\bgroup%
12     \fn@startnote%
13     \fn@prefntext%
14     \ignorespaces%
15 }
```

Removed print-version formatting, added closing paragraph tag:

```
16 \def\fn@endfntext{%
17 \LWR@htmltagc{/LWR@tagregularparagraph}%
18   \fn@postfntext%
19   \egroup%
20   \begingroup%
21   \let\@makefntext\@empty%
22   \let\@finalstrut\@gobble%
23   \let\rule\@gobbletwo%
24   \@footnotetext{\unvbox\z@}%
25 \endgroup%
26 }
```

These have been redefined, so re-\let them again:

```
27 \let\endfootnote\fn@endfntext
28 \let\endfootnotetext\endfootnote
```

## Package 30

# lwarp-footnotehyper.sty

## 102 Footnotehyper

Pkg footnotehyper footnotehyper is a hyperref-safe version of footnote. For lwarp, footnotehyper is emulated.

for HTML output: Discard all options for lwarp-footnotehyper:

```
1 \RequirePackage{footnote}
2 \LWR@ProvidesPackageDrop{footnotehyper}
```

## Package 31

# lwarp-framed.sty

## 103 Framed

*(Based on original code by DONALD ARSENEAU.)*

Pkg `framed` `framed` is supported and patched by `lwarp`.

for HTML output: Accept all options for `lwarp-framed`:

```

1 \LWR@ProvidesPackagePass{framed}

2
3 \renewenvironment{framed}{%
4 \LWR@forcenewpage
5 \BlockClass{framed}%
6 }
7 {\endBlockClass}
8
9 \renewenvironment{oframed}{%
10 \LWR@forcenewpage
11 \BlockClass{framed}%
12 }
13 {\endBlockClass}
14
15
16 \renewenvironment{shaded}{%
17 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
18 \LWR@forcenewpage
19 \BlockClass{framed}[background: \#\LWR@tempcolor]%
20 }
21 {\endBlockClass}
22
23 \renewenvironment{shaded*}{%
24 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
25 \LWR@forcenewpage
26 \BlockClass{framed}[background: \#\LWR@tempcolor]%
27 }
28 {\endBlockClass}
29
30
31 \renewenvironment{leftbar}{%
32 \LWR@forcenewpage
33 \BlockClass{framedleftbar}
34 \def\FrameCommand{}}%
```



```

35 \MakeFramed {}
36 }%
37 {\endMakeFramed\endBlockClass}
38
39
40 \renewenvironment{snugshade}{%
41 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
42 \LWR@forcenewpage
43 \BlockClass{snugframed}[background: \#\LWR@tempcolor]%
44 }
45 {\endBlockClass}
46
47 \renewenvironment{snugshade*}{%
48 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
49 \LWR@forcenewpage
50 \BlockClass{snugframed}[background: \#\LWR@tempcolor]%
51 }
52 {\endBlockClass}
53
54 \let\oframed\framed
55 \let\endoframed\endframed
56
57
58 \RenewEnviron{titled-frame}[1]{%
59 \CustomFBox{#1}{0pt}{0pt}{0pt}{0pt}{\BODY}
60 }

\CustomFBox {<toptitle>} {<bottitle>} {<thicknesstop>} {<bottom>} {<left>} {<right>}
{<text contents>}

61 \renewcommand{\CustomFBox}[7]{%
62 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
63 \LWR@forcenewpage
64 \begin{BlockClass}{framed}[border: 3px solid \#\LWR@tempcolor]%
65 \ifthenelse{\isempty{#1}}{}{% not empty
66 \begin{BlockClass}{framedtitle}[background: \#\LWR@tempcolor]%
67 \textcolor{TFTitleColor}{\textbf{#1}}%
68 \end{BlockClass}
69 }% not empty
70
71 #7
72
73 \ifthenelse{\isempty{#2}}{}{% not empty
74 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
75 \begin{BlockClass}{framedtitle}[background: \#\LWR@tempcolor]%
76 \textcolor{TFTitleColor}{\textbf{#2}}%
77 \end{BlockClass}
78 }% not empty
79 \end{BlockClass}

```

---

```

80 }

\TitleBarFrame [\marker] {\i\title} {\i\contents}

81 \renewcommand\TitleBarFrame[3] [] {
82 \CustomFBox
83   {\i2}{\i}%
84   \fboxrule\fboxrule\fboxrule\fboxrule
85   {\i3}%
86 }

87 \renewcommand{\TF@Title}[1]{\i1}

MakeFramed {\i\settings}

88 \let\MakeFramed\relax
89 \let\endMakeFramed\relax
90
91 \NewEnviron{MakeFramed}[1]{%
92 \FrameCommand{\begin{minipage}{\linewidth}\BODY\end{minipage}}%
93 }

\fb@put@frame {\i\frame cmd no split} {\i\frame cmd split}

94 \renewcommand*\fb@put@frame[2]{%
95 \relax%
96 \@tempboxa%
97 }

```

## Package 32

# lwarp-ftnright.sty

## 104 Ftnright

Pkg `ftnright` `ftnright` is ignored.

for HTML output: Discard all options for `lwarp-ftnright`:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

## Package 33

# lwarp-geometry.sty

## 105 Geometry

Pkg `geometry` `geometry` is preloaded by `lwarp`, but must be nullified as seen by the user's source code.

for HTML output: Discard all options for `lwarp-geometry`:

```
1 \LWR@ProvidesPackageDrop{geometry}

2 \renewcommand*{\geometry}[1]{}
3 \renewcommand*{\newgeometry}[1]{}
4 \renewcommand*{\restoregeometry}{}
5 \renewcommand*{\savegeometry}[1]{}
6 \renewcommand*{\loadgeometry}[1]{}

```

## Package 34

# lwarp-glossaries.sty

## 106 Glossaries

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

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

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

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

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

for HTML output:

```
1 \PassOptionsToPackage{xindy}{glossaries}
2 \LWR@ProvidesPackagePass{glossaries}
3 \setupglossaries{nonumberlist}
4 \setglossarystyle{index}
```

## Package 35

# lwarp-graphics.sty

## 107 Graphics

Pkg graphics graphics is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphics}

## Package 36

# lwarp-graphicx.sty

## 108 Graphicx

Pkg graphicx graphicx is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphicx}

## Package 37

# lwarp-hyperref.sty

## 109 Hyperref

Pkg hyperref hyperref is emulated during HTML output, and the hyperref package is ignored.

for HTML output:

```

1 % \LWR@ProvidesPackageDrop{hyperref}
2 \typeout{Using the lwarp html version of package 'hyperref' -- discarding options.}
3 \typeout{   Are not using ProvidesPackage, so that other packages}
4 \typeout{   do not attempt to patch lwarp's version of 'hyperref'.}
5 % \ProvidesPackage{lwarp-#1-#2}
6 \DeclareOption*{}
7 \ProcessOptions\relax

8 \newcommand*{\hypersetup}[1]{}
9 \newcommand*{\hyperbaseurl}[1]{}

```

Insert an image with alt text:

```

10 \NewDocumentCommand{\LWR@hyperimageb}{m +m}{%
11 \LWR@htmltag{img src="#1" alt="#2" class="hyperimage"{} }%
12 \endgroup%
13 \LWR@ensuredoingapar%
14 }
15
16 \newcommand{\hyperimage}{%
17 \LWR@ensuredoingapar%
18 \begingroup\catcode'\_ =12
19 \LWR@hyperimageb%
20 }

```

Creates an HTML anchor to category.name with the given text.

```

21 \NewDocumentCommand{\hyperdef}{m m +m}{%
22 \LWR@ensuredoingapar%
23 \LWR@subsublabel{#1.#2}%
24 #3%
25 }

```

Creates an HTML link to URL#category.name with the given text.

```

26 \NewDocumentCommand{\LWR@hyperrefb}{m m m +m}{%
27 \LWR@htmltag{a href="#1\LWR@hashmark#2.#3"%

```

```

28 #4%
29 \LWR@htmltag{/a}%
30 \endgroup%
31 }

```

Creates text as an HTML link to the L<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, `\cleverref` is used instead.

```

50 \NewDocumentCommand{\autoref}{s m}{%
51 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
52 }

```

For HTML, `\cleverref` is used instead.

```

53 \NewDocumentCommand{\autopageref}{s m}{%
54 \IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
55 }

```

```

56 \newcommand{\pdfstringdef}[2]{}

```

```

57 \newcommand{\pdfbookmark}[3][]{ }

```

```
58 \newcommand{\currentpdfbookmark}[2]{}

```

```
59 \newcommand{\subpdfbookmark}[2]{}

```

```
60 \newcommand{\belowpdfbookmark}[2]{}

```

```
61 \newcommand{\texorpdfstring}[2]{#2}

```

From hyperref.

```
62 \def\hypercalcbp#1{%
63 \strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax
64 }%

```

```
65 \newcommand{\Acrobatmenu}[2]{}

```

```
66 \newcommand*{\TextField}[2][{}]{

```

```
67 \newcommand*{\CheckBox}[2][{}]{

```

```
68 \newcommand{\ChoiceMenu}[3][{}]{

```

```
69 \newcommand*{\PushButton}[2][{}]{

```

```
70 \newcommand*{\Submit}[2][{}]{

```

```
71 \newcommand*{\Reset}[2][{}]{

```

```
72 \newcommand*{\LayoutTextField}[2]{}

```

```
73 \newcommand*{\LayoutChoiceField}[2]{}

```

```
74 \newcommand*{\LayoutCheckField}[2]{}

```

```
75 \newcommand*{\MakeRadioField}[2]{}

```

```
76 \newcommand*{\MakeCheckField}[2]{}

```

```
77 \newcommand*{\MakeTextField}[2]{}

```

```
78 \newcommand*{\MakeChoiceField}[2]{}

```

```
79 \newcommand{\MakeFieldButton}[1]{}

```



## Package 38

# lwarp-indentfirst.sty

## 110 Indentfirst

Pkg indentfirst indentfirst is ignored.

Discard all options for lwarp-indentfirst:

for HTML output: 1 \LWR@ProvidesPackageDrop{indentfirst}

## Package 39

# lwarp-inputenc.sty

## 111 Inputenc

Pkg inputenc Error if inputenc is loaded after lwarp.

Discard all options for lwarp-inputenc:

for HTML output: 1 \LWR@ProvidesPackageDrop{inputenc}

2 \LWR@loadbefore{inputenc}

## Package 40

# lwarp-keyfloat.sty

## 112 Keyfloat

Pkg keyfloat keyfloat is supported with minor adjustments.

for HTML output: 1 \LWR@ProvidesPackagePass{keyfloat}

After keyfloat has loaded:

```

2 \AtBeginDocument{

3 \let\KFLT@boxinner\relax
4 \let\endKFLT@boxinner\relax
5
6 \NewEnviron{KFLT@boxinner}
7 {%
8 \LWR@traceinfo{kflt@boxinner}%
9 \LWR@stoppars%
10 \KFLT@frame{\BODY}%
11 \LWR@startpars%
12 \LWR@traceinfo{ended kflt@boxinner}%
13 }

14 \DeclareDocumentEnvironment{KFLT@marginfloat}{0{-1.2ex} m}
15 {% start
16 \LWR@maybeinthisfloat%
17 \LWR@forcenewpage
18 \LWR@stoppars%
19 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
20 \LWR@startpars%
21 \captionsetup{type=#2}%
22 }
23 {
24 \LWR@htmldivclassend{div}
25 }

26 \DeclareDocumentEnvironment{marginfigure}{o}
27 {\begin{KFLT@marginfloat}{figure}}
28 {\end{KFLT@marginfloat}}
29
30 \DeclareDocumentEnvironment{margintable}{o}
31 {\begin{KFLT@marginfloat}{table}}
32 {\end{KFLT@marginfloat}}
```

---

```
33 \DeclareDocumentEnvironment{keywrap}{m +m}
34 {%
35 \begin{BlockClass}{marginblock}
36 \setlength{\linewidth}{#1}
37 #2%
38 \end{BlockClass}
39 }
40 {%
41 }

42 }% AtBeginDocument
```

## Package 41

# lwarp-layout.sty

## 113 Layout

Pkg layout layout is ignored.

for HTML output: Discard all options for lwarp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}
2 \NewDocumentCommand{\layout}{s}{}
```

## Package 42

# lwarp-letterspace.sty

## 114 Letterspace

Pkg letterspace letterspace is a subset of microtype, which is pre-loaded by lwarp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for lwarp-letterspace:

```
1 \LWR@ProvidesPackageDrop{letterspace}
2 \newcommand*\lsstyle{}
3 \newcommand\textls[2] [] {}
4 \def\textls#1#{}
5 \newcommand*\lslig[1]{#1}
```

## Package 43

# lwarp-lettrine.sty

## 115 Lettrine

(Based on original code by DANIEL FLIPO.)

Pkg lettrine Emulated.

for HTML output: Discard all options for lwarp-lettrine:

```
1 \LWR@ProvidesPackageDrop{lettrine}
```

The initial letter is in a `<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}{\}
28
29 \newcommand*{\LettrineFont}[1]{\InlineClass{lettrine}{#1}}
30 \newcommand*{\LettrineFontEPS}[1]{\includegraphics[height=1.5ex]{#1}}
```

## Package 44

# lwarp-lips.sty

## 116 Lips

Pkg lips lips is emulated during HTML output, and the lips package is ignored.

```
1 % \LWR@ProvidesPackageDrop{lips}
2 \PackageInfo{lwarp}{Using the lwarp version of package 'lips'.}%
3 \ProvidesPackage{lwarp-lips}
4
5 \NewDocumentCommand{\Lips}{-}{\textellipsis}
6
7 \NewDocumentCommand{\BracketedLips}{-}{[\textellipsis]}
8
9 \let\lips\Lips
10 \let\olips\lips
11
12 \DeclareOption*{}
13 \DeclareOption{mla}{
14 \let\lips\BracketedLips
15 }
16 \ProcessOptions\relax
17
18 \newcommand \LPNobreakList {}
```

## Package 45

# lwarp-listings.sty

## 117 Listings

(Based on original code by CARSTEN HEINZ, BROOKS MOSES, JOBST HOFFMANN.)

`\pkg{listings}` listings is supported with some limitations. Text formatting is not yet supported.

for HTML output: `1 \begin{warpHTML}`

`2 \LWR@ProvidesPackagePass{listings}`

Patches to embed listings inside `pre` tags:

`3 \let\LWR@origlst@Init\lst@Init`

`4 \let\LWR@origlst@DeInit\lst@DeInit`

`5`

`6 \let\LWR@origlsthkEveryPar\lsthk@EveryPar`

`7`

`8 \renewcommand{\l@lstlisting}[2]{\hypertocfloat{1}{\lstlisting}{1ol}{#1}{#2}}`

Done at the start of a listing.

`9 \renewcommand{\lst@Init}[1]{%`

First, perform the listings initialization:

`10 \LWR@traceinfo{\lst@Init}%`

`11 \renewcommand*\@capttype{\lstlisting}%`

`12 \LWR@origlst@Init{#1}%`

`13 \LWR@traceinfo{finished origlst@Init}%`

`14 \lst@ifdisplaystyle%`

Creating a display.

Disable line numbers, produce the `<pre>`, then reenable line numbers.

`15 \LWR@traceinfo{About to create verbatim.}%`

`16 \let\lsthk@EveryPar\relax%`

`17 \LWR@forcenewpage`

`18 \LWR@atbeginverbatim{programlisting}%`

`19`

`20 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%`

`21 \else%`

Inline, so open a `<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 \@tempskipa}%
74 %             \def\lst@vskip{\nobreak\@vskip\@tempskipa\nobreak}%
75 %             \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
76 %             \ifx #1t\allowbreak \fi
77             \ifx\lst@title\@empty

```

New lwarp code to create a caption:

```

78             \lst@makecaption\fnnum\lstlisting{\ignorespaces \lst@caption}
79             \else

```

New lwarp code to create a title:

```

80 %             \lst@maketitle\lst@title % (AS)
81 \LWR@traceinfo{Making title: \lst@title}%
82 \begin{BlockClass}{lstlistingtitle}% lwarp
83 \lst@maketitle\lst@title% lwarp
84 \end{BlockClass}% lwarp

```

```

85             \fi
86 \LWR@traceinfo{About to assign label: !\lst@label!}%
87             \ifx\lst@label\@empty\else
88 \leavevmode% gets rid of bad space factor error
89 \GetTitleStringExpand{\lst@caption}%
90 \edef\LWR@lntemp{\GetTitleStringResult}%
91 \edef\@currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
92 \label{\lst@label}\fi
93 \LWR@traceinfo{Finished assigning the label.}%

```

Not needed for lwarp:

```

94 %             \ifx #1b\allowbreak \fi
95             \endgroup{}%
96     \fi
97 \LWR@traceinfo{end of making a listings display caption}%
98     \else
99 \LWR@traceinfo{INLINE}%
100    \fi
101 \LWR@traceinfo{DONE WITH CAPTION at #1}%
102 }

```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```

103 \lst@Key{numbers}{none}{%
104     \let\lst@PlaceNumber\@empty
105     \lstKV@SwitchCases{#1}%
106     {none&\\%
107         left&\def\lst@PlaceNumber{%
108 % \llap{
109 \LWR@orignormalfont%
110 \lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
111 % }
112 }
113 \\%
114     right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont
115         \kern\VerbatimHTMLWidth \kern\lst@numbersep
116         \lst@numberstyle{\thelstnumber}}}%
117     }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}
118 \end{warpHTML}

```

## Package 46

# lwarp-longtable.sty

## 118 Longtable

Pkg `longtable` `longtable` is emulated during HTML output, and the `longtable` package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{longtable}`

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

`\warpprintonly{row contents}`

instead of

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

Doing so helps avoid “Misplaced `\noalign`.” when using `\begin{warpprint}`.

Keep the `\endfirsthead` row, which is still relevant to HTML output.

⚠ `\kill` is ignored, place a `\kill` line inside

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

or place it inside `\warpingprintonly`.

See:

<http://tex.stackexchange.com/questions/43006/why-is-input-not-expandable>

Env `longtable` `* [⟨horizontal⟩] {⟨colspec⟩}` Emulates the `longtable` environment.

Per the `caption` package, the starred version steps the counter per caption. The unstarred version steps the counter once at the beginning, but not at each caption.

Options `[c]`, `[l]`, and `[r]` are thrown away.

```
2 \newenvironment{longtable*}[2][{}]{%
3 \LWR@floatbegin{table}%
4 \setcaptiontype{LTcaption}%
5 \caption@setoptions{longtable}%
6 \caption@setoptions{@longtable}%
7 \caption@LT@setup%
8 \booltrue{LWR@starredlongtable}%
9 \let\captionlistentry\LWR@LTcaptionlistentry%
10 \LWR@tabular{#2}
11 }
```

```

12 {\endLWR@tabular\LWR@floatend}
13
14 \newenvironment{longtable}[2][\%
15 \LWR@floatbegin{table}%
16 \setcaptiontype{\LTcapttype}%
17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{\LTcapttype}%
21 \let\captionlistentry\LWR@LTcaptionlistentry%
22 \LWR@tabular{#2}
23 }
24 {\endLWR@tabular\LWR@floatend}
25

```

Provided for compatibility, but ignored:

```

26 \newcounter{LTchunksz}
27 \def\endhead{\LWR@tabularendofline}% throws away options //[dim] and /*
28 \def\endfirsthead{\LWR@tabularendofline}
29 \def\endfoot{\LWR@tabularendofline}
30 \def\endlastfoot{\LWR@tabularendofline}
31 \newcommand\tabularnewline{\LWR@tabularendofline}
32 \newcommand{\setlongtables}{}% Obsolete command, does nothing.
33 \newlength{\LTleft}
34 \newlength{\LTRight}
35 \newlength{\LTpre}
36 \newlength{\LTpost}
37 \newlength{\LTcapwidth}

38 \renewcommand*{\kill}{\LWR@tabularendofline}

```

## Package 47

# lwarp-lscape.sty

## 119 Lscape

Pkg lscape lscape is nullified.  
 for HTML output: Discard all options for lwarp-lscape.

```

1 \LWR@ProvidesPackageDrop{lscape}

2 \newenvironment*{landscape}{}{}

```

## Package 48

# lwarp-ltcaption.sty

## 120 Ltcaption

Pkg ltcaption ltcaption is emulated during HTML output, and the ltcaption package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{ltcaption}`

`\LTcapttype` is already defined by lwarp.

`longtable*` is already defined by lwarp-longtable.

```

2 \newlength{\LTcapskip}
3 \newlength{\LTcapleft}
4 \newlength{\LTcapright}
5 \newcommand*{\LTcapmarginsfalse}{}

```

## Package 49

# lwarp-marginfix.sty

## 121 Marginfix

Pkg marginfix Not used.

for HTML output: Discard all options for lwarp-marginfix:

```
1 \LWR@ProvidesPackageDrop{marginfix}

2 \newcommand*{\marginskip}[1]{}
3 \newcommand*{\clearmargin}{}
4 \newcommand*{\softclearmargin}{}
5 \newcommand*{\extendmargin}[1]{}
6 \newcommand*{\mparshift}[1]{}
7 \newdimen\marginheightadjustment
8 \newdimen\marginposadjustment
9 \newcommand*{\blockmargin}[1][1]{}
10 \newcommand*{\unblockmargin}[1][1]{}
11 \newcommand*{\marginphantom}[2][1]{}

```

## Package 50

# lwarp-marginnote.sty

## 122 Marginnote

Pkg marginnote Emulated.

for HTML output: Discard all options for lwarp-marginnote:

```

1 \LWR@ProvidesPackageDrop{marginnote}

2 \NewDocumentCommand{\marginnote}{o +m o}{\marginpar{#2}}
3 \newcommand*{\marginnoteleftadjust}{}
4 \newcommand*{\marginnoterightadjust}{}
5 \newcommand*{\marginnotetextwidth}{}
6 \let\marginnotetextwidth\textwidth
7 \newcommand*{\marginnotevadjust}{}
8 \newcommand*{\marginfont}{}
9 \newcommand*{\raggedleftmarginnote}{}
10 \newcommand*{\raggedrightmarginnote}{}

```

## Package 51

# lwarp-mcaption.sty

## 123 Mcaption

Pkg mcaption mcaption is nullified.

for HTML output: Discard all options for lwarp-mcaption:

```

1 \LWR@ProvidesPackageDrop{mcaption}

2 \newenvironment{margincap}{}{}
3 \newcommand*{\margincapalign}{}
4 \newlength{\margincapsep}

```

## Package 52

# lwarp-mdframed.sty

## 124 Mdfamed

**Pkg** **mdframed** **mdframed** is loaded with options forced to **framemethod=none**.

**for HTML output:** `1 \LWR@ProvidesPackageDrop{mdframed}`

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

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

**font** For title font, use

`frametitlefont=\textbf,`

instead of

`frametitlefont=\bfseries,`

where `\textbf` must appear just before the comma and will receive the following text as its argument (since the text happens to be between braces in the **mdframed** source). Since **lwarp** does not support `\bfseries` and friends, only one font selection may be made at a time.

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

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

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

**CSS classes** Environments created or encapsulated by **mdframed** are enclosed in a `<div>` of class **md<environmentname>**, or **mdframed** otherwise.

Frame titles are placed into a `<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**

`2 \LWR@origRequirePackage{amsthm}`



Do not require *Tikz* or *pstricks*:

```
3 \LWR@origRequirePackage[framemethod=none]{mdframed}
```

To handle CSS and paragraphs, patch code at start and end of environment and contents. `\LWR@origraggedright` helps avoid hyphenation.

```
4 \mdfsetup{
5 startcode={\LWR@mdframedstart\LWR@origraggedright},
6 endcode={\LWR@mdframedend},
7 startinnercode={\LWR@startpars\LWR@origraggedright},
8 endinnercode={\LWR@stoppars},
9 }
```

Given the `mdframed` key, print the color.

```
10 \newcommand*{\LWR@mdfprintcolor}[1]{%
11 \convertcolorspec{named}{\csuse{mdf@#1}}{HTML}\LWR@tempcolor%
12 \#\LWR@tempcolor
13 }
```

Given the `mdframed` key, print the length.

```
14 \newcommand*{\LWR@mdfprintlength}[1]{%
15 \rndprintlength{\csuse{mdf@#1@length}}
16 }
```

Actions before an `mdframe` starts.

Encapsulate a frame inside a `<div>` of the desired `class`.

```
17 \newcommand*{\LWR@mdframedstart}{%
```

Turn off paragraph handling during the generation of the encapsulating tags:

```
18 \LWR@stoppars%
```

Below, print HTML pt units:

```
19 \uselengthunit{PT}%
```

Open a `<div>` and with custom `class` and custom `style`:

```
20 \LWR@htmltagc{div class="\LWR@mdthisenv" \LWR@orignewline
21 style=" \LWR@orignewline
```

Convert and print the background color:

```
22 background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```
23 border: \LWR@mdfprintlength{linewidth} solid
24 \LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
25 border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
26 \ifbool{mdf@shadow}{%
27 box-shadow:
28 \LWR@mdfprintlength{shadowsize}
29 \LWR@mdfprintlength{shadowsize}
30 \LWR@mdfprintlength{shadowsize}
31 \LWR@mdfprintcolor{shadowcolor} ;
32 }
33 {box-shadow: none ;}
34 \LWR@orignewline

35 "}
36 % \LWR@htmldivclass{\LWR@mdthisenv}
```

`mdframed` environment may not work with the modified `\hspace` and `\rule`, so restore them to their originals while inside `mdframed`:

```
37 \let\hspace\LWR@orighspace%
38 \let\rule\LWR@origrule%
39 }
```

Actions after an `mdframe` ends.

After closing the `<div>`, globally restore to the default environment type:

```
40 \newcommand*{\LWR@mdframedend}{}
```

Close the custom `<div>`:

```
41 \LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
42 \gdef\LWR@mdthisenv{mdframed}
```

Resume paragraph handling:

```
43 \LWR@startpars%
44 }
```

Encapsulation of the original which places the title inside a `<span>` of class `mdframedtitle`:

```
45 \let\LWR@origmdfframedtitleenv\mdfframedtitleenv
46
47 \newlength{\LWR@titleroundcorner}
48
49 \renewrobustcmd\mdfframedtitleenv[1]{%
50 \LWR@origmdfframedtitleenv{%
```

Below, print HTML pt lengths:

```
51 \uselengthunit{PT}%
```

Open a `<span>` with a custom class and custom style:

```
52 \LWR@htmltagc{span class="mdframedtitle" \LWR@orignewline
53 style=" \LWR@orignewline
```

Convert and print the title background color:

```
54 background:
55 \LWR@mdfprintcolor{frametitlebackgroundcolor}
56 ; \LWR@orignewline
```

Convert and print the title rule:

```
57 \ifbool{mdf@frametitlerule}{%
58 border-bottom:
59 \LWR@mdfprintlength{frametitlerulewidth}
60 solid
61 \LWR@mdfprintcolor{frametitlerulecolor}
62 ; \LWR@orignewline
63 }{}%
```

The title's top border radius is adjusted for the line width:

```
64 border-radius:
65 \setlength{\LWR@titleroundcorner}
66 {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{0pt}}
67 \rndprintlength{\LWR@titleroundcorner}
68 \rndprintlength{\LWR@titleroundcorner}
69 0pt 0pt
70 \LWR@orignewline
```

Finish the custom style and the opening span tag:

```
71 " \LWR@orignewline
72 }% span
```

Restrict paragraph tags inside a span:

```
73 \begin{LWR@nestspan}%
```

Print the title inside the span:

```
74 #1%
```

Close the span and unnest the paragraph tag restriction:

```
75 \LWR@htmltagc{/span}%
```

```
76 \end{LWR@nestspan}%
```

```
77 }
```

```
78 }
```

Common code for `\LWR@mdfsubtitle` and `\LWR@mdfsubsubtitle`.

Encapsulate the subtitle inside a `<span>` of class `mdframedsubtitle`:

```
79 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
```

```
80 {% the following empty line is required
```

```
81
```

Special handling for `mdframed`: Subtitles have `\pars` around them, so temporarily disable them here.

```
82 \let\par\LWR@origpar%
```

Open a `<span>` with a custom class and custom style:

```
83 \LWR@htmltagc{span class="mdframed#1title"
```

```
84 style=" \LWR@orignewline
```

Convert and print the background color:

```
85 background:
```

```
86 \LWR@mdfprintcolor{#1titlebackgroundcolor}
```

```
87 ; \LWR@orignewline
```

Convert and print the above line:

```
88 \ifbool{mdf@#1titleaboveline}{%
```

```
89 border-top:
```

```
90 \LWR@mdfprintlength{#1titleabovelinewidth}
```

```
91 solid
```

```
92 \LWR@mdfprintcolor{#1titleabovelinecolor}
```

```
93 ; \LWR@orignewline
```

```
94 }{}%
```

Convert and print the below line:

```

95 \ifbool{mdf@#1titlebelowline}{%
96 border-bottom:
97 \LWR@mdfprintlength{#1titlebelowlinewidth}
98 solid
99 \LWR@mdfprintcolor{#1titlebelowlinecolor}
100 ; \LWR@orignewline
101 }{}%
```

Finish the custom style and the opening span tag:

```

102 "% span
```

Restrict paragraph tags inside a span:

```

103 \begin{LWR@nestspan}%
```

Perform the original subtitle action:

```

104 \IfNoValueTF{#2}
105 {\csuse{LWR@origmdf#1title}{#3}}%
106 {\csuse{LWR@origmdf#1title}[#2]{#3}}%
```

Close the span and unnest the paragraph tag restriction:

```

107 \LWR@htmltagc{/span}% the following empty line is required
108 \end{LWR@nestspan}% must follow the /span or an extra <p> appears
109
110 }
```

```

111 \let\LWR@origmdfsubtitle\mdfsubtitle
112
113 \newcommand*{\LWR@mdfsubtitle}{%
114 \LWR@mdfsubtitlecommon{sub}%
115 }
116 \let\mdfsubtitle\LWR@mdfsubtitle

117 \let\LWR@origmdfsubsubtitle\mdfsubsubtitle
118
119 \newcommand*{\LWR@mdfsubsubtitle}{%
120 \LWR@mdfsubsubtitlecommon{subsub}%
121 }
122 \let\mdfsubsubtitle\LWR@mdfsubsubtitle
```

Stores the environment of the frame about to be created:

```

123 \newcommand*{\LWR@mdthisenv}{mdframed}
```

Modified from the original to remember the environment.

```

124 \renewrobustcmd*\newmdenv[2] [] {%
125 \newenvironment{#2}%
126 {%
127 \mdfsetup{#1}%
128 \renewcommand*\LWR@mdthisenv}{md#2}%
129 \begin{mdframed}%
130 }
131 {\end{mdframed}}%
132 }

```

Modified from the original to remember the environment.

```

133 \renewrobustcmd*\surroundwithmdframed[2] [] {%
134 \BeforeBeginEnvironment{#2}{%
135 \renewcommand*\LWR@mdthisenv}{md#2}%
136 \begin{mdframed}[#1]}%
137 \AfterEndEnvironment{#2}{\end{mdframed}}%
138 }

```

[*numberedlike*] {*caption*} [*within*]

Modified from the original to remember the environment.

```

139 \let\LWR@origmdtheorem\mdtheorem
140
141 \DeclareDocumentCommand{\LWR@mdtheorem}{0}{ m o m o }{%
142 \LWR@origmdtheorem[#1]{#2}[#3]{#4}[#5}%
143 \BeforeBeginEnvironment{#2}{\renewcommand*\LWR@mdthisenv}{md#2}}%
144 }
145
146 \let\mdtheorem\LWR@mdtheorem

```

[*numberedlike*] {*caption*} [*within*]

Modified from the original to remember the environment.

```

147 \DeclareDocumentCommand\newmdtheoremenv{0}{ m o m o }{%
148 \ifboolexpr{ test {\IfNoValueTF {#3}} and test {\IfNoValueTF {#5}} }%
149   {\newtheorem{#2}{#4}}{%
150     \IfValueTF{#3}{\newtheorem{#2}[#3]{#4}}{%
151       \IfValueTF{#5}{\newtheorem{#2}{#4}[#5]}{%
152         }%
153     }%
154   \BeforeBeginEnvironment{#2}{%
155     \renewcommand*\LWR@mdthisenv}{md#2}%
156     \begin{mdframed}[#1]}%
157   \AfterEndEnvironment{#2}{%
158     \end{mdframed}}%
159 }

```

## Package 53

# lwarp-microtype.sty

## 125 Microtype

Pkg microtype microtype is pre-loaded by lwarp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for lwarp-microtype:

```

1 \LWR@ProvidesPackageDrop{microtype}

2 \DeclareDocumentCommand{\DeclareMicrotypeSet}{o m m}{}
3 \DeclareDocumentCommand{\UseMicrotypeSet}{o m}{}
4 \DeclareDocumentCommand{\DeclareMicrotypeSetDefault}{o m}{}
5 \DeclareDocumentCommand{\SetProtrusion}{o m m}{}
6 \DeclareDocumentCommand{\SetExpansion}{o m m}{}
7 \DeclareDocumentCommand{\SetTracking}{o m m}{}
8 \DeclareDocumentCommand{\SetExtraKerning}{o m m}{}
9 \DeclareDocumentCommand{\SetExtraSpacing}{o m m}{}
10 \DeclareDocumentCommand{\DisableLigatures}{o m}{}
11 \DeclareDocumentCommand{\DeclareCharacterInheritance}{o m m}{}
12 \DeclareDocumentCommand{\DeclareMicrotypeVariants}{m}{}
13 \DeclareDocumentCommand{\DeclareMicrotypeAlias}{m m}{}
14 \DeclareDocumentCommand{\LoadMicrotypeFile}{m}{}
15 \DeclareDocumentCommand{\DeclareMicrotypeBabelHook}{m m}{}
16 \DeclareDocumentCommand{\microtypesetup}{m}{}
17 \DeclareDocumentCommand{\microtypecontext}{m}{}
18 \DeclareDocumentCommand{\textmicrotypecontext}{m m}{#2}
19 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
20 \DeclareDocumentCommand{\lsstyle}{}{}
21 \DeclareDocumentCommand{\textls}{o +m}{}
22 \DeclareDocumentCommand{\slig}{m}{#1}
23 }
24 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
25 \def\DeclareMicrotypeVariants#1#{\@gobble}
26 \@onlypreamble\DeclareMicrotypeSet
27 \@onlypreamble\UseMicrotypeSet
28 \@onlypreamble\DeclareMicrotypeSetDefault
29 \@onlypreamble\DisableLigatures
30 \@onlypreamble\DeclareMicrotypeVariants
31 \@onlypreamble\DeclareMicrotypeBabelHook

```

## Package 54

# lwarp-mparhack.sty

## 126 Mparhack

Pkg mparhack Not used.

for **HTML output:** Discard all options for lwarp-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```



## Package 55

# lwarp-multicol.sty

## 127 Multicol

Pkg multicol multicol is emulated during HTML output, and the multicol package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{multicol}[2015/09/13]

Multicols are converted into a 1–3 column display, browser-supported.

The optional multicols heading is placed inside a <div> of class multicolsheading.

The content is placed inside a <div> of class multicols.

```
2 \begin{warpHTML}
```

```
3 \NewDocumentEnvironment{multicols}{s m o}
```

HTML div class to contain everything:

```
4 {
5 \LWR@forcenewpage
6 \BlockClass{multicols}
```

Optional HTML div class for the heading:

```
7 \IfValueTF{#3}{\begin{BlockClass}{multicolsheading}#3\end{BlockClass}}{}}
```

When done with the environment, close the div:

```
8 {\endBlockClass}
```

Emulated null functions which are not used in HTML:

```
9 \newcommand*{\columnbreak}{}
10 \newcommand*{\RLmulticolcolumns}{}
11 \newcommand*{\LRmulticolcolumns}{}
12
13 \newlength{\premulticols}
14 \newlength{\postmulticols}
15 \newlength{\multicolsep}
16 \newlength{\multicolbaselineskip}
17 \newlength{\multicoltolerance}
```

---

```
18 \newlength{\multicolpretolerance}
19 \newcommand*{\columnseprulecolor}{\normalcolor}
20 \newcounter{columnbadness}
21 \newcounter{finalcolumnbadness}
22 \newcounter{collectmore}
23 \newcounter{unbalance}
24 \newlength{\multicolovershoot}
25 \newlength{\multicolundershoot}

26 \end{warpHTML}
```

## Package 56

# lwarp-multirow.sty

## 128 Multirow

Pkg multirow multirow is emulated during HTML output, and the multirow package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{multirow}

## Package 57

# lwarp-nameref.sty

## 129 Nameref

Pkg nameref nameref is emulated by lwarp.

for HTML output: Discard all options for lwarp-nameref:

```
1 \typeout{Using the lwarp html version of package 'nameref' -- discarding options.}
2 \typeout{   Are not using ProvidesPackage, so that other packages}
3 \typeout{   do not attempt to patch lwarp's version of 'nameref'.}
4 \DeclareOption*{}
5 \ProcessOptions\relax
```

## Package 58

# lwarp-needspace.sty

## 130 Needspace

Pkg `needspace` needspace is not used during HTML conversion.

for HTML output: Discard all options for lwarp-needspace:

```
1 \LWR@ProvidesPackageDrop{needspace}
2
3 \newcommand*{\needspace}[1]{ }
4 \DeclareDocumentCommand{\Needspace}{s m}{ }
```

## Package 59

# lwarp-newclude.sty

## 131 Newclude

Pkg `newclude` Error if newclude is loaded after lwarp.

Discard all options for lwarp-newclude:

for HTML output: 

```
1 \LWR@ProvidesPackageDrop{newclude}

2 \LWR@loadbefore{newclude}
```

## Package 60

# lwarp-newunicodechar.sty

## 132 Newunicodechar

Pkg newunicodechar Error if newunicodechar is loaded after lwarp.  
Discard all options for lwarp-newunicodechar:

for HTML output: 1 \LWR@ProvidesPackageDrop{newunicodechar}

2 \LWR@loadbefore{newunicodechar}

## Package 61

# lwarp-nextpage.sty

## 133 Nextpage

Pkg nextpage nextpage is nullified.

for HTML output: Discard all options for lwarp-nextpage.

1 \LWR@ProvidesPackageDrop{nextpage}

2 \newcommand{\cleartoevenpage}[1] [] {}

3 \newcommand{\movetoevenpage}[1] [] {}

4 \newcommand{\cleartooddpage}[1] [] {}

5 \newcommand{\movetooddpage}[1] [] {}

## Package 62

# lwarp-nowidow.sty

## 134 Nowidow

Pkg `nowidow` `nowidow` is not used during HTML conversion.

Discard all options for `lwarp-nowidow`:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{nowidow}

2 \newcommand*{\nowidow}[1] [] {}
3 \newcommand*{\setnowidow}[1] [] {}

4 \newcommand*{\noclub}[1] [] {}
5 \newcommand*{\setnoclub}[1] [] {}
```

## Package 63

# lwarp-ntheorem.sty

## 135 Ntheorem

(Based on original code by WOLFGANG MAY, ANDREAS SCHEDLER.)


Pkg `ntheorem` `ntheorem` is patched for use by `lwarp`.


### CSS styling of theorems and proofs:

**Theorem:** `<div>` of class `theorembody<theoremstyle>`


**Theorem Header:** `<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 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@intheorem% new
104     }#2}%
105     \def\@opargbegintheorem####1####2####3{%
106       \LWR@forcenewpage% new
107       \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
108       \LWR@intheorem% new
109     }#3}%
110   }%
111 }%
112 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
113 }
114
115 \renewtheoremstyle{plain}%
116   {\item[\hskip\labelsep \theorem@headerfont
117 \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
118   {\item[\hskip\labelsep \theorem@headerfont
119 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
120
121 \renewtheoremstyle{break}%
122   {\item[
123 % \rlap{\vbox{\hbox{
124 \hskip\labelsep \theorem@headerfont
125 \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
126 % } \hbox{\strut}}}}
127 ]}%
128   {\item[
129 % \rlap{\vbox{\hbox{
130 \hskip\labelsep \theorem@headerfont
131 \InlineClass{theoremheaderbreak}{##1\ ##2\ (##3)\theorem@separator}\newline

```

---

```

132 % }\hbox{\strut}}
133 ]]
134
135 \renewtheoremstyle{change}%
136 {\item[\hskip\labelsep
137 \theorem@headerfont
138 \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]]}%
139 {\item[\hskip\labelsep
140 \theorem@headerfont
141 \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]]}
142
143 \renewtheoremstyle{changebreak}%
144 {\item[
145 % \rlap{\vbox{\hbox{
146 \hskip\labelsep \theorem@headerfont
147 \InlineClass{theoremheaderchangebreak}{##2\ ##1\theorem@separator}\newline
148 % }\hbox{\strut}}}}
149 ]}%
150 {\item[
151 % \rlap{\vbox{\hbox{
152 \hskip\labelsep \theorem@headerfont
153 \InlineClass{theoremheaderchangebreak}{##2\ ##1\ (##3)\theorem@separator}\newline
154 % }\hbox{\strut}}}}
155 ]}
156
157 \renewtheoremstyle{margin}%
158 {\item[\hskip\labelsep\theorem@headerfont
159 \InlineClass{theoremheadermargin}{##2 \quad ##1\theorem@separator}
160 ]}%
161 {\item[\hskip\labelsep\theorem@headerfont
162 \InlineClass{theoremheadermargin}{##2 \quad ##1\ (##3)\theorem@separator}
163 ]}
164
165 \renewtheoremstyle{marginbreak}%
166 {\item[\hskip\labelsep\theorem@headerfont
167 \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\theorem@separator}\newline
168 ]}%
169 {\item[\hskip\labelsep\theorem@headerfont
170 \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\ (##3)\theorem@separator}\newline
171 ]}
172
173 \renewtheoremstyle{nonumberplain}%
174 {\item[\theorem@headerfont\hskip\labelsep
175 \InlineClass{theoremheaderplain}{##1\theorem@separator}]]}%
176 {\item[\theorem@headerfont\hskip\labelsep
177 \InlineClass{theoremheaderplain}{##1\ (##3)\theorem@separator}]]}
178
179 \renewtheoremstyle{nonumberbreak}%
180 {\item[
181 % \rlap{\vbox{\hbox{

```

```

182 \hskip\labelsep \theorem@headerfont
183 \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline
184 % }\hbox{\strut}}
185 ]}%
186 {\item[
187 % \rlap{\vbox{\hbox{
188 \hskip\labelsep \theorem@headerfont
189 \InlineClass{theoremheaderbreak}{##1\ (##3)\theorem@separator}\newline
190 % }\hbox{\strut}}
191 ]}
192
193 \renewtheoremstyle{empty}%
194 {\item[]}%
195 {\item[\theorem@headerfont \hskip\labelsep\relax
196 \InlineClass{theoremheaderplain}{##3}]}
197
198 \renewtheoremstyle{emptybreak}%
199 {\item[]}%
200 {\item[\theorem@headerfont \hskip\labelsep\relax
201 \InlineClass{theoremheaderplain}{##3}] \ \newline}

```

The following manually adjust the CSS for the standard configuration objects which are not a purely plain style:

```

202 \ifbool{LWR@ntheoremamsthm}{-}{-}
203 % upright text via CSS
204 \newtheoremstyle{plainupright}%
205 {\item[\hskip\labelsep \theorem@headerfont
206 \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
207 {\item[\hskip\labelsep \theorem@headerfont
208 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
209
210 % upright text and small caps header via CSS
211 \newtheoremstyle{nonumberplainuprightsc}%
212 {\item[\theorem@headerfont\hskip\labelsep
213 \InlineClass{theoremheadersc}{##1\theorem@separator}]}%
214 {\item[\theorem@headerfont\hskip \labelsep
215 \InlineClass{theoremheadersc}{##1\ (##3)\theorem@separator}]}

```

The following standard configuration is renewed using the new CSS:

```

216 \theoremstyle{plainupright}
217 \theorembodyfont{\upshape}
218 \theoremsymbol{\ensuremath{\_ \Box}}
219 \renewtheorem{Example}{Example}
220 \renewtheorem{example}{Example}
221 \renewtheorem{Beispiel}{Beispiel}
222 \renewtheorem{beispiel}{Beispiel}
223 \renewtheorem{Bemerkung}{Bemerkung}

```

```

224 \renewtheorem{bemerkung}{Bemerkung}
225 \renewtheorem{Anmerkung}{Anmerkung}
226 \renewtheorem{anmerkung}{Anmerkung}
227 \renewtheorem{Remark}{Remark}
228 \renewtheorem{remark}{Remark}
229 \renewtheorem{Definition}{Definition}
230 \renewtheorem{definition}{Definition}
231
232 \theoremstyle{nonumberplainuprightsc}
233 \theoremsymbol{\ensuremath{\_\blacksquare}}
234 \renewtheorem{Proof}{Proof}
235 \renewtheorem{proof}{Proof}
236 \renewtheorem{Beweis}{Beweis}
237 \renewtheorem{beweis}{Beweis}
238 \qedsymbol{\ensuremath{\_\blacksquare}}
239
240 \theoremsymbol{}
241 }% not amsthm

```

Only if the `amsthm` option was given:

```

242 \ifbool{LWR@theoremamsthm}{
243
244 \gdef\th@plain{%
245   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
246   \def\@begintheorem##1##2{%
247     \LWR@forcenewpage% new
248     \BlockClass{theorembodyplain}% new
249     \LWR@inctheorem% new
250     \item[\hskip\labelsep
251 %   \theorem@headerfont
252     \InlineClass{theoremheaderplain}{##1\ ##2.}
253   ]}%
254   \def\@opargbegintheorem##1##2##3{%
255     \LWR@forcenewpage% new
256     \BlockClass{theorembodyplain}% new
257     \LWR@inctheorem% new
258     \item[\hskip\labelsep
259 %   \theorem@headerfont
260     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
261   ]}}
262
263 \gdef\th@nonumberplain{%
264   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
265   \def\@begintheorem##1##2{%
266     \LWR@forcenewpage% new
267     \BlockClass{theorembodyplain}% new
268     \LWR@inctheorem% new
269     \item[\hskip\labelsep
270 %   \theorem@headerfont

```

```

271 \InlineClass{theoremheaderplain}{##1.}
272 ]}%
273 \def\@opargbegintheorem##1##2##3{%
274 \LWR@forcenewpage% new
275 \BlockClass{theorembodyplain}% new
276 \LWR@inctheorem% new
277 \item[\hskip\labelsep
278 % \theorem@headerfont
279 \InlineClass{theoremheaderplain}{##1\ (##3).}
280 ]}}
281
282 \gdef\th@definition{%
283 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
284 \def\@begintheorem##1##2{%
285 \LWR@forcenewpage% new
286 \BlockClass{theorembodydefinition}% new
287 \LWR@inctheorem% new
288 \item[\hskip\labelsep
289 % \theorem@headerfont
290 \InlineClass{theoremheaderdefinition}{##1\ ##2.}
291 ]}%
292 \def\@opargbegintheorem##1##2##3{%
293 \LWR@forcenewpage% new
294 \BlockClass{theorembodydefinition}% new
295 \LWR@inctheorem% new
296 \item[\hskip\labelsep
297 % \theorem@headerfont
298 \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
299 ]}}
300
301 \gdef\th@nonumberdefinition{%
302 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
303 \def\@begintheorem##1##2{%
304 \LWR@forcenewpage% new
305 \BlockClass{theorembodydefinition}% new
306 \LWR@inctheorem% new
307 \item[\hskip\labelsep
308 % \theorem@headerfont
309 \InlineClass{theoremheaderdefinition}{##1.}
310 ]}%
311 \def\@opargbegintheorem##1##2##3{%
312 \LWR@forcenewpage% new
313 \BlockClass{theorembodydefinition}% new
314 \LWR@inctheorem% new
315 \item[\hskip\labelsep
316 % \theorem@headerfont
317 \InlineClass{theoremheaderdefinition}{##1\ (##3).}
318 ]}}
319
320 \gdef\th@remark{%

```

```

321 \def\theorem@headerfont{\itshape}\normalfont%
322 \def\@begintheorem##1##2{%
323 \LWR@forcenewpage% new
324 \BlockClass{theorembodyremark}% new
325 \LWR@intheorem% new
326 \item[\hskip\labelsep
327 % \theorem@headerfont
328 \InlineClass{theoremheaderremark}{##1\ ##2.}
329 ]}%
330 \def\@opargbegintheorem##1##2##3{%
331 \LWR@forcenewpage% new
332 \BlockClass{theorembodyremark}% new
333 \LWR@intheorem% new
334 \item[\hskip\labelsep
335 % \theorem@headerfont
336 \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
337 ]}}
338
339 \gdef\th@nonumberremark{%
340 \def\theorem@headerfont{\itshape}\normalfont%
341 \def\@begintheorem##1##2{%
342 \LWR@forcenewpage% new
343 \BlockClass{theorembodyremark}% new
344 \LWR@intheorem% new
345 \item[\hskip\labelsep
346 % \theorem@headerfont
347 \InlineClass{theoremheaderremark}{##1.}
348 ]}%
349 \def\@opargbegintheorem##1##2##3{%
350 \LWR@forcenewpage% new
351 \BlockClass{theorembodyremark}% new
352 \LWR@intheorem% new
353 \item[\hskip\labelsep
354 % \theorem@headerfont
355 \InlineClass{theoremheaderremark}{##1\ (##3).}
356 ]}}
357
358 \gdef\th@proof{%
359 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
360 \def\@begintheorem##1##2{%
361 \LWR@forcenewpage% new
362 \BlockClass{theorembodyproof}% new
363 \LWR@intheorem% new
364 \item[\hskip\labelsep
365 % \theorem@headerfont
366 \InlineClass{theoremheaderproof}{##1.}
367 ]}%
368 \def\@opargbegintheorem##1##2##3{%
369 \LWR@forcenewpage% new
370 \BlockClass{theorembodyproof}% new

```

---

```

371 \LWR@inctheorem% new
372     \item[\hskip\labelsep
373 % \theorem@headerfont
374 \InlineClass{theoremheaderproof}{##1\ (##3).}
375 ]}}
376
377
378
379 \newcounter{proof}%
380 \if@thmmarks
381 \newcounter{currproofctr}%
382 \newcounter{endproofctr}%
383 \fi
384
385 \gdef\proofSymbol{\openbox}
386
387 \newcommand{\proofname}{Proof}
388
389 \newenvironment{proof}[1][\proofname]{
390 \th@proof
391 \def\theorem@headerfont{\itshape}%
392 \normalfont
393 \theoremsymbol{\ensuremath{\_}\blacksquare}}
394 \@thm{proof}{proof}{#1}
395 }%
396 {\@endtheorem}
397
398 }{}% amsthm option

```

Patched for CSS:

```

399 \let\LWR@origendtheorem\@endtheorem
400 \renewcommand{\@endtheorem}{%
401 \ifbool{LWR@theoremmarks}{%
402 \ifsetendmark%
403 \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
404 \setendmarkfalse%
405 \fi%
406 }{}%
407 \LWR@origendtheorem%
408 \ifbool{LWR@theoremmarks}{\global\setendmarktrue}{}%
409 \endBlockClass%
410 }

411 \gdef\NoEndMark{\global\setendmarkfalse}

```

Redefined to reuse the float mechanism to add list-of-theorem links:



`\thm@thmline {⟨1: printed type⟩} {⟨2: #⟩} {⟨3: optional⟩} {⟨4: page⟩}`

```
412 \renewcommand{\thm@@thmline@noname}[4]{%
413 \hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
414 }
415
416 \renewcommand{\thm@@thmline@name}[4]{%
417 \hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
418 }
```

This was redefined by `ntheorem` when loaded, so it is now redefined for `lwarp`:

```
419 \def\thm@@thmline{\thm@@thmline@name}
```

Patch for CSS:

```
420 \def\listtheorems#1{
421 \LWR@html@elementclass{nav}{lothm}%
422 \begingroup
423 \c@tocdepth=-2%
424 \def\thm@list{#1}\thm@processlist
425 \endgroup
426 \LWR@html@elementclassend{nav}{lothm}%
427 }
```

Proof QED symbol:

```
428
429 \newcommand{\qed}{\qquad\the\qedsymbol}
430
431 \AtBeginDocument{
432 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
433 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
434 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
435 }
```

`\thref {⟨label⟩}`

```
436 \newcommand*{\thref}[1]{\cref{#1}}
```

## Package 64

# lwarp-pagenote.sty

## 136 Pagenote

Pkg pagenote pagenote works as-is.

It is only included as an `lwarp-pagenote.sty` file because past versions of `lwarp` used `pagenote` to emulate footnotes, and so the file may exist on current installations, and should be over-written by this newer version.

for HTML output: 1 \LWR@ProvidesPackagePass{pagenote}

## Package 65

# lwarp-parskip.sty

## 137 Parskip

Pkg parskip parskip is ignored.

for HTML output: Discard all options for lwarp-parskip.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

## Package 66

# lwarp-placeins.sty

## 138 Placeins

Pkg placeins placeins is not used during HTML conversion.

Discard all options for lwarp-placeins:

for HTML output: 1 \LWR@ProvidesPackageDrop{placeins}

```
2 \newcommand*{\FloatBarrier}{} 
```

## Package 67

# lwarp-ragged2e.sty

## 139 Ragged2e

Pkg ragged2e ragged2e is not used during HTML conversion.

Discard all options for lwarp-ragged2e:

```
for HTML output: 1 \LWR@ProvidesPackageDrop{ragged2e}

2 \newcommand*{\Centering}{\centering}
3 \newcommand*{\RaggedLeft}{\raggedleft}
4 \newcommand*{\RaggedRight}{\raggedright}
5 \newcommand*{\justifying}{\}
6 \newlength{\CenteringLeftskip}
7 \newlength{\RaggedLeftLeftskip}
8 \newlength{\RaggedRightLeftskip}
9 \newlength{\CenteringRightskip}
10 \newlength{\RaggedLeftRightskip}
11 \newlength{\RaggedRightRightskip}
12 \newlength{\CenteringParfillskip}
13 \newlength{\RaggedLeftParfillskip}
14 \newlength{\RaggedRightParfillskip}
15 \newlength{\JustifyingParfillskip}
16 \newlength{\CenteringParindent}
17 \newlength{\RaggedLeftParindent}
18 \newlength{\RaggedRightParindent}
19 \newlength{\JustifyingParindent}
20 \newenvironment*{Center}{\center}{\endcenter}
21 \newenvironment*{FlushLeft}{\flushleft}{\endflushleft}
22 \newenvironment*{FlushRight}{\flushright}{\endflushright}
23 \newenvironment*{justify}{\justifying}{\endjustifying}
```

## Package 68

# lwarp-rotating.sty

## 140 Rotating

Pkg rotating rotating is emulated during HTML output, and the rotating package is ignored.

All rotations are ignored in HTML output.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{rotating}

2 \let\sidewaystable\table
3 \let\endsidewaystable\endtable
4
5 \let\sidewaysfigure\figure
6 \let\endsidewaysfigure\endfigure
7
8 \newenvironment*{sideways}{}{}
9 \newenvironment*{turn}[1]{}{}
10 \newenvironment*{rotate}[1]{}{}
11 \NewDocumentCommand{\turnbox}{m +m}{#2}
12 \let\rotcaption\caption
13 \let\@makerotcaption\@makecaption
```

## Package 69

# lwarp-setspace.sty

## 141 Setspace

Pkg    **setspace**    setspace is not used during HTML conversion.

Discard all options for lwarp-setspace:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{setspace}
2
3 \newcommand*{\setstretch}[1]{}
4 \newcommand*{\SetSinglespace}[1]{}
5 \newcommand*{\singlespacing}{}
6 \newcommand*{\onehalfspacing}{}
7 \newcommand*{\doublespacing}{}
8
9 \newenvironment*{singlespace}
10 {
11 \LWR@forcenewpage
12 \BlockClass{singlespace}
13 }
14 {\endBlockClass}
15
16 \newenvironment*{singlespace*}
17 {
18 \LWR@forcenewpage
19 \BlockClass{singlespace}
20 }
21 {\endBlockClass}
22
23 \newenvironment*{spacing}[1]{
24
25 }{
26
27 }
28
29 \newenvironment*{onehalfspace}
30 {
31 \LWR@forcenewpage
32 \BlockClass{onehalfspace}
33 }
34 {\endBlockClass}
35
36 \newenvironment*{doublespace}
37 {

```

```
38 \LWR@forcenewpage
39 \BlockClass{doublespace}
40 }
41 {\endBlockClass}
```

clearpage

## Package 70

# lwarp-showidx.sty

## 142 Showidx

Pkg `showidx` `showidx` is ignored.

for HTML output: Discard all options for `lwarp-showidx`:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

## Package 71

# lwarp-showkeys.sty

## 143 Showkeys

Pkg `showkeys` `showkeys` is ignored.

for HTML output: Discard all options for `lwarp-showkeys`:

```
1 \LWR@ProvidesPackageDrop{showkeys}
```

```
2 \NewDocumentCommand{\showkeys}{s}{}
```

## Package 72

# lwarp-sidecap.sty

## 144 Sidecap

Pkg sidecap sidecap is nullified.

for HTML output: Discard all options for lwarp-sidecap.

```
1 \LWR@ProvidesPackageDrop{sidecap}
```

See:

<http://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment>  
regarding the creation of starred environments with xparse.

```
2 \NewDocumentEnvironment{SCtable}{soo}
3 {\IfValueTF{#3}{\table[#3]}\table}}
4 {\endtable}
5
6 \ExplSyntaxOn
7 \cs_new:cpn {SCtable*} {\SCtable*}
8 \cs_new_eq:cN {endSCtable*} \endSCtable
9 \ExplSyntaxOff
10
11
12 \NewDocumentEnvironment{SCfigure}{soo}
13 {\IfValueTF{#3}{\figure[#3]}\figure}}
14 {\endfigure}
15
16 \ExplSyntaxOn
17 \cs_new:cpn {SCfigure*} {\SCfigure*}
18 \cs_new_eq:cN {endSCfigure*} \endSCfigure
19 \ExplSyntaxOff
20
21
22 \newenvironment*{wide}{}{}
```



## Package 73

# lwarp-sidenotes.sty

## 145 Sidenotes

(Based on original code by ANDY THOMAS, OLIVER SCHEBAUM.)

Pkg sidenotes Patched for lwarp.

for HTML output: Load the original package:

```
1 \LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with lwarp:

Stop paragraph handling while creating the caption:

```
2 \RenewDocumentCommand \sidecaption {s o o m}
3 {
4   \LWR@stoppars
5   \captionsetup{style=sidecaption}
6   \IfBooleanTF{#1}
7   { % starred
8     \IfNoValueOrEmptyTF{#2}
9     {\marginnote{\caption*{#4}}}
10    {\marginnote{\caption*{#4}}[#2]}
11  }
12  { % unstarred
13    \IfNoValueOrEmptyTF{#2}
14    {\def\@sidenotes@sidecaption@tof{#4}}
15    {\def\@sidenotes@sidecaption@tof{#2}}
16    \IfNoValueOrEmptyTF{#3}
17    {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}}
18    {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}[#3]}
19  }
20 \LWR@startpars
21 }
```

Borrowed from the lwarp version of keyfloat:

```
22 \NewDocumentEnvironment{KFLT}sidenotes@marginfloat{-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 74

# lwarp-soul.sty

## 146 Soul

(Based on original code by MELCHIOR FRANZ.)

Pkg soul Emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{soul}

Storage for the colors to use:

```
2 \newcommand*\LWR@soululcolor{}\{}
3
4 \newcommand*\LWR@soulstcolor{}\{}
5
6 % \definecolor{\LWR@soulhlcolordefault}{HTML}{F8E800}
7 % \newcommand*\LWR@soulhlcolor{\LWR@soulhlcolordefault}
8 \newcommand*\LWR@soulhlcolor{}\{}

```

Basic markup with CSS:

```
9 \newcommand{\so}[1]{\InlineClass{letterspacing}{#1}}
10 \newcommand{\caps}[1]{\InlineClass{capsspacing}{#1}}

```

Add colors if not empty:

```
11 \newcommand{\LWR@soulcolor}[4]{%
12 \ifcsempy{#2}%
13 {%
14 \InlineClass{#3}{#1}}%
15 {%
16 \convertcolorspec{named}{\csuse{#2}}{HTML}\LWR@tempcolor%
17 \InlineClass{#3}{#4: \#\LWR@tempcolor}{#1}%
18 }%
19 }
20
21 \newcommand{\ul}[1]{%
22 \LWR@soulcolor{#1}{\LWR@soululcolor}{uline}{text-decoration-color}%
23 }
24
25 \newcommand{\st}[1]{
26 \LWR@soulcolor{#1}{\LWR@soulstcolor}{sout}{text-decoration-color}%
27 }

```

```

28
29 \newcommand{\hl}[1]{
30 \LWR@soulcolor{#1}{\LWR@soulhlcolor}{highlight}{background-color}%
31 }

```

Nullified:

```

32 \newcommand*\soulaccent}[1]{}
33 \newcommand*\soulregister}[2]{}
34 \newcommand*\sloppyword}[1]{#1}
35 \newcommand*\sodef}[5]{\DeclareRobustCommand*#1[1]{\so{##1}}}
36 \newcommand*\resetso{}
37 \newcommand*\capsdef}[5]{}
38 \newcommand*\capsreset{}
39 \newcommand*\capssave}[1]{}
40 \newcommand*\capssselect}[1]{}
41 \newcommand*\setul}[2]{}
42 \newcommand*\resetul{}
43 \newcommand*\setuldepth}[1]{}
44 \newcommand*\setuloverlap}[1]{}

```

Set colors:

```

45 \newcommand*\setulcolor}[1]{\renewcommand{\LWR@soululcolor}{#1}}
46 \newcommand*\setstcolor}[1]{\renewcommand{\LWR@soulstcolor}{#1}}
47 \newcommand*\sethlcolor}[1]{\renewcommand{\LWR@soulhlcolor}{#1}}

```

Long versions of the user-level macros:

```

48 \let\textso\so
49 \let\textul\ul
50 \let\texthl\hl
51 \let\textcaps\caps

```


## Package 75

# lwarp-subfig.sty

## 147 Subfig

(Based on original code by STEVEN DOUGLAS COCHRAN.)

Pkg subfig subfig is supported and patched by lwarp.

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

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

for HTML output: Accept all options for lwarp-subfig:

```
1 \LWR@ProvidesPackagePass{subfig}
```

```
\sf@@@subfloat {<1 type>} [<2 lof entry>] [<3 caption>] {<4 contents>}
```

The outer minipage allows side-by-side subfloats with \hfill between.

```
2 \long\def\sf@@@subfloat#1[#2][#3]#4{%
3 \begin{minipage}{\linewidth}% new
4 \LWR@stoppars% new
5   \ifundefined{FBsc@max}{}%
6     {\FB@readaux{\let\FBsuboheight\relax}}%
7   \@tempcnta=\@ne
8   \if@minipage
9     \@tempcnta=\z@
10  \else\ifdim \lastskip=\z@ \else
11    \@tempcnta=\tw@
12  \fi\fi
13  \ifmaincaptiontop
14    \sf@top=\sf@nearskip
15    \sf@bottom=\sf@farskip
16  \else
17    \sf@top=\sf@farskip
18    \sf@bottom=\sf@nearskip
19  \fi
20  \leavevmode
21  \setbox\@tempboxa \hbox{#4}%
22  \@tempdima=\wd\@tempboxa
```

```

23 \ifundefined{FBsc@max}{}%
24 {\global\advance\Xhsize-\wd\@tempboxa
25 \dimen@=\ht\@tempboxa
26 \advance\dimen@\dp\@tempboxa
27 \ifdim\dimen@>\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\@capttype name}}}%
124 {\@nameuse{thesub\@capttype}}}%
125 {#3}
126 \LWR@htmlblocktag{/figcaption}% new
127 \LWR@traceinfo{sfsubcap B2}% new
128 % }%
129 }%
130     {\dimen@ \ht0%
131     \advance \dimen@ \dp0%
132     \ifdim \dimen@ > \FBsc@max
133     \global \FBsc@max \dimen@
134     \fi
135     \FB@readaux{\let \FBsubcheight \relax}%
136     \ifx \FBsubcheight \relax
137     \def \next{
138 % \parbox[t]{\the \@tempdima}
139 }%
140     \else
141     \def \next{
142 % \parbox[t][\FBsubcheight][t]{\the \@tempdima}
143 }%
144     \fi
145     \vbox{%
146 % \hb@xt@\the \@tempdima{%
147
148 % \hss
149 % \next{%
150 \LWR@traceinfo{sfsubcap C1}% new
151     \caption@make
152     {\@nameuse{sub\@capttype name}}}%
153     {\@nameuse{thesub\@capttype}}}%
154     {#3}
155 \LWR@traceinfo{sfsubcap C1}% new
156 % }%
157 % \hss
158
159 % }
160 }
161 }%
162 \egroup
163 \LWR@startpars% new
164 }

\caption@@@make {\caption label} {\caption text}

165 \renewcommand \caption@@@make [2] {%
166 \LWR@startpars% new

```



```

167 \sbox\@tempboxa{#1}%
168 \ifdim\wd\@tempboxa=\z@
169 \let\caption@lsep\relax
170 \fi
171 \caption@ifempty{#2}{%
172 \let\caption@lsep\@empty
173 \let\caption@tfmt\@firstofone
174 }%
175 % \@setpar{\@par\caption@@par}\caption@@par
176 \renewcommand{\@par}{\LWR@closeparagraph\LWR@orig@@par}% new
177 \caption@applyfont
178 \caption@fmt
179 {\ifcaption@star\else
180 \begingroup
181 \caption@labelfont
182 #1%
183 \endgroup
184 \fi}%
185 {\ifcaption@star\else
186 \begingroup
187 \caption@iflf\caption@labelfont
188 \relax\caption@lsep
189 \endgroup
190 \fi}%
191 {{\caption@textfont
192 \caption@ifstrut
193 {\vrule\@height\ht\strutbox\@width\z@}%
194 }%
195 \nobreak\hskip\z@skip % enable hyphenation
196 \caption@tfmt{#2}
197 \LWR@ensuredoingapar% new
198 \caption@ifstrut
199 {\ifhmode\@finalstrut\strutbox\fi}%
200 }%
201 \par}}
202 \LWR@stoppars% new
203 }

```

Patches for \sf@sub@label:

```

204 \def\subfloat@label{%
205 \LWR@ensuredoingapar% new
206 \@ifnextchar(% ) match left parenthesis
207 {\sf@sub@label}
208 {\sf@sub@label(Sub\@capttype\space
209 \ifundefined{thechapter}{\@nameuse{thechapter}\space}%
210 \@nameuse{p@sub\@capttype}%
211 \@nameuse{thesub\@capttype}.)}}

```

Patches for \subref.

The unstarred version uses a `\ref` link whose printed text comes from the `sub@<label>`:

```
212 \renewcommand{\sf@subref}[1]{%
213 \LWR@subnewref{#1}{sub@#1}%
214 }
```

The starred version uses the printed `sub@<label>` which is stored as if it were a page number:

```
215 \renewcommand{\sf@@subref}[1]{\LWR@origpageref{sub@#1}}
```

Defining new subfloats. The `l@sub<type>` for each is redefined.

```
216 \let\LWR@orig@newsfloat\@newsfloat
217
218 \def\@newsfloat[#1]#2{%
219 \LWR@orig@newsfloat[#1]{#2}%
220 \renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}
221 }
```

Pre-defined for figures and tables:

```
222 \renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2}}
223 \renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}
224 % \def\subfigure{\subfloat}
225 % \def\subtable{\subfloat}
```

## Package 76

# lwarp-tabularx.sty

## 148 Tabularx

Pkg `tabularx` `tabularx` is emulated by `lwarp`.

for **HTML output**: Discard all options for `lwarp-tabularx`:

```
1 \LWR@ProvidesPackageDrop{tabularx}

2 \NewDocumentEnvironment{tabularx}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabularx*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}
```

## Package 77

# lwarp-tabulary.sty

## 149 Tabulary

Pkg tabulary tabulary is emulated by lwarp.

for HTML output: Discard all options for lwarp-tabulary.

Column types L, C, R, and J are emulated by lwarp core code.

```
1 \LWR@ProvidesPackageDrop{tabulary}

2 \NewDocumentEnvironment{tabulary}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabulary*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}
9
10 \newdimen\tymin
11 \newdimen\tymax
12 \def\tyformat{}
```

## Package 78

# lwarp-textpos.sty

## 150 Textpos

Pkg `textpos` `textpos` is emulated during HTML output, and the `textpos` package is ignored.

```
for HTML output: 1 \LWR@ProvidesPackageDrop{textpos}

2 \NewDocumentEnvironment{textblock}{m r()}{\}{}
3 \NewDocumentEnvironment{textblock*}{m o r()}{\}{}
4 \newcommand*{\TPGrid}[3][\]{}
5 \NewDocumentCommand{\TPMargin}{s o}{}
6 \newcommand*{\textblockcolour}[1]{}
7 \newcommand*{\textblockrulecolour}[1]{}
8 \newcommand*{\textblockcolor}[1]{}
9 \newcommand*{\textblockrulecolor}[1]{}
10 \newcommand*{\tekstblokkulur}[1]{}
11 \newcommand*{\tekstblokrulekulur}[1]{}
12 \newlength{\TPHorizModule}
13 \newlength{\TPVertModule}
14 \newlength{\TPboxrulesize}
15 \newcommand{\textblocklabel}[1]{}
16 \newcommand*{\showtextsize}{}
17 \newcommand{\textblockorigin}[2]{}

```

## Package 79

# lwarp-theorem.sty

## 151 Theorem

(Based on original code by FRANK MITTELBAACH.)

Pkg theorem theorem is patched for use by lwarp.

### CSS styling of theorems and proofs:

**Theorem:** <div> of class theorembody<theoremstyle>

**Theorem Header:** <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 \theoremstyle
27      \expandafter \endcsname \the \theorem@bodyfont
28      \@thm{#1}{#2}}%
29  \global \expandafter \let \csname end#1\endcsname \@endtheorem
30  \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
31  }}
32
33 \gdef\@ynthm#1#2{%
34 \expandafter\@ifdefinable\csname #1\endcsname
35   {
36 \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
37 \@definecounter{#1}%
38   \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
39   \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
40   \expandafter{\csname th@the \theoremstyle \expandafter
41     \endcsname \the\theorem@bodyfont \@thm{#1}{#2}}%
42   \global \expandafter \let \csname end#1\endcsname \@endtheorem
43   \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
44   }}
45
46 \gdef\@othm#1[#2]#3{%
47 \expandafter\ifx\csname c@#2\endcsname\relax
48   \@nocounterr{#2}%
49   \else
50   \expandafter\@ifdefinable\csname #1\endcsname
51   {
52 \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
53 \expandafter \xdef \csname the#1\endcsname
54   {\expandafter \noexpand \csname the#2\endcsname}%
55   \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
56   \expandafter{\csname th@the \theoremstyle \expandafter
57     \endcsname \the\theorem@bodyfont \@thm{#2}{#3}}%
58   \global \expandafter \let \csname end#1\endcsname \@endtheorem
59   \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
60 }%
61 \fi}

```

The following are patched for CSS.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc.  
They are gathered together here.

Each theorem is encased in a BlockClass environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader`.

```

62 \gdef\th@plain{%\normalfont\itshape
63   \def\@begintheorem##1##2{%
64     \LWR@forcenewpage% new
65     \BlockClass{theorembody\LWR@thisthmstyle}% new
66       \item[\hskip\labelsep
67 \InlineClass{theoremheader}{##1\ ##2}
68 ]}%
69 \def\@opargbegintheorem##1##2##3{%
70   \LWR@forcenewpage% new
71   \BlockClass{theorembody\LWR@thisthmstyle}% new
72     \item[\hskip\labelsep
73 \InlineClass{theoremheader}{##1\ ##2\ (##3)}
74 ]}
75 }
76
77 \gdef\th@break{%\normalfont\slshape
78   \def\@begintheorem##1##2{%
79     \LWR@forcenewpage% new
80     \BlockClass{theorembody\LWR@thisthmstyle}% new
81     \item[\hskip \labelsep
82 \InlineClass{theoremheader}{##1\ ##2}\newline%
83 ]}%
84 \def\@opargbegintheorem##1##2##3{%
85   \LWR@forcenewpage% new
86   \BlockClass{theorembody\LWR@thisthmstyle}% new
87     \item[\hskip \labelsep
88 \InlineClass{theoremheader}{##1\ ##2\ (##3)}\newline
89 ]}
90 }
91
92 \gdef\th@marginbreak{%\normalfont\slshape
93   \def\@begintheorem##1##2{
94     \LWR@forcenewpage% new
95     \BlockClass{theorembody\LWR@thisthmstyle}% new
96     \item[\hskip\labelsep %
97 \InlineClass{theoremheader}{##2 \quad ##1}\newline
98 ]}%
99 \def\@opargbegintheorem##1##2##3{%
100   \LWR@forcenewpage% new
101   \BlockClass{theorembody\LWR@thisthmstyle}% new
102   \item[\hskip\labelsep %
103 \InlineClass{theoremheader}{##2 \quad ##1\ %
104 (##3)}\newline
105 ]}
106 }
107
```



```

108 \gdef\th@changebreak{%\normalfont\slshape
109 \def\@begintheorem##1##2{
110 \LWR@forcenewpage% new
111 \BlockClass{theorembody\LWR@thisthmstyle}% new
112 \item[\hskip\labelsep
113 \InlineClass{theoremheader}{##2\ ##1}\newline
114 ]}%
115 \def\@opargbegintheorem##1##2##3{%
116 \LWR@forcenewpage% new
117 \BlockClass{theorembody\LWR@thisthmstyle}% new
118 \item[\hskip\labelsep
119 \InlineClass{theoremheader}{ ##2\ ##1\ %
120 (##3)}\newline
121 ]}
122 }
123
124 \gdef\th@change{%\normalfont\slshape
125 \def\@begintheorem##1##2{
126 \LWR@forcenewpage% new
127 \BlockClass{theorembody\LWR@thisthmstyle}% new
128 \item[\hskip\labelsep
129 \InlineClass{theoremheader}{##2\ ##1}
130 ]}%
131 \def\@opargbegintheorem##1##2##3{%
132 \LWR@forcenewpage% new
133 \BlockClass{theorembody\LWR@thisthmstyle}% new
134 \item[\hskip\labelsep
135 \InlineClass{theoremheader}{##2\ ##1\ (##3)}
136 ]}
137 }
138
139 \gdef\th@margin{%\normalfont\slshape
140 \def\@begintheorem##1##2{
141 \LWR@forcenewpage% new
142 \BlockClass{theorembody\LWR@thisthmstyle}% new
143 \item[\hskip\labelsep
144 \InlineClass{theoremheader}{##2 \quad ##1}
145 ]}%
146 \def\@opargbegintheorem##1##2##3{%
147 \LWR@forcenewpage% new
148 \BlockClass{theorembody\LWR@thisthmstyle}% new
149 \item[\hskip\labelsep
150 \InlineClass{theoremheader}{##2 \quad ##1\ (##3)}
151 ]}
152 }

```

Patched for CSS:

```

153 \gdef\@endtheorem{\endBlockClass\endtrivlist}

```

## Package 80

# lwarp-threeparttable.sty

## 152 Threeparttable

Pkg `threeparttable` `threeparttable` is emulated during HTML output, and the `threeparttable` package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{threeparttable}`

Prints the table note item header inside a CSS class of `tnoteitemheader`.

```
2 \newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}
```

To emulate `threeparttable`:

```
3 \newenvironment*{threeparttable}[1][b]{}{}
```

```
4 \newenvironment*{tablenotes}[1]{}
5 {%
6 \LWR@forcenewpage
7 \BlockClass{tnotes}%
8 \setlist[description]{format=\LWR@printtablenote}%
9 \description%
10 }
11 {%
12 \enddescription%
13 \endBlockClass%
14 }
```

```
15 \newcommand{\tnote}[1]{\textsuperscript{#1}}
```

## Package 81

# lwarp-tikz.sty

## 153 Tikz

Pkg **tikz** tikz is supported.

Accept all options for lwarp-tikz:

```
1 \LWR@ProvidesPackagePass{tikz}
```

**catcodes** lwarp changes the catcode of `$` for its own use. The Tikz `babel` library temporarily changes catcodes back to normal for Tikz's use. tikz v3.0.0 introduced the `babel` library which handles catcode changes. For older versions, lwarp must change `$`'s catcode itself.

for HTML output: 2 \begin{warpHTML}

```
3 \newboolean{LWR@tikzbabel}
```

```
4
```

```
5 \@ifpackagelater{tikz}{2013/12/20}% Test for Tikz version v3.0.0
```

```
6 {\usetikzlibrary{babel}\booltrue{LWR@tikzbabel}}
```

```
7 {\boolfalse{LWR@tikzbabel}}
```

Env **tikzpicture** tikzpicture environment is enclosed inside a `\lateximage`. May be used as-is, and its contents will be converted to an image.

```
8 \BeforeBeginEnvironment{tikzpicture}{%
```

```
9 \lateximage%
```

```
10 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
```

```
11 {}%
```

```
12 {\catcode'\$=3} % dollar sign is math shift
```

```
13 }
```

```
14
```

```
15 \AfterEndEnvironment{tikzpicture}{%
```

```
16 \endlateximage%
```

```
17 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
```

```
18 {}%
```

```
19 {\catcode'\$=\active}%
```

```
20 }
```

```
21 \end{warpHTML}
```

## Package 82

# lwarp-titles.sty

## 154 Titles

Pkg titles titles is loaded and used by lwarp during HTML output. All user options and macros are ignored and disabled.

Discard all options for lwarp-titles:

for HTML output: 1 \LWR@ProvidesPackageDrop{titles}

\pagestyle and \thispagestyle are already disabled in the lwarp code.

```

2 \RenewDocumentCommand{\newpagestyle}{m o m}{}
3 \RenewDocumentCommand{\renewpagestyle}{m o m}{}

4 \RenewDocumentCommand{\sethead}{o o o m m m}{}
5 \RenewDocumentCommand{\setfoot}{o o o m m m}{}

6 \RenewDocumentCommand{\settitledmarks}{s m}{}

7 \renewcommand*{\headrule}{}
8 \renewcommand*{\footrule}{}

9 \renewcommand*{\setheadrule}[1]{}
10 \renewcommand*{\setfootrule}[1]{}

11 \newcommand*{\makeheadrule}{}
12 \newcommand*{\makefootrule}{}

13 \renewcommand{\setmarkboth}[1]{}

14 \RenewDocumentCommand{\widenhead}{s o o m m}{}

15 \renewcommand*{\bottitledmarks}{}
16 \renewcommand*{\toptitledmarks}{}
17 \renewcommand*{\firsttitledmarks}{}
18 \renewcommand*{\nexttoptitledmarks}{}
19 \renewcommand*{\outertitledmarks}{}
20 \renewcommand*{\innertitledmarks}{}

21 \RenewDocumentCommand{\newtitledmark}{s m}{}

```

```
22 \RenewDocumentCommand{\pretitlemark}{s m m}{}{}
```

```
23 \renewcommand{\ifsamemark}[4]{}{}
```

```
24 \NewDocumentCommand{\setfloathead}{s o o o m m m m m}{}{}
```

```
25 \NewDocumentCommand{\setfloatfoot}{s o o o m m m m m}{}{}
```

```
26 \NewDocumentCommand{\nextfloathead}{s o o o m m m m m}{}{}
```

```
27 \NewDocumentCommand{\nextfloatfoot}{s o o o m m m m m}{}{}
```

```
28 \newcommand{\newmarkset}[1]{}{}
```

```
29 \NewDocumentCommand{\newextramarkset}{s m m}{}{}
```

```
30 \newcommand{\botextramarks}[1]{}{}
```

```
31 \newcommand{\topextramarks}[1]{}{}
```

```
32 \newcommand{\firstextramarks}[1]{}{}
```

```
33 \newcommand{\nexttoextramarks}[1]{}{}
```

```
34 \newcommand{\outerextramarks}[1]{}{}
```

```
35 \newcommand{\innerextramarks}[1]{}{}
```

## Package 83

# lwarp-titlesec.sty

## 155 Titlesec

Pkg titlesec titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titlesec:

```
for HTML output: 1 \LWR@ProvidesPackageDrop{titlesec}

2 \newcommand*{\titlelabel}[1]{}

3 \newcommand\titleformat{%
4   \@ifstar{\ttl@format@s}%
5     {\ttl@format@i}}
6 \newcommand{\ttl@format@s}[1]{}
7 \NewDocumentCommand{\ttl@format@i}{m o m m m o}{}

8 \@ifundefined{chapapp}{\let\@chapapp\chaptername}{}
9 \newcommand\chaptertitlename{\@chapapp}

10 \NewDocumentCommand{\titlespacing}{s m m m m o}{}

11 \newcommand*{\filright}{}
12 \newcommand*{\filcenter}{}
13 \newcommand*{\filleft}{}
14 \newcommand*{\fillast}{}
15 \newcommand*{\filinner}{}
16 \newcommand*{\filouter}{}

17 \newcommand\wordsep{\fontdimen\tw@\font \@plus
18   \fontdimen\thr@\font \@minus \fontdimen4\font}

19 \NewDocumentCommand{\titleline}{s o m}{}

20 \providecommand*{\titlerule}{\@ifstar{\ttl@row}{\ttl@rule}}
21 \newcommand*{\ttl@rule}[1] [] {}
22 \newcommand*{\ttl@row}[2] [] {}

23 \newcommand{\iftitlemeasuring}[2] {#2}

24 \newcommand{\assignpagestyle}[2] {#2}

25 \NewDocumentCommand{\titleclass}{m o m o}
```

## Package 84

# lwarp-titletoc.sty

## 156 Titletoc

Pkg titletoc titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titletoc:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{titletoc}

2 \NewDocumentCommand{\dottedcontents}{m o m m m}{\}

3 \newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
4 \NewDocumentCommand{\ttl@tcstar}{m o m m m o o}{\}
5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m m o}{\}

6 \newcommand{\contentsmargin}[2] [] {}

7 \newcommand*{\thecontentslabel}{\thecontentslabel}
8 \newcommand*{\thecontentspage}{\thecontentspage}

9 \newcommand{\contentslabel}[2] [] {\thecontentslabel}
10 \newcommand{\contentspage}[1] [] {\thecontentspage}

11 \newcommand{\contentspush}[1] {}

12 \newcommand{\contentsuse}[2] {}

13 \newcommand*{\startcontents}[1] [] {}
14 \newcommand*{\stopcontents}[1] [] {}
15 \newcommand*{\resumecontents}[1] [] {}

16 \newcommand{\printcontents}[4] [] {}

17 \newcommand{\startlist}[2] [] {}
18 \newcommand{\stoplist}[2] [] {}
19 \newcommand{\resumelist}[2] [] {}

20 \newcommand{\printlist}[4] [] {}

```

## Package 85

# lwarp-titling.sty

## 157 Titling

`Pkg titling` titling is used by lwarp. The following patches are not needed by lwarp, but are required if the user requests titling.

lwarp uses page notes for footnotes, so the various titling footnote restyling commands have no effect.

Pass all options to lwarp-titling:

**for HTML output:**    `1 \LWR@ProvidesPackagePass{titling}`

Patch `\@bsmtitleempty`:

```
2 \let\LWR@orig@bsmtitleempty\@bsmtitleempty
3 \renewcommand*{\@bsmtitleempty}{%
4 \LWR@orig@bsmtitleempty%
5 \global\let\published\relax%
6 \global\let\subtitle\relax%
7 }
```

Patch `\keepthetitle`:

```
8 \let\LWR@origkeepthetitle\keepthetitle
9 \renewcommand*{\keepthetitle}{%
10 \LWR@orig@keepthetitle%
11 \global\let\@published\@empty%
12 \global\let\@subtitle\@empty%
13 }
```

Patch `\killtitle`:

```
14 \let\LWR@origkilltitle\killtitle
15 \renewcommand*{\killtitle}{%
16 \LWR@orig@killtitle%
17 \global\let\thepublished\relax%
18 \global\let\thesubtitle\relax%
19 }
```



## Package 86

# lwarp-tocloft.sty

## 158 Tocloft

Pkg `tocloft` `tocloft` is emulated. Most user options and macros are ignored and disabled. `\newlistof` and `\cftchapterprecis` are supported.

Discard all options for `lwarp-tocloft`:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{tocloft}

2 \newcommand{\tocloftpagestyle}[1]{}

3 \newcommand*\cftmarktoc{}
4 \newcommand*\cfttoctitlefont{}
5 \newcommand*\cftaftertoctitle{}

6 \newlength{\cftbeforetoctitleskip}
7 \newlength{\cftaftertoctitleskip}

8 \newcommand*\cftmarklof{}
9 \newcommand*\cftloftitlefont{}
10 \newcommand*\cftafterloftitle{}

11 \newlength{\cftbeforeloftitleskip}
12 \newlength{\cftafterloftitleskip}

13 \newcommand*\cftmarklot{}
14 \newcommand*\cftlottitlefont{}
15 \newcommand*\cftafterlottitle{}

16 \newlength{\cftbeforelottitleskip}
17 \newlength{\cftafterlottitleskip}

18 \newcommand*\cftdot}{.}
19 \providecommand*\cftdotsep}{1}
20 \newcommand*\cftnodots}{5000}
21
22 \providecommand{\cftdotfill}[1]{}

23 \newcommand*\cftsetpnumwidth}[1]{}
24 \newcommand*\cftsetrmarg}[1]{}

```

```
25 \newcommand*{\cftpnumalign}[1]{}

26 \newlength{\cftparskip}

27 \newlength{\cftbeforepartskip}
28 \newlength{\cftpartincent}
29 \newlength{\cftpnumwidth}
30 \newcommand*{\cftpfont}{}
31 \newcommand*{\cftpapresnum}{}
32 \newcommand*{\cftpataftersnum}{}
33 \newcommand*{\cftpataftersnumb}{}
34 \newcommand*{\cftpleader}{}
35 \newcommand*{\cftpdotsep}{1}
36 \newcommand*{\cftpfont}{}
37 \newcommand*{\cftpartafterpnum}{}

38 \newlength{\cftbeforechapskip}
39 \newlength{\cftchapindent}
40 \newlength{\cftchapnumwidth}
41 \newcommand*{\cftchapfont}{}
42 \newcommand*{\cftchappresnum}{}
43 \newcommand*{\cftchapaftersnum}{}
44 \newcommand*{\cftchapaftersnumb}{}
45 \newcommand*{\cftchapleader}{}
46 \newcommand*{\cftchapdotsep}{1}
47 \newcommand*{\cftchapfont}{}
48 \newcommand*{\cftchapafterpnum}{}

49 \newlength{\cftbeforesecskip}
50 \newlength{\cftsecindent}
51 \newlength{\cftsecnumwidth}
52 \newcommand*{\cftsecfont}{}
53 \newcommand*{\cftsecpresnum}{}
54 \newcommand*{\cftsecaftersnum}{}
55 \newcommand*{\cftsecaftersnumb}{}
56 \newcommand*{\cftsecleader}{}
57 \newcommand*{\cftsecdotsep}{1}
58 \newcommand*{\cftsecfont}{}
59 \newcommand*{\cftsecafterpnum}{}

60 \newlength{\cftbeforesubsecskip}
61 \newlength{\cftsubsecindent}
62 \newlength{\cftsubsecnumwidth}
63 \newcommand*{\cftsubsecfont}{}
64 \newcommand*{\cftsubsecpresnum}{}
65 \newcommand*{\cftsubsecaftersnum}{}
66 \newcommand*{\cftsubsecaftersnumb}{}
67 \newcommand*{\cftsubsecleader}{}
68 \newcommand*{\cftsubsecdotsep}{1}
```

```
69 \newcommand*{\cftsubsecpagefont}{}
70 \newcommand*{\cftsubsecafterpnum}{}

71 \newlength{\cftbeforesubsubsecskip}
72 \newlength{\cftsubsubsecindent}
73 \newlength{\cftsubsubsecnumwidth}
74 \newcommand*{\cftsubsubsecfont}{}
75 \newcommand*{\cftsubsubsecpresnum}{}
76 \newcommand*{\cftsubsubsecaftersnum}{}
77 \newcommand*{\cftsubsubsecaftersnumb}{}
78 \newcommand*{\cftsubsubsecleader}{}
79 \newcommand*{\cftsubsubsecdotsep}{1}
80 \newcommand*{\cftsubsubsecpagefont}{}
81 \newcommand*{\cftsubsubsecafterpnum}{}

82 \newlength{\cftbeforeparaskip}
83 \newlength{\cftparaindent}
84 \newlength{\cftparanumwidth}
85 \newcommand*{\cftparafont}{}
86 \newcommand*{\cftparapresnum}{}
87 \newcommand*{\cftparaftersnum}{}
88 \newcommand*{\cftparaftersnumb}{}
89 \newcommand*{\cftparaleader}{}
90 \newcommand*{\cftparadotsep}{1}
91 \newcommand*{\cftparapagefont}{}
92 \newcommand*{\cftparaafterpnum}{}

93 \newlength{\cftbeforesubparaskip}
94 \newlength{\cftsubparaindent}
95 \newlength{\cftsubparanumwidth}
96 \newcommand*{\cftsubparafont}{}
97 \newcommand*{\cftsubparapresnum}{}
98 \newcommand*{\cftsubparaftersnum}{}
99 \newcommand*{\cftsubparaftersnumb}{}
100 \newcommand*{\cftsubparaleader}{}
101 \newcommand*{\cftsubparadotsep}{1}
102 \newcommand*{\cftsubparapagefont}{}
103 \newcommand*{\cftsubparaafterpnum}{}

104 \newlength{\cftbeforefigskip}
105 \newlength{\cftfigindent}
106 \newlength{\cftfignumwidth}
107 \newcommand*{\cftfigfont}{}
108 \newcommand*{\cftfigpresnum}{}
109 \newcommand*{\cftfigaftersnum}{}
110 \newcommand*{\cftfigaftersnumb}{}
111 \newcommand*{\cftfigleader}{}
112 \newcommand*{\cftfigdotsep}{1}
113 \newcommand*{\cftfigpagefont}{}

```

```

114 \newcommand*{\cftfigafterpnum}{}

115 \newlength{\cftbeforesubfigskip}
116 \newlength{\cftsubfigindent}
117 \newlength{\cftsubfignumwidth}
118 \newcommand*{\cftsubfigfont}{}
119 \newcommand*{\cftsubfigpresnum}{}
120 \newcommand*{\cftsubfigaftersnum}{}
121 \newcommand*{\cftsubfigaftersnumb}{}
122 \newcommand*{\cftsubfigleader}{}
123 \newcommand*{\cftsubfigdotsep}{1}
124 \newcommand*{\cftsubfigpagefont}{}
125 \newcommand*{\cftsubfigafterpnum}{}

126 \newlength{\cftbeforetabskip}
127 \newlength{\cfttabindent}
128 \newlength{\cfttabnumwidth}
129 \newcommand*{\cfttabfont}{}
130 \newcommand*{\cfttabpresnum}{}
131 \newcommand*{\cfttabaftersnum}{}
132 \newcommand*{\cfttabaftersnumb}{}
133 \newcommand*{\cfttableader}{}
134 \newcommand*{\cfttabdotsep}{1}
135 \newcommand*{\cfttabpagefont}{}
136 \newcommand*{\cfttabafterpnum}{}

137 \newlength{\cftbeforesubtabskip}
138 \newlength{\cftsubtabindent}
139 \newlength{\cftsubtabnumwidth}
140 \newcommand*{\cftsubtabfont}{}
141 \newcommand*{\cftsubtabpresnum}{}
142 \newcommand*{\cftsubtabaftersnum}{}
143 \newcommand*{\cftsubtabaftersnumb}{}
144 \newcommand*{\cftsubtableader}{}
145 \newcommand*{\cftsubtabdotsep}{1}
146 \newcommand*{\cftsubtabpagefont}{}
147 \newcommand*{\cftsubtabafterpnum}{}

148 \newcommand{\cftsetindents}[3]{}

149 \newcommand{\pagenumbersoff}[1]{}
150 \newcommand{\pagenumberon}[1]{}

```

Emulated through the \newfloat mechanism.

```

151 \NewDocumentCommand{\newlistof}{o m m}
152 {%
153 \IfValueTF{#1}

```

```

154 {\newfloat{#2}{tbp}{#3}[#1]}
155 {\newfloat{#2}{tbp}{#3}}
156 \@namedef{listof#2}{\listof{#2}{#4}}
157 \@namedef{#2depth}{1}
158 \expandafter\newlength\csuse{cftbefore#2skip}
159 \expandafter\newlength\csuse{cft#2indent}
160 \expandafter\newlength\csuse{cft#2numwidth}
161 \@namedef{cft#2font}{}
162 \@namedef{cft#2presnum}{}
163 \@namedef{cft#2aftersnum}{}
164 \@namedef{cft#2aftersnumb}{}
165 \@namedef{cft#2leader}{}
166 \@namedef{cft#2dotsep}{1}
167 \@namedef{cft#2pagefont}{}
168 \@namedef{cft#2afterpnum}{}
169 }

```

\cftchapterprecis from tocloft:

```

170 \newcommand{\cftchapterprecis}[1]{%
171   \cftchapterprecishere{#1}
172   \cftchapterprecistoc{#1}}
173 \newcommand{\cftchapterprecishere}[1]{%
174   \begin{quote}\textit{#1}\end{quote}}
175 \newcommand{\cftchapterprecistoc}[1]{
176   \addtocontents{toc}{%
177     {
178       \protect\begin{quote}#1\protect\end{quote}}
179   }
180 }

```

## Package 87

# lwarp-trivfloat.sty

## 159 Trivfloat

**Pkg** `trivfloat` `trivfloat` is forced to use the built-in `lwarp` emulation for floats.

Discard all options for `lwarp-trivfloat`. This tells `trivfloat` not to use `floatrow` or `memoir`.

**for HTML output:** `1 \LWR@ProvidesPackageDrop{trivfloat}`  
`2 \LWR@origRequirePackage{trivfloat}`

**for HTML & PRINT:** `3 \begin{warpall}`

To create a new float type and change its name:

---

```
\trivfloat{example}
\renewcommand{\exemplename}{Example Name}
\crefname{example}{example}{examples}
\Crefname{example}{Example}{Examples}
```

---

`4 \end{warpall}`

**\tfl@chapter@fix** Nullified at the beginning of the document. Is used by `trivfloat` to correct float chapter numbers, but is not needed for `lwarp`.

**for HTML output:** `5 \begin{warpHTML}`  
`6 \AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}`  
`7 \end{warpHTML}`

### 159.1 Combining `\newfloat`, `\trivfloat`, and `algorithmicx`

**for HTML & PRINT:** `8 \begin{warpall}`

For both print and HTML output:



When using `float`, `trivfloat`, or `algorithmicx` at the same time, be aware of conflicting file usage. `algorithmicx` uses `.loa`. `trivfloat` by default starts with `.loa` and goes up

for additional floats, skipping .lof and .lot.



When using `\newfloat`, be sure to manually assign higher letters to the `\newfloat` files to avoid .loa used by `algorithmicx`, and any files used by `trivfloat`. Also avoid using .lof and .lot.



When using `\trivfloat`, you may force it to avoid conflicting with `algorithmicx` by starting `trivfloat`'s file extensions with .lob:

---

```
\makeatletter
\setcounter{tfl@float@cnt}{1}} % start trivfloats with .lob
\makeatletter
```

---

```
9 \end{warpall}
```

## Package 88

# lwarp-ulem.sty

## 160 Ulem

*(Based on original code by DONALD ARSENEAU.)*

Pkg ulem Emulated.

for HTML output: Original lwarp definitions:

```
1 \let\LWR@ulemorigemph\emph
2 \let\LWR@ulemorigtextbf\textbf
```

Basic markup commands, using CSS:

```
3 \NewDocumentCommand{\uline}{+m}{%
4 \InlineClass{uline}{#1}%
5 }
6
7 \NewDocumentCommand{\uuline}{+m}{%
8 \InlineClass{uuline}{#1}%
9 }
10
11 \NewDocumentCommand{\uwave}{+m}{%
12 \InlineClass{uwave}{#1}%
13 }
14
15 \NewDocumentCommand{\sout}{+m}{%
16 \InlineClass{sout}{#1}%
17 }
18
19 \NewDocumentCommand{\xout}{+m}{%
20 \InlineClass{xout}{#1}%
21 }
22
23 \NewDocumentCommand{\dashuline}{+m}{%
24 \InlineClass{dashuline}{#1}%
25 }
26
27 \NewDocumentCommand{\dotuline}{+m}{%
28 \InlineClass{dotuline}{#1}%
29 }
```



Nullified parameters:

```
30 \NewDocumentCommand{\ULthickness}{-}{-}
31 \newlength{\ULdepth}
```

Nullified/emulated macros:

```
32 \NewDocumentCommand{\markoverwith}{m}{-}
33 \NewDocumentCommand{\ULon}{+m}{\uline{#1}\egroup}
```

`\useunder` only works with `\textbf`, etc, but not `\bfseries`, etc.

```
34 \NewDocumentCommand{\useunder}{m m m}{%
35 \relax%
36 \ifx\relax#3\relax\else % argumentative command
37   \def#3{#1}\MakeRobust{#3}\fi
38 }
```

Triggered by package options, also available for the users:

```
39 \newcommand*{\normalem}{\let\emph\LWR@ulemorigemph}
40 \newcommand*{\ULforem}{\let\emph\uline}
41 \ULforem% default
```

Package options:

```
42 \DeclareOption{normalem}{\normalem}
43 \DeclareOption{ULforem}{\ULforem}
44 \DeclareOption{normalbf}{-}
45 \DeclareOption{UWforbf}{\useunder{\uwave}{\bf}\textbf}}
```

Emulate the original package:

```
46 \LWR@ProvidesPackageDrop{ulem}
```

## Package 89

# lwarp-verse.sty

## 161 Verse

(Based on original code by PETER WILSON.)

**Pkg verse** verse is supported and patched by lwarp.

**for HTML output:** Pass all options for lwarp-verse:

```
1 \LWR@ProvidesPackagePass{verse}
```

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

---

```
\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}
```

---

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

**Len \leftmargini**

**Len \TMLvleftskip**

**Len \TMLleftmargini**

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

**Env verse** The `verse` environment will be placed inside a HTML `pre`.

```
2 \AfterEndPreamble{
```

At the beginning of the `verse` environment:

```

3 \AtBeginEnvironment{verse}
4 {%

Pkg  verse  The verse or memoir packages can place stanza numbers to the left with their
Pkg  memoir \flagverse command. Do not allow them to go into the left margin, which would
\flagverse cause pdfcrop to crop the entire page further to the left:

Len  \leftskip
5 \ifdef{\vleftskip}{%
6 \setlength{\vleftskip}{\HTMLvleftskip}
7 \setlength{\leftmargini}{\HTMLleftmargini}
8 }{}
9 \LWR@forcenewpage
10 \LWR@atbeginverbatim{verse}
11 \unskip\vspace{-\baselineskip}
12 }
```

After the end of the `verse` environment, which places the `pre` tag at the regular left margin:

```

13 \AfterEndEnvironment{verse}{
14 \unskip\vspace{-\baselineskip}
15 \LWR@afterendverbatim
16 }
```

Patch to place poemtitle inside an HTML span of class `poemtitle`:

```

17 \ifdef{\poemtitle}{
18 \DeclareDocumentCommand{\@vstypeptitle}{m}{%
19   \vspace{\beforepoemtitleskip}%
20   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
21   \vspace{\afterpoemtitleskip}%
22 }
23 }{}
24
25 }
```

## Package 90

# lwarp-wallpaper.sty

## 162 Wallpaper

Pkg wallpaper wallpaper is emulated during HTML output, and the wallpaper package is ignored.

```
for HTML output: 1 \LWR@ProvidesPackageDrop{wallpaper}

2 \newcommand*{\CenterWallPaper}[2]{}
3 \newcommand*{\ThisCenterWallPaper}[2]{}
4 \newcommand*{\TileWallPaper}[3]{}
5 \newcommand*{\ThisTileWallPaper}[3]{}
6 \newcommand*{\TileSquareWallPaper}[2]{}
7 \newcommand*{\ThisTileSquareWallPaper}[2]{}
8 \newcommand*{\ULCornerWallPaper}[2]{}
9 \newcommand*{\ThisULCornerWallPaper}[2]{}
10 \newcommand*{\LLCornerWallPaper}[2]{}
11 \newcommand*{\ThisLLCornerWallPaper}[2]{}
12 \newcommand*{\URCornerWallPaper}[2]{}
13 \newcommand*{\ThisURCornerWallPaper}[2]{}
14 \newcommand*{\LRCornerWallPaper}[2]{}
15 \newcommand*{\ThisLRCornerWallPaper}[2]{}
16 \newcommand*{\ClearWallPaper}{}
17 \newlength{\wpXoffset}
18 \newlength{\wpYoffset}
```

## Package 91

# lwarp-wrapfig.sty

## 163 Wrapfig

Pkg wrapfig wrapfig is emulated during HTML output, and the wrapfig package is ignored.

```
for HTML output: 1 \LWR@ProvidesPackageDrop{wrapfig}

Computed width of a wrapped object. Used to print the HTML style.

2 \newlength{\LWR@wrapwidth}

3
4 \newcommand*{\LWR@wrapposition}{}
5
6 \newcommand*{\LWR@subwrapfigure}[2]{%
7 \LWR@maybeinthisfloat%
8 \renewcommand*{\LWR@wrapposition}{}%
9 \ifthenelse{%
10 \equal{#1}{r}\OR\equal{#1}{R}\OR%
11 \equal{#1}{o}\OR\equal{#1}{O}%
12 }{%
13 {\renewcommand*{\LWR@wrapposition}{float:right}}%
14 {\renewcommand*{\LWR@wrapposition}{float:left}}%
15 \setlength{\LWR@wrapwidth}{#2}%
16 \addtolength{\LWR@wrapwidth}{4em}%
17 \uselengthunit{PT}%
18 \LWR@forcenewpage
19 \LWR@stoppars%
20 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{\LWR@thisfloat}"
21 style="width:\rndprintlength{\LWR@wrapwidth} ; %
22 \LWR@wrapposition"%
23 }
24 \LWR@startpars
25 }
26
27
28 \NewDocumentEnvironment{wrapfigure}{o m o m}
29 {%
30 \LWR@subwrapfigure{#2}{#4}%
31 \captionsetup{type=figure}%
32 }
33 {
34 \LWR@htmldivclassend{div}
```

```
35 }
36
37
38 \NewDocumentEnvironment{wraptable}{o m o m}
39 {%
40 \LWR@subwrapfigure{#2}{#4}%
41 \captionsetup{type=table}%
42 }
43 {
44 \LWR@htmldivclassend{div}
45 }
46
47
48 \NewDocumentEnvironment{wrapfloat}{m o m o m}
49 {%
50 \LWR@subwrapfigure{#3}{#5}%
51 \captionsetup{type=#1}%
52 }
53 {
54 \LWR@htmldivclassend{div}
55 }
56
57 \newlength{\wrapoverhang}
```

## Package 92

# lwarp-xcolor.sty

## 164 Xcolor

Pkg `xcolor` `xcolor` is supported by `lwarp`.

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

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

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

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

for HTML output: `1 \LWR@ProvidesPackagePass{xcolor}`

`2 \newcommand*{\LWR@tempcolor}{}%`

defaulting to black.

`3 \newcommand*{\LWR@currenttextcolor}{black}`

`\LWR@colorstyle {<1: styletext>} {<2: model>} {<3: color>} {<4: spancontents>}`

Creates a styled span with a color converted to HTML hex colorspace. Uses `LWR@spandepth` to prevent paragraph tags inside the span. If used for `\textcolor`, with a styletext of `color:`, then the new color is copied into `\LWR@currenttextcolor` for possible re-use in `\rule`.

`4 \NewDocumentCommand{\LWR@colorstyle}{m m m m}{%`

Use the `xcolor` package to convert to an HTML color space:

`5 \convertcolorspec{#2}{#3}{HTML}\LWR@tempcolor%`

If is a `\textcolor`, save a copy of this color for use by `\rule`:

`6 \ifthenelse{\equal{#1}{color:}}{%`

`7 {\renewcommand*{\LWR@currenttextcolor}{\#\LWR@tempcolor}}{}}%`

Create the HTML `<span>` with the styled color:

```

8 \LWR@htmltagc{span style="#1\#\LWR@tempcolor"{} }%
9 \begin{LWR@nestspan}%

```

Prevent additional paragraph tags inside this span:

Print the contents then close the span:

```

10 #4%
11 \LWR@htmltagc{/span}%
12 \end{LWR@nestspan}%

```

For paragraph-tag handling:

```

13 \LWR@ensuredoingapar%
14 }

```

`\color` appears in the L<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*} is converted into an HTML hex color span.

```

15 \NewDocumentCommand{\LWR@textcolor}{0{named} m m}{%
16 \begingroup%
17 \LWR@colorstyle{color:}{#1}{#2}{#3}%
18 \endgroup%
19 }

```

`\pagecolor` [*model*] {*color*} is ignored. Use `\CSSFilename` instead.

```

20 \newcommand*{\LWR@pagecolor}[2][named]{ }

```

`\colorbox` [*model*] {*color*} {*text*} is converted into an HTML hex background color span.

```

21 \NewDocumentCommand{\LWR@colorbox}{0{named} m m}{%
22 \begingroup%
23 \LWR@colorstyle{background:}{#1}{#2}{#3}%
24 \endgroup%
25 }

```

`\fcolorbox` [*framemodel*] {*framecolor*} [*boxmodel*] {*boxcolor*} {*text*} is converted into a framed HTML hex background color span.



A background color of "none" creates a colored frame without a background color.

```
26 \NewDocumentCommand{\LWR@fcolorbox}{O{named} m O{named} m m}{%
27 \begingroup%
28 \ifthenelse{\equal{#4}{none}}{% no background color
29 \LWR@colorstyle{border:1px solid }{#1}{#2}{#5}%
30 }{% yes background color
31 \LWR@colorstyle{border:1px solid }{#1}{#2}%
32 {\LWR@colorstyle{background:}{#3}{#4}{#5}}%
33 }%
34 \endgroup%
35 }
```

Redirect to new definitions:

```
36 \let\textcolor\LWR@textcolor
37 \let\pagecolor\LWR@pagecolor
38 \let\colorbox\LWR@colorbox
39 \let\fcolorbox\LWR@fcolorbox
```


## Package 93

# lwarp-xfrac.sty

## 165 Xfrac

Pkg **xfrac** Supported by adding xfrac instances.

for HTML output: 1 \LWR@ProvidesPackagePass{xfrac}

 font size In the user's document preamble, **lwarp** should be loaded after font-related setup. During HTML conversion, this font is used by **lwarp** to generate its initial PDF output containing HTML tags, later to be converted by **pdftotext** to a plain text file. While the text may be in any font which **pdftotext** can read, the math is directly converted into SVG images using this same user-selected font. **xfrac** below is set for the Latin Modern (lmr) font. If another font is used, it may be desirable to redefine **\xfracHTMLfontsize** with a different em size.

**\sfrac** [*instance*] {*num*} [*sep*] {*denom*}

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. **\scalebox** is made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a **lateximage**, no adjustments are necessary.

for HTML & PRINT: 2 \begin{warpall}

User-redefinable macro which controls the font size of the fraction.

3 \newcommand\*{\xfracHTMLfontsize}{.6em}

4 \end{warpall}

for HTML output: 5 \begin{warpHTML}

font size A span for a small font, used in the numerator and denominator:

```
6 \newcommand*{\LWR@htmlsmallfontstart}{%
7 \LWR@htmltagc{span style="font-size:\xfracHTMLfontsize"}{}}%
8 \LWR@nestspan%
9 %
10 }
```

```

11
12 \newcommand*{\LWR@htmlsmallfontend}{%
13 \LWR@htmltagc{/span}%
14 \endLWR@nestspan%
15 }

```

`\scalebox` A nullified `\scalebox` command, to avoid introducing HTML scaling tags:

```

16 \NewDocumentCommand{\LWR@noscalebox}{m o m}{#3}

```

instances Instances of `xfrac` for various font choices:

Produce HTML tags for a small superscript numerator and a small (non-subscript) denominator.

Scaling is turned off so that `pdftotext` correctly reads the result.

```

17 \DeclareInstance{xfrac}{default}{text}{
18 numerator-format = {%
19 \let\scalebox\LWR@noscalebox%
20 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
21 denominator-format = {%
22 \let\scalebox\LWR@noscalebox%
23 \LWR@htmlsmallfontstart{ }\,\#1\LWR@htmlsmallfontend},

```

For `pdftotext`, do not scale the text:

```

24 scaling = false
25 }
26 \DeclareInstance{xfrac}{lmr}{text}{
27 numerator-format = {%
28 \let\scalebox\LWR@noscalebox%
29 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
30 denominator-format = {%
31 \let\scalebox\LWR@noscalebox%
32 \LWR@htmlsmallfontstart{ }\,\#1\LWR@htmlsmallfontend},

```

For `pdftotext`, do not scale the text:

```

33 scaling = false
34 }
35 \DeclareInstance{xfrac}{lmss}{text}{
36 numerator-format = {%
37 \let\scalebox\LWR@noscalebox%
38 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
39 denominator-format = {%
40 \let\scalebox\LWR@noscalebox%
41 \LWR@htmlsmallfontstart{ }\,\#1\LWR@htmlsmallfontend},

```

For `pdftotext`, do not scale the text:

```
42 scaling = false
43 }
44 \DeclareInstance{xfrac}{lmtt}{text}{
45 numerator-format = {%
46 \let\scalebox\LWR@noscalebox%
47 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
48 denominator-format = {%
49 \let\scalebox\LWR@noscalebox%
50 \LWR@htmlsmallfontstart{ }\, #1\LWR@htmlsmallfontend},
```

For `pdftotext`, do not scale the text:

```
51 scaling = false
52 }
```

```
53 \end{warpHTML}
```

# Change History and Index

## Change History

v0.10		Docs: Commands for a successful HTML conversion. . . . . 65
General: 2016/03/08 Initial version	1	
v0.11		Docs: Commands into a warpprint environment. . . . . 65
General: 2016/03/11 . . . . .	1	Docs: Newclude limitations. . . 78
Added section:		Docs: Table: Cross-referencing data structures. . . . . 263
Operating-System portability.	94	Docs: Table: Float data structures. . . . . 273
Added section: Selecting the operating system. . . . .	64	Docs: Trademarks section. . . 341
Test Suite: limages and index in README.txt . . . . .	1	Docs: Troubleshooting cross-references. . . . . 86
Test Suite: MS-Windows in README.txt . . . . .	1	Test Suite: Assigned cleveref name for Test Float. . . . . 1
v0.12		Test Suite: Floatrow . . . . . 1
\LWR@newhtmlfile: Bugfix: TOC with numbered files. . . . .	191	v0.15
General: 2016/03/14 . . . . .	1	\printpublished: No div if empty. . . . . 214
Global: Uses \p@(type) in float captions. . . . .	1	\printsubtitle: No div if empty. 215
Test Suite: Sub-figures . . . . .	1	General: 2016/04/06 . . . . . 1
v0.13		Added . . . . . 371
\CaptionSeparator: Fix for newer babel package. . . . .	275	Ampersand (&): Fixed handling when passed as an argument. 239
\LWR@LwarpStart: \up and \fup .	205	Docs: Added warning icons for items needing special attention. 91
\published: Default to empty published. . . . .	211	Docs: Clarify print/HTML output. . . . . 64
\subtitle: Default to empty subtitle. . . . .	211	Docs: Moved the supported functions table to the introduction. . . . . 27
General: 2016/03/24 . . . . .	1	Files: lwarp_formal.css added. . 1
Removed package: subfig . . . . .	1	Fix: steps counter . . . . . 370
Test Suite: Ordinals, Subcaption	1	Fixed & handling. . . . . 368
tikzpicture: Fix dollar-redefined bug for newer package. . . . .	451	Test Suite: test_suite_formal.css file added. . . . . 1
v0.14		v0.16
\LWR@htmlsectionfilename: Fix: Links to home page. . . . .	167	General: 2016/04/11 . . . . . 1
General: 2016/03/31 . . . . .	1	
floatrow: Added. . . . .	366	

<code>\tittingpage</code> : Improved print-output spacing. . . . .	216	<code>\LWR@subhyperrefclass</code> : Improved HTML output linebreaks. . . .	271
Added XeLaTeX, LuaLaTeX support. . . . .	101	<code>\LWR@subinlineimage</code> : Surpress extra space. . . . .	272
Docs: Font and UTF-8 support. 61		General: 2016/05/19 . . . . .	1
Docs: Moved location of <code>\usepackage{lwarp}</code> . . . . .	62	File: lwarp.css: Improved TOC outline display. . . . .	1
Docs: Text not converting. . . .	86	Files: lwarp.css and lwarp_formal.css: Improved responsive design. . . . .	1
Fix: amsmath options clash . .	105	Microtype disabled during HTML generation . . . . .	102
Fix: newtxmath compatibility. 105		PDF Unicode input characters. 92	
Lwarp no longer selects fonts. . . . .	61, 101	Test Suite: Verse package . . . .	1
Removed package: suffix . . . .	1	<code>lateximage</code> : pdfcrop: --hires added. . . . .	304
Test Suite: Improved tittingpage. . . . .	216	Reorganize <code>\HomeHTMLFilename</code> logic. . . . .	304
Test Suite: Lwarp no longer selects fonts. . . . .	1	Surpress extra space. . . . .	304
Test Suite: Supports XeLaTeX, LuaLaTeX. . . . .	1	<code>verse</code> : Supports verse, memoir packages. . . . .	466
<code>xfrac</code> : Adjusted for the use of any font: . . . . .	475	<code>minipage</code> : Fix: <code>\linewidth</code> , <code>\textwidth</code> , <code>\textheight</code> inside a minipage. . . . .	325
v0.17		v0.19	
<code>\LWR@htmlsectionfilename</code> : Fix: Links when entire doc is one HTML page. . . . .	167	<code>\HTMLFilename</code> : Docs: Escape filename underscores. . . . .	166
General: 2016/04/14 . . . . .	1	<code>\HomeHTMLFilename</code> : Docs: Escape filename underscores. . . . .	166
<code>mdframed</code> : Added. . . . .	400	<code>\LWR@LwarpStart</code> : Enabled \\ equal to <code>\newline</code> . . . . .	204
Test Suite: Fix: Print-version front-matter page numbers. . . .	1	<code>\LWR@doubledollar</code> : MathJax support. . . . .	291
Test Suite: Mdfamed . . . . .	1	<code>\LWR@filestart</code> : lwarp_mathjax.txt loaded. . .	201
v0.18		<code>\LWR@hspace</code> : Fix: <code>\hspace</code> length computations. . . . .	334
<code>\LWR@hspace</code> : <code>\hspace</code> supported. 334		<code>\LWR@minipagestartpars</code> : Surpresses paragraph tags between minipages. . . . .	332
<code>\LWR@includegraphicsb</code> : Add: svgz file extension. . . . .	314	<code>\LWR@singledollar</code> : MathJax support. . . . .	291
em, ex, %, px dimensions preserved. . . . .	314	<code>\LWR@tabledatamultirowtag</code> : Added optional args. . . . .	258
Fix: <code>\linewidth</code> , <code>\textwidth</code> , <code>\textheight</code> inside a minipage. . . . .	314	<code>\LateximageFontSizeName</code> : Add: User-adjustable math/lateximage font size. . .	303
Improved HTML output linebreaks. . . . .	314	<code>\minipagefullwidth</code> : Added: No width tag for the next minipage in HTML. . . . .	324
<code>\LWR@myshorttoc</code> : Reorganize <code>\HomeHTMLFilename</code> logic. . .	279		
<code>\LWR@newhtmlfile</code> : sideTOC after title, improving responsive design. . . . .	191		
<code>\LWR@requesttoc</code> : Reorganize <code>\HomeHTMLFilename</code> logic. . .	207		
<code>\LWR@subhyperref</code> : Improved HTML output linebreaks. . . .	271		

<code>\rule</code> : Added	337	<code>\LWR@hspace</code> : Add: Supports HTML thin breakable space.	334
<code>\warpHTMLonly</code> : Added.	95	<code>\LWR@htmldivclass</code> : Added optional style.	174
<code>\warpprintonly</code> : Replaces <code>\rowprintedonly</code> .	95	<code>\LWR@htmlclass</code> : Added optional style.	173
General: 2016/06/08	1	<code>\LWR@htmlsectionfilename</code> : HTMLFilename: removed additional trailing '-', and may be empty.	167
Added.	474	Sections called "Index" or "index" have an underscore prepended to their filenames if no prefix.	167
Avoids MathJax.	290	<code>\LWR@includegraphicsb</code> : Fix: <code>\linewidth</code> in a floatrow.	314
<code>cleveref</code> : Loaded		Fix: Expands filename.	314
<code>\AtEndPreamble</code> .	320	<code>\LWR@longtabledatacaptiontag</code> : Fix: Pars in captions.	256
CSS for table note item.	450	<code>\LWR@section</code> : Combined higher-level sections together into files.	195
Docs: <code>multirow</code> browser bug workaround.	74, 236	<code>\LWR@setOSWindows</code> : Auto-detects operating system.	94
Docs: Math options.	62	<code>\LWR@subhtmlclass</code> : Factored code.	173
Docs: Table: Cross-referencing data structures, updated.	263	<code>\SetHTMLFileNumber</code> : Add: Control file numbers.	166
File: <code>lwarp_mathjax.txt</code> added.	1	<code>\cpagerefFor</code> : User-redefinable word for page references.	321
File: <code>lwarp.css</code> : <code>tnoteitemheader</code> added	1	<code>\dotfill</code> : Inserts an ellipsis.	331
Introduction: MathJax support mentioned.	26	<code>\hfill</code> : Inserts a <code>\qqquad</code> .	331
MathJax support added.	292, 296, 298	<code>\hrulefill</code> : Inserts a short rule.	331
Options: <code>mathsvg</code> and <code>mathjax</code>	96	<code>\hyperindexref</code> : Print mode provided in case <code>hyperref</code> not used.	287
Supports colored <code>\rule</code> .	471	<code>\pageref</code> : Added.	270
titles: null <code>\pagestyle</code> and <code>\thispagestyle</code> for HTML.	105	<code>\tracinglwarp</code> : Added.	109
<code>tikzpicture</code> : Adapts to <code>tikz</code> version.	451	General: 2017/02/09	1
<code>equation</code> : MathJax support.	295	<code>afterpage</code> : Added.	344
v0.20		<code>alltt</code> : Added.	346
<code>\BlockClassSingle</code> : Renamed from "LWR@htmldivclassline".	175	<code>bookmark</code> : Added.	349
<code>\HTMLDescription</code> : Added <code>\NewHTMLdescription</code> . (Renamed in v0.30.)	185	caption and subcaption supported.	1
<code>\HTMLFilename</code> : No longer escape underscores.	166	<code>cleveref</code> and referencing patches: Applied <code>\AfterEndPreamble</code> .	320
<code>\HomeHTMLFilename</code> : No longer escape underscores.	166	<code>draftwatermark</code> : Added.	355
<code>\InlineClass</code> : Renamed from "inlineclass".	176	<code>eso-pic</code> : Added.	360
<code>\LWR@LwarpStart</code> : Fix: math cross references.	205	<code>everypage</code> : Added.	361
<code>\LWR@closeparagraph</code> : <code>\unskip</code> extra spaces.	179	<code>extramarks</code> : Added.	361
No break tags in the start/end of a tabular.	179	<code>fancyhdr</code> : Added.	362
<code>\LWR@endoffline</code> : Fix: <code>\\</code>	332		
<code>\LWR@filestart</code> : Adds meta description.	201		

hyperref: Additional user macros. ....	382	Test Suite: HTML meta descriptions. ....	1
keyfloat: Added. ....	386	Verbatim: Added. ....	227
letterspace: User-interface emulated. ....	388	verbatim: Added. ....	225
listings: Added. ....	391	BlockClass: Added optional style. ....	175
lrcaption: Added. ....	397	Renamed from "blockclass". ..	175
lwarp-newproject: Added. ....	112	LWR@nestspan: Fix: Minipages inside a span. ....	171
microtype: User-interface emulated. ....	407	v0.21	
needspace: Added. ....	412	\LWR@LwarpStart: Changed lateximages to a .txt file. .	204
nowidow: Added. ....	414	\LWR@filestart: Skip title if not given. ....	201
placeins: Added. ....	427	\LWR@newhtmlfile: Skip title if not given. ....	191
ragged2e: Added. ....	428	\marginpar: Fixed source listing. ....	187
setspace: Improved support. ..	430	General: 2017/02/23 .....	1
textpos: Added. ....	445	fontenc: Added. ....	372
titles: Added. ....	452	fontspec: Added. ....	372
titlesec: Added. ....	454	inputenc: Added. ....	385
titletoc: Added. ....	455	newclude: Added. ....	412
titling: Improved compatibility. ....	456	newunicodechar: Added. ....	413
tocloft: Added. ....	457	lwarpmk: Fix: lwarpmk again for Windows. ....	150
wallpaper: Added. ....	468	lwarpmk: Fix: lwarpmk images for Windows. ....	150
wrapfig: Added. ....	469	lwarpmk: Fix: lwarpmk uses lateximages text file instead of shell script. ....	150
Added @, <, > columns. ....	236	Add: Errors for misplaced packages. ....	98
Added single-expansion data arrays. ....	164	Docs: Added internet class. .	32
Code factored into independent lwarp_html files. ....	341	Docs: Added TeX2page, GladTeX. ....	32
Docs: Examples for generating HTML file names. ....	54	Docs: Installing on Windows. .	38
Docs: Improved index. ....	1	File lwarp_tutorial.txt added. ....	43
Enhanced titling support. ....	213	v0.22	
File: lwarp.css: Minor fixes for validation. ....	1	\LWR@parseDcolumn: Added tabular D column. ....	245
File: lwarpmk used to compile print, HTML, indexes, and lateximages. ....	1	\LWR@parsebangcolumn: Added tabular ! column. ....	242
Fix: \linewidth in a floatrow. ....	369	\LWR@parsetablecols: Unknown table column types become l. Added tabular D, !, X columns. ....	247
Improved float caption type handling. ....	363	\LWR@printmccoldata: Added tabular D, !, and X columns. ....	253
Moved sidebar and example code to test suite. ....	1	General: 2017/03/02 .....	1
Page geometry set to 6in wide with large margins. ....	103	abstract: Added. ....	342
Parallel versions of aux files for print/HTML. ....	1	changepage: Added. ....	352
Removed reliance on make, grep, gawk. ....	1	dcolumn: Added. ....	354
Tabular: \unskip extra spaces. ....	236		



<b>enumerate</b> : Added. . . . .	358	<b>emptypage</b> : Added. . . . .	356
<b>ftnright</b> : Added. . . . .	379	<b>framed</b> : Added. . . . .	376
<b>geometry</b> : Nullified commands. . . . .	379	<b>lips</b> : Added. . . . .	390
<b>indentfirst</b> : Added. . . . .	385	<b>mdframed</b> : Help avoid	
<b>layout</b> : Added. . . . .	388	hyphenation. . . . .	401
<b>lscap</b> : Added. . . . .	397	<b>ntheorem</b> : Added. . . . .	415
<b>mcaption</b> : Added. . . . .	399	<b>showidx</b> : Added. . . . .	431
<b>nameref</b> : Added. . . . .	411	<b>theorem</b> : Added. . . . .	446
<b>nextpage</b> : Added. . . . .	413	Basic L <sup>A</sup> T <sub>E</sub> X theorems: improved	
<b>parskip</b> : Added. . . . .	427	CSS. . . . .	231
<b>showkeys</b> : Added. . . . .	431	Docs: Adds credits for patched	
<b>sidecap</b> : Added. . . . .	432	code. . . . .	1
<b>tabularx</b> : Added. . . . .	443	Docs: Testing <b>lwarp</b> . . . . .	85
<b>varioref</b> : Supported. . . . .	70	Fix: Allows XeL <sup>A</sup> T <sub>E</sub> X and	
<b>verse</b> : Added. . . . .	466	LuaL <sup>A</sup> T <sub>E</sub> X to preload graphics	
v0.23		and graphicx. . . . .	99
<b>\LWR@parsetablecols</b> : Fix for vert		v0.26	
bar column type. . . . .	247	General: 2017/03/31 . . . . .	1
<b>\LWR@printmccoldata</b> : Fix for vert		<b>cutwin</b> : Added. . . . .	353
bar column type. . . . .	253	<b>endnotes</b> : Added. . . . .	357
General: 2017/03/02 . . . . .	1	<b>floatflt</b> : Added. . . . .	365
v0.24		<b>footmisc</b> : Added. . . . .	372
<b>\LWR@hspace</b> : Add: <b>\hspace \fill</b>		<b>footnotehyper</b> : Added. . . . .	375
converts to 2em . . . . .	334	<b>footnote</b> : Added. . . . .	374
<b>\LWR@htmlfileref</b> : Fix: Index		<b>marginfix</b> : Added. . . . .	398
links while <b>\tracinglwarp</b> . . . . .	266	<b>marginnote</b> : Added. . . . .	399
<b>\hypertocfloat</b> : List of floats		<b>mparhack</b> : Added. . . . .	408
responds to <b>lofdepth</b> ,		<b>pagenote</b> : Supported as-is. . . . .	426
<b>lotdepth</b> . . . . .	285	<b>sidenotes</b> : Added. . . . .	433
General: 2017/03/15 . . . . .	1	<b>lwarp.css</b> : Improved responsive	
<b>floatrow</b> : Support for <b>subfig</b> . . . . .	366	<b>marginpar</b> and <b>marginblock</b> . . . . .	114
<b>subfig</b> : Added. . . . .	437	Docs: Improved MiK <sub>T</sub> E <sub>X</sub> install	
<b>tikz</b> : For tikz v3.0.0 or later,		instructions. . . . .	35, 38
auto-loads tikz babel library if		Dollar span avoided in a	
necessary. . . . .	451	<b>lateximage</b> . . . . .	290
Docs: Filename underscore. . . . .	52, 66	Footnotes now are L <sup>A</sup> T <sub>E</sub> X boxes	
No longer preloads <b>subcaption</b> ;		instead of <b>pagenotes</b> . . . . .	185
conflicted with <b>subfig</b> . . . . .	104	<b>lateximage</b> : Labels track page	
<b>picture</b> : Fix for inline images. . . . .	322	numbers of <b>lateximages</b> . . . . .	304
<b>tikzpicture</b> : Fix for inline		Print mode now uses a <b>minipage</b>	
images. . . . .	451	of <b>\linewidth</b> . . . . .	304
v0.25		<b>picture</b> : Fix for <b>\makebox</b> in	
<b>\LWR@loadnever</b> : Added the ability		<b>picture</b> . . . . .	322
to prevent conflicting packages. . . . .	99	v0.27	
<b>\addcontentsline</b> : Handles		General: 2017/04/04 . . . . .	1
theorems. . . . .	278	<b>lettrine</b> : Added. . . . .	389
General: 2016/03/22 . . . . .	1	<b>microtype</b> : Fix with XeL <sup>A</sup> T <sub>E</sub> X,	
<b>amsthm</b> : Added. . . . .	347	LuaL <sup>A</sup> T <sub>E</sub> X. . . . .	407
<b>ccaption</b> : Prevented. . . . .	351	<b>soul</b> : Added. . . . .	435
<b>ellipsis</b> : Added. . . . .	355	<b>ulem</b> : Added. . . . .	464

Docs: Installing utilities for MacOS. . . . .	40	Docs: Modifying lwarpmk and index processing. . . . .	85
Docs: Limitations of saveboxes. . . . .	75	File <code>lwarp_mathjax.txt</code> : Updated CDN repository. . . . .	148
Fix for table footnote par tags. . . . .	186	Forced oneside to maintain large right margin. . . . .	103
Page geometry modified to reduce line overflow. . . . .	103		
v0.28		v0.29	
<code>\@wrindex</code> : Improved indexing. . . . .	287	<code>\LWR@includegraphicsb</code> : Fix: Error when no optional arguments. . . . .	314
<code>\HTMLauthor</code> : Added <code>\HTMLauthor</code> . (Renamed in v0.30.) . . . . .	184	General: 2017/04/15 . . . . .	1
<code>\LWR@LwarpEnd</code> : If <code>FormatEPUB</code> or <code>FormatWordProcessor</code> , no bottom nav. . . . .	207	<code>lwarpmk</code> : Add: <code>language</code> option for config files. . . . .	150
<code>\LWR@LwarpStart</code> : <code>FormatWordProcessor</code> forces single-file output. . . . .	204	Add: <code>lwarpmklang</code> option for <code>lwarp</code> . . . . .	96
<code>\LWR@filestart</code> : Adds HTML meta author. . . . .	201	Docs: Using a glossary . . . . .	59
<code>\LWR@forcenewpage</code> : Forces new PDF page before major environments. . . . .	170	File <code>*.lwarpmkconf</code> : Add: <code>language</code> option for config files. . . . .	113
<code>\LWR@htmlcomment</code> : Breaks ligatures in HTML comments. . . . .	172	File <code>lwarpmk.conf</code> : Add: <code>language</code> option for config files. . . . .	112
<code>\LWR@includegraphicsb</code> : Adapts to graphics syntax. . . . .	314	v0.30	
<code>\LWR@newhtmlfile</code> : If <code>FormatEPUB</code> or <code>FormatWordProcessor</code> : skips headers, footers, nav. . . . .	191	<code>\CSSFilename</code> : Renamed from <code>\NewCSS</code> . . . . .	184
<code>\LWR@parsetablecols</code> : Added L, C, R, J column types. . . . .	247	<code>\HTMLauthor</code> : Renamed from <code>\HTMLauthor</code> . . . . .	184
<code>\LWR@startref</code> : Removed space. . . . .	268	<code>\HTMLDescription</code> : Renamed from <code>\NewHTMLdescription</code> . . . . .	185
<code>\chapter</code> : If EPUB, prints footnotes before each section. . . . .	199	<code>\HTMLFirstPageTop</code> : Renamed from <code>\SetFirstPageTop</code> . . . . .	183
<code>\hyperindexref</code> : Improved indexing. . . . .	287	<code>\HTMLLanguage</code> : Renamed from <code>\MetaLanguage</code> . . . . .	201
<code>\textup</code> : Fixed span class. . . . .	329	<code>\HTMLPageBottom</code> : Renamed from <code>\SetPageBottom</code> . . . . .	183
General: 2017/04/14 . . . . .	1	<code>\HTMLPageTop</code> : Renamed from <code>\SetPageTop</code> . . . . .	183
glossaries: Added. . . . .	380	General: 2017/04/29 . . . . .	1
graphics: Added. . . . .	381	<code>lwarp-newproject</code> removed, and combined with <code>lwarp</code> . . . . .	112
<code>tabularx</code> : Fix for optional pos. . . . .	443	<code>lwarpmk</code> : Add: <code>xdyfile</code> configuration option. . . . .	150
<code>tabulary</code> : Added. . . . .	444	<code>lwarpmk</code> : Fix: <code>xindy</code> and <code>texindy</code> adjusted for <code>pdflatex</code> , <code>xelatex</code> and <code>lualatex</code> . . . . .	150
<code>lwarpmk</code> : Add: <code>printglossary</code> and <code>htmlglossary</code> commands. . . . .	150	<code>lwarpmk</code> : Fix: <code>xindy</code> now used for print index generation with <code>latexmk</code> . . . . .	150
Added boolean <code>FormatEPUB</code> . . . . .	110		
Added boolean <code>FormatWordProcessor</code> . . . . .	110		
Added boolean <code>HTMLDebugComments</code> . . . . .	110		
Added boolean <code>HTMLMarkFloats</code> . . . . .	110		

lwarpmk: language now used for both index and glossary generation. ....	150	Option latexmk replaces macro \UseLatexmk. ....	97
File: lwarpmk_html.xdy renamed to lwarpmk_xdy. ....	148	Option lwarpmkclang changed to IndexLanguage. ....	96
Fix: *.css files only written in print mode. ....	114	Option xdyFilename added. ..	97
Fix: lwarpmk_xdy only written in print mode. ....	148	Options HomeHTMLFilename and HTMLFilename replace macros \HomeHTMLFilename and \HTMLFilename. ....	97
Fix: lwarpmk_mathjax.txt only written in print mode. ....	148	v0.31	
Option OSWindows replaces macro \warpOSWindows. ....	97	General: 2017/05/15 .....	1
		keyfloat: Improved	
		compatibility. ....	386

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols			
\\$ .....	<i>290</i>	\$\$ .....	<i>290</i>
\& .....	<i>165</i> , 4582		
\( .....	<u>5618</u>	<b>A</b>	
\) .....	<u>5618</u>	abstract (environment) .....	<u>4201</u>
\@@setcpageref .....	<u>6337</u>	abstract (package) .....	<i>342</i>
\@@setcref .....	<u>6333</u>	\abstractname .....	<i>54</i> , <u>4200</u>
\@@setcrefrange .....	<u>6334</u>	accents .....	<i>193</i>
@author .....	<i>209</i>	\addcontentsline .....	<u>5369</u>
\@begintheorem .....	<u>4439</u>	Adobe (program) .....	<i>34</i>
\@capttype .....	<u>5315</u>	\affiliation .....	<u>3867</u>
\@currentlabelname .....	<u>5098</u>	afterpage (package) .....	<i>344</i>
@date .....	<i>209</i>	algorithmic	
\@dlbfloat .....	<u>5293</u>	with newfloat, trivfloat .....	<i>462</i>
\@endtheorem .....	<u>4449</u>	algorithmicx (package) .....	<i>345</i>
\@float .....	<u>5293</u>	align (environment) .....	<u>5834</u>
\@fnsymbol .....	<u>4161</u>	align* (environment) .....	<u>5856</u>
\@makecaption .....	<u>5317</u>	alltt (package) .....	<i>346</i>
\@maketitle .....	<u>4172</u>	\AmS .....	<u>6686</u>
\@opargbegintheorem .....	<u>4444</u>	amsmath (package) .....	<i>105</i>
@published .....	<i>209</i>	amsthm (package) .....	<i>347</i>
@subtitle .....	<i>209</i>	\and .....	<i>210</i>
@title .....	<i>209</i>	array (package) .....	<i>263</i>
\@wrglossary .....	<u>5566</u>	AsciiDoc (program) .....	<i>33</i>
\@wrindex .....	<u>5559</u>	AsciiDoctor (program) .....	<i>33</i>
\\ .....	<i>332</i>	Asciidoctor-LaTeX (program) .....	<i>33</i>
\$ .....	<i>290</i>	\attrib .....	<i>77</i> , <i>224</i> , <i>466</i>
		\attribution .....	<u>4211</u>

author		
HTML meta tag	68, 184	
\author	67, 210, 3874	
autosec	194	
<b>B</b>		
babel (package)	79	
\backmatter	3528	
\BaseJobname	2882	
BaseJobname (option)	96	
baseline		
tabular	246	
\bfseries	6497	
\BibTeX	6681	
BlockClass (environment)	3099	
\BlockClassSingle	3111	
bookmark (package)	349	
booktabs (package)	351	
booleans:		
CombineHigherDepths	53, 188	
FileSectionNames	54, 166	
FormatEPUB	81, 110	
FormatWordProcessor	83, 110	
HTMLDebugComments	110	
HTMLMarkFloats	83, 110	
LWR@amsmultline	296	
LWR@doingapar	177	
LWR@doinghline	237	
LWR@doingstartpars	177	
LWR@doingtbrule	237	
LWR@freezethisfloat	275	
LWR@infloatrow	313	
LWR@intabularmetadata	238	
LWR@minipagefullwidth	324	
LWR@minipagethispar	324	
LWR@skippingmrowcell	238	
LWR@starredlongtable	255	
LWR@startedrow	237	
LWR@tableparcell	237	
LWR@tracinglwarp	109	
LWR@validtablecol	245	
LWR@verbtags	225	
mathjax	95	
usingOSWindows	94	
warpingHTML	95	
warpingprint	95	
bugs	86	
BVerbatim (environment)	4424	
<b>C</b>		
calc (package)	104	
Calibre	81	
caption (package)	104	
\caption@begin	5348	
\caption@end	5348	
\captionlistentry	5352	
\captionof	5382	
\CaptionSeparator	5316	
ccaption (package)	351	
center (environment)	6056	
\centering	205	
changepage (package)	352	
\chapter	3653	
\citetitle	4218	
classes:		
internet	32	
cleveref (package)	320	
cmap (package)	62	
CombineHigherDepths (boolean)	53, 188	
comment (package)	93	
Computer Modern	61	
\ConTeXt	6678	
counters:		
FileDepth	53, 188	
lofdepth	284	
lotdepth	284	
LWR@externalfilecnt	290	
LWR@htmlfilenumber	167	
LWR@latestautopage	276	
LWR@lateximagedepth	303	
LWR@lateximagenumber	303	
LWR@Llpage	303	
LWR@midrulecounter	251	
LWR@minipagedepth	323	
LWR@nextautofloat	276	
LWR@nextautopage	276	
LWR@nextequation	292	
LWR@prevFileDepth	195	
LWR@spandepth	177	
LWR@tablecolspos	241	
LWR@tablecolswidth	241	
LWR@tabletotalcols	241	
LWR@tabletotalcolsnext	241	
LWR@thisfloat	275	
SideTOCDepth	53, 282	
tocdepth	52	
\cpagerefFor	6336	
cross-references		
missing or incorrect	86	

CSS		
file selection	69	
lwrap.css	68	
per HTML page	69	
project-specific changes	68	
\CSSFilename	51, 54, 69, 3288	
cutwin (package)	353	
<b>D</b>		
danger icon	91	
\date	67	
dcolumn (package)	354	
debugging	86	
HTML debug comments	110	
\DeclareGraphicsExtensions	6086	
\DeclareGraphicsRule	6086	
Deja Vu	61	
description		
HTML meta tag	68, 185	
description (environment)	4511	
displaymath (environment)	5627	
\dotfill	6513	
draftwatermark (package)	355	
<b>E</b>		
ellipsis (package)	355	
\emph	6486	
emptypage (package)	356	
\end@dlbfloat	5307	
\end@float	5307	
endnotes (package)	357	
\enlargethispage	6616	
\enskip	333, 6580	
enumerate (environment)	4480	
enumerate (package)	80, 358	
enumitem (package)	79, 104	
environ (package)	105	
environments:		
abstract	4201	
align	5834	
align*	5856	
BlockClass	3099	
BVerbatim	4424	
center	6056	
description	4511	
displaymath	5627	
enumerate	4480	
equation	5694	
flalign	5878	
flalign*	5900	
flushleft	6068	
flushright	6062	
gather	5790	
gather*	5812	
itemize	4462	
lateximage	5970	
longtable	2	
LVerbatim	4436	
LWR@nestspan	2996	
LWR@tabular	5055	
math	5626	
minipage	6404	
multline	5743	
multline*	5766	
picture	321, 6354	
quote	4226	
theindex	5542	
tikzpicture	8	
titlepage	67, 3908	
titlingpage	67, 3914, 3927	
Verbatim	4291	
verbatim	4247	
VerbatimClass	4283	
verse	2	
warpall	45, 64	
warpHTML	53, 56, 64	
warpprint	46, 55, 64	
epigraph (package)	359	
EPUB		
conversion software	81	
HTML conversion settings	81, 110	
equation (environment)	5694	
error messages	86	
eso-pic (package)	360	
etoolbox (package)	93	
everyhook (package)	103	
everypage (package)	361	
expl3 (package)	103	
extramarks (package)	361	
<b>F</b>		
fancyhdr (package)	362	
fancyvrb (package)	103	
FAQ	86	
fbox		
around a minipage	66	
\fbox	6476	
file		
inaccessible	53	
underscore	52, 66	



viewport ..... 203  
 tabular column conversion ..... 246  
 \HTMLAuthor ..... 55, 68, 3302  
 HTMLDebugComments (boolean) ..... 110  
 \HTMLDescription ..... 55, 68, 3307  
 \HTMLentity ..... 2872  
 \HTMLFilename ..... 2883  
 HTMLFilename (option) ..... 52, 97  
 \HTMLFirstPageTop ..... 55, 3265  
 \HTMLLanguage ..... 55, 3710  
 HTMLleftmargin (length) . 78, 224, 466  
 HTMLMarkFloats (boolean) ..... 83, 110  
 \HTMLPageBottom ..... 55, 210, 3281  
 \HTMLPageTop ..... 55, 3273  
 \HTMLUnicode ..... 2873  
 HTMLvleftskip (length) .. 78, 224, 466  
 \hyperindexref ..... 5573  
 hyperref (package) ..... 270, 382  
 \hypertoc ..... 5484  
 \hypertocfloat ..... 5497

## I

icon  
   warning ..... 91  
 ifplatform (package) ..... 93  
 images  
   graphicx package ..... 309  
   in strange places ..... 87  
 \includegraphics ..... 6269  
 indentfirst (package) ..... 385  
 InDesign (program) ..... 34  
 index  
   language ..... 59  
   processing ..... 47, 48  
   UTF-8 ..... 62  
 IndexLanguage (option) 59, 79, 96, 380  
 \InlineClass ..... 3119  
 inputenc (package) ..... 61, 385  
 internet (class) ..... 32  
 item, list, empty ..... 66  
 itemize (environment) ..... 4462  
 \itshape ..... 6502

## J

JavaScript  
   MathJax ..... 70, 288

## K

keyfloat (package) ..... 386  
 kvoptions (package) ..... 95

## L

label  
   in HTML ..... 205  
   math environment ..... 297  
 language  
   glossary ..... 59  
   index ..... 59  
 language HTML metadata ..... 201  
 \LaTeX ..... 6664  
 LaTeX2HTML (program) ..... 32  
 \LaTeXe ..... 6664  
 lateximage (environment) ..... 5970  
 \LateximageFontSizeName ..... 5934  
 lateximages  
   font size ..... 70, 288, 303  
 latexmk (option) ..... 52, 97  
 LaTeXML (program) ..... 32  
 layout (package) ..... 388  
 lengths:  
   HTMLleftmargin ..... 78, 224, 466  
   HTMLvleftskip ..... 78, 224, 466  
   LWR@minipageheight ..... 323  
   LWR@minipagewidth ..... 323  
   LWR@tempheight ..... 334  
   LWR@tempraise ..... 334  
   LWR@tempwidth ..... 334  
   \VerbatimHTMLWidth ..... 225  
   vleftmargin ..... 78, 224, 466  
   vleftskip ..... 78, 224, 466, 467  
 letterspace (package) ..... 388  
 lettrine (package) ..... 389  
 LibreOffice (program) ..... 33  
 ligatures ..... 61, 62, 102  
 line numbers ..... 91  
 \linebreak ..... 6610  
 \LinkHome ..... 2934  
 Linux (program) ..... 64, 94  
 lips (package) ..... 390  
 list item, empty ..... 66  
 listings (package) ..... 391  
 \listof ..... 5450  
 lmodern (package) ..... 61  
 lofdepth (counter) ..... 284  
 longtable (environment) ..... 2  
 longtable (package) ..... 395  
 lotdepth (counter) ..... 284  
 lscap (package) ..... 397  
 ltcaption (package) ..... 397  
 LuaLaTeX  
   detection ..... 101

file & section names	193	\LWR@endhideamsmath	5736
\LuaLaTeX	6672	\LWR@endminipage	6368
LuaLaTeX (program) [requirement]	36	\LWR@endofline	6557
\LuaTeX	6672	\LWR@endsubminipage	6389
LVerbatim (environment)	4436	\LWR@ensuredoingapar	3160
lwrap		\LWR@externalfilecnt (counter)	290
loading	62	\LWR@filenamenoblanks	3379
options	62	\LWR@filestart	3717
lwrap (package)	62	\LWR@findword	288
lwrap.css (file)	68, 114	\LWR@floatbegin	5277
lwrap.xdy		\LWR@floatend	5295
customizing	69	\LWR@forcenewpage	2968
lwrap.xdy (file)	69, 148	\LWR@freezethisfloat (boolean)	275
lwrap_formal.css (file)	143	\LWR@futurenonospacelet	4539
lwrap_mathjax.txt (file)	148	\LWR@getexparray	2868
lwrap_sagebrush.css (file)	138	\LWR@getmynexttoken	4546
lwrap_tutorial.txt (file)	43	\LWR@hidelatexequation	5669
lwrapmk		\LWR@hspace	6588
customizing	85	\LWR@htmlblockcomment	3041
lwrapmk (option)	97	\LWR@htmlblocktag	3049
lwrapmk (program)	85, 150	\LWR@htmlclosecomment	3027
lwrapmk.conf (file)	112	\LWR@htmlcomment	3034
lwrapmk.lua (file)	85	\LWR@htmldivclass	3076
\LWR@addmathjax	5683	\LWR@htmldivclassend	3079
\LWR@afterendverbatim	4275	\LWR@htmlmlelement	3089
\LWR@amsmultline (boolean)	296	\LWR@htmlmlelementclass	3063
\LWR@atbeginverbatim	4267	\LWR@htmlmlelementclassend	3068
\LWR@beginhideamsmath	5728	\LWR@htmlmlelementclassline	3082
\LWR@botnavigation	2942	\LWR@htmlmlelementend	3092
\LWR@caption@begin	5329	\LWR@htmlfilenumber (counter)	167
\LWR@caption@end	5339	\LWR@htmlfileref	5124
\LWR@clearmidrules	4812	\LWR@htmlmathlabel	5714
\LWR@closeparagraph	3186	\LWR@htmlloopencomment	3027
\LWR@closeprevious	2962	\LWR@htmlrefsectionfilename	2927
\LWR@closetabledatacell	4560	\LWR@HTMLsanitize	5937
\LWR@colafterspec	241	\LWR@HTMLsanitizeexpand	5955
\LWR@colatspec	241	\LWR@htmlsectionfilename	2894
\LWR@colbangspec	241	\LWR@htmlspan	3002
\LWR@colbeforespec	241	\LWR@htmlspanclass	3010
\LWR@copyfile	352	\LWR@htmltag	3018
\LWR@createautosec	3534	\LWR@htmltagc	2989
\LWR@currentcss	3287	\LWR@includegraphicsb	6177
\LWR@descitem	4499	\LWR@indexitem	5547, 5551, 5555
\LWR@docmidrule	4831	\LWR@indexsection	5531
\LWR@doingapar (boolean)	177	\LWR@infloatrow (boolean)	313
\LWR@doinghline (boolean)	237	\LWR@intabularmetadata (boolean)	238
\LWR@doingstartpars (boolean)	177	\LWR@itemizeitem	4455
\LWR@doingtbrule (boolean)	237	\LWR@latestautopage (counter)	276
\LWR@domulticolumn	4904	\LWR@lateximagedepth (counter)	303
\LWR@doubledollar	5596	\LWR@lateximagedepthref	5128



LWR@lateximagenumber (counter) ..	303	LWR@printmccoltype .....	4835
\LWR@lateximagenumberref .....	5131	\LWR@printthetitle .....	4003
LWR@Lpage (counter) .....	303	\LWR@ProvidesPackageDrop .....	342
\LWR@loadafter .....	97	\LWR@ProvidesPackagePass .....	330
\LWR@loadbefore .....	108	\LWR@pushoneclose .....	3537
\LWR@loadnever .....	113	\LWR@requesttoc .....	3837
\LWR@longtabledatacaptiontag ..	4925	\LWR@requirepackagenames .....	287
\LWR@lookforpackagename .....	291	\LWR@restoremathlatexformatting	5651
\LWR@LwarpEnd .....	3844, 6708	\LWR@rotboxorigin .....	6280
\LWR@LwarpStart .....	3774, 6708	\LWR@rotstyle .....	6167
\LWR@maketitlesetup .....	4152	\LWR@rule .....	6617
\LWR@maybeinthisfloat .....	5312	\LWR@scalestyle .....	6170
\LWR@maybeinnewtable row .....	4761	\LWR@section .....	3544
LWR@midrulecounter (counter) ....	251	\LWR@sectionnumber .....	3531
LWR@midrules .....	251	\LWR@setexparray .....	2865
LWR@minipagedepth (counter) ....	323	\LWR@setlatestname .....	5100
LWR@minipagefullwidth (boolean) ..	324	\LWR@setOSWindows .....	30
LWR@minipageheight (length) ....	323	\LWR@sidetoc .....	5463
\LWR@minipagestartpars .....	6562	\LWR@singledollar .....	5608
\LWR@minipagestoppars .....	6568	LWR@skippingmrowcell (boolean) .	238
LWR@minipagethispar (boolean) ...	324	LWR@spandepth (counter) .....	177
LWR@minipagewidth (length) ....	323	\LWR@splabel .....	5134
\LWR@multicolother .....	4856	LWR@starredlongtable (boolean) .	255
\LWR@multicolpartext .....	4851	LWR@startedrow (boolean) .....	237
\LWR@multicolskip .....	4862	\LWR@startnewdepth .....	3538
\LWR@mynexttoken .....	4538	\LWR@startpars .....	3224
\LWR@myshorttoc .....	5394	\LWR@startref .....	5171
\LWR@nameref .....	5121	\LWR@stoppars .....	3236
LWR@nestspan (environment) ....	2996	\LWR@stripperiod .....	5099
\LWR@newhtmlfile .....	3457	\LWR@strresult .....	4612
\LWR@newlabel .....	5158	\LWR@subcmidrule .....	4823
\LWR@newref .....	5207	\LWR@subhtml elementclass .....	3054
LWR@nextautofloat (counter) ....	276	\LWR@subhyperref .....	5231
LWR@nextautopage (counter) ....	276	\LWR@subhyperrefclass .....	5235
LWR@nextequation (counter) ....	292	\LWR@subinlineimage .....	5270
\LWR@openparagraph .....	3165	\LWR@subminipage .....	6383
\LWR@origcolspec .....	4613	\LWR@subnewref .....	5201
\LWR@originname .....	6281	\LWR@subsublabel .....	5137
\LWR@originnames .....	6288	\LWR@subtableofcontents .....	5409
\LWR@parseaftercolumn .....	4670	\LWR@syncmathjax .....	5630
\LWR@parseatcolumn .....	4618	\LWR@tablecolspec .....	4611
\LWR@parsebangcolumn .....	4640	LWR@tablecolspos (counter) .....	241
\LWR@parsebeforecolumn .....	4661	LWR@tablecolwidth (counter) ....	241
\LWR@parseDcolumn .....	4697	\LWR@tabledatacolumn tag .....	5009
\LWR@parsenormalcolumn .....	4683	\LWR@tabledata multicolumn tag ..	4989
\LWR@parsepcolumn .....	4693	\LWR@tabledata multirow tag .....	4995
\LWR@parseskipcolumn .....	4680	\LWR@tabledata singlecolumn tag ..	4777
\LWR@parsetablecols .....	4702	LWR@tableparcell (boolean) ....	237
LWR@prevFileDepth (counter) ....	195	LWR@tabletotalcols (counter) ....	241
\LWR@printmccoldata .....	4865	LWR@tabletotalcolsnext (counter) .	241



<code>\numberline</code> .....	5481	<code>enumitem</code> .....	79, 104
numbers		<code>environ</code> .....	105
left margin .....	91	<code>epigraph</code> .....	359
<b>O</b>			
OpenOffice (program) .....	33	<code>eso-pic</code> .....	360
options:		<code>etoolbox</code> .....	93
BaseJobname .....	96	<code>everyhook</code> .....	103
HomeHTMLFilename .....	52, 97	<code>everypage</code> .....	361
HTMLFilename .....	52, 97	<code>expl3</code> .....	103
IndexLanguage ....	59, 79, 96, 380	<code>extramarks</code> .....	361
latexmk .....	52, 97	<code>fancyhdr</code> .....	362
lwarpmk .....	97	<code>fancyvrb</code> .....	103
mathjax .....	52, 62, 96	<code>float</code> .....	363
mathsvg .....	52, 62, 96	<code>floatflt</code> .....	365
OSWindows .....	64, 94, 97	<code>floatrow</code> .....	366
warpHTML .....	62, 96	<code>fontenc</code> .....	61, 372
warpprint .....	62, 95	<code>fontspec</code> .....	61, 372
xdyFilename .....	69, 97	<code>footmisc</code> .....	373
<code>\OSPathSymbol</code> .....	29	<code>footnote</code> .....	374
OSWindows (option) .....	64, 94, 97	<code>footnotehyper</code> .....	375
<b>P</b>			
packages		<code>framed</code> .....	376
required .....	101	<code>ftnright</code> .....	379
packages:		<code>geometry</code> .....	102, 379
abstract .....	342	<code>getttitlestring</code> .....	103
afterpage .....	344	<code>glossaries</code> .....	79, 380
algorithmicx .....	345	<code>graphics</code> .....	309, 381
alltt .....	346	<code>graphicx</code> .....	309, 381
amsmath .....	105	<code>hyperref</code> .....	270, 382
amsthm .....	347	<code>ifplatform</code> .....	93
array .....	263	<code>indentfirst</code> .....	385
babel .....	79	<code>inputenc</code> .....	61, 385
bookmark .....	349	<code>keyfloat</code> .....	386
booktabs .....	351	<code>kvoptions</code> .....	95
calc .....	104	<code>layout</code> .....	388
caption .....	104	<code>letterspace</code> .....	388
ccaption .....	351	<code>lettrine</code> .....	389
changepage .....	352	<code>lips</code> .....	390
cleveref .....	320	<code>listings</code> .....	391
cmap .....	62	<code>lmodern</code> .....	61
comment .....	93	<code>longtable</code> .....	395
cutwin .....	353	<code>lscape</code> .....	397
dcolumn .....	354	<code>ltcaption</code> .....	397
draftwatermark .....	355	<code>lwrap</code> .....	62
ellipsis .....	355	<code>makeidx</code> .....	104
emptypage .....	356	<code>marginfix</code> .....	398
endnotes .....	357	<code>marginnote</code> .....	399
enumerate .....	80, 358	<code>mcaption</code> .....	399
		<code>mdframed</code> .....	400
		<code>memoir</code> .....	467
		<code>microtype</code> .....	62, 102, 407
		<code>mparhack</code> .....	408

multicol	409	\pagebreak	6612
multirow	258, 411	pagenote (package)	426
nameref	411	\pagenumbering	280
needspace	412	\pageref	5214
newclude	78, 412	\pagerefPageFor	5213
newfloat	104	\pagestyle	276
newtxmath	78	Pandoc (program)	33
newunicodechar	61, 413	\paragraph	3692
nextpage	413	\parbox	6461
nowidow	414	\parsemulticolumnalignment	4892
ntheorem	415	parskip (package)	427
pagenote	426	\part	3641
parskip	427	pdfcrop (program) [requirement]	36
placeins	427	pdfLaTeX (program) [requirement]	36
ragged2e	428	pdfseparate (program) [requirement]	36, 41
refcount	104	pdftocairo (program) [requirement]	36, 41
rotating	429	pdftotext (program) [requirement]	36, 41
setspace	430	Perl	41
showidx	431	perl (program) [requirement]	41
showkeys	431	\phantomsection	6658
sidecap	432	picture (environment)	321, 6354
sidenotes	433	placeins (package)	427
siunitx	78, 309	Plastex (program)	32
soul	435	\popclose	2838
subfig	437	Poppler	36, 41
tabularx	443	\postpublished	209
tabulary	444	\postsuffix	209
textcomp	62	\prepublished	209
textpos	445	\presuffix	209, 218
theorem	446	\printauthor	210, 4020, 4035
threeparttable	450	\printdate	210, 4025, 4037
tikz	451	\printindex	5534
titlesps	105, 452	\printpublished	210, 3986, 4032
titlesec	454	\PrintStack	2947
titletoc	455	\printsuffix	210, 4011, 4034
titling	104, 456	\printthanks	209, 213
tocloft	457	\printtitle	210, 3995, 4033
trivfloat	462	problems	86
ulem	464	programs	
verse	466, 467	utility	35
wallpaper	468	programs:	
wrapfig	469	[requirement]:	
xcolor	471	LuaLaTeX	36
xfrac	106, 474	MathJax	36
xifthen	104	pdfcrop	36
xparse	103	pdfLaTeX	36
xstring	104	pdfseparate	36, 41
zref	105	pdftocairo	36, 41
page			
inaccessible	53		





<b>V</b>	
Verbatim (environment) . . . . .	<a href="#">4291</a>
verbatim (environment) . . . . .	<a href="#">4247</a>
VerbatimClass (environment) . . . .	<a href="#">4283</a>
\VerbatimHTMLWidth (length) . . . .	<a href="#">225</a>
verse (environment) . . . . .	<a href="#">2</a>
verse (package) . . . . .	<a href="#">466</a> , <a href="#">467</a>
viewport	
HTML meta tag . . . . .	<a href="#">203</a>
vleftmargini (length) . . . .	<a href="#">78</a> , <a href="#">224</a> , <a href="#">466</a>
vleftskip (length) . . . .	<a href="#">78</a> , <a href="#">224</a> , <a href="#">466</a> , <a href="#">467</a>
<b>W</b>	
wallpaper (package) . . . . .	<a href="#">468</a>
warning icon . . . . .	<a href="#">91</a>
warpall (environment) . . . . .	<a href="#">45</a> , <a href="#">64</a>
warpHTML (environment) . . . .	<a href="#">53</a> , <a href="#">56</a> , <a href="#">64</a>
warpHTML (option) . . . . .	<a href="#">62</a> , <a href="#">96</a>
\warpHTMLonly . . . . .	<a href="#">44</a> , <a href="#">56</a> , <a href="#">65</a>
warpingHTML (boolean) . . . . .	<a href="#">95</a>
warpingprint (boolean) . . . . .	<a href="#">95</a>
warpprint (environment) . . . .	<a href="#">46</a> , <a href="#">55</a> , <a href="#">64</a>
warpprint (option) . . . . .	<a href="#">62</a> , <a href="#">95</a>
\warpprintonly . . . . .	<a href="#">43</a> , <a href="#">56</a> , <a href="#">65</a>
Windows (program) . . . . .	<a href="#">64</a> , <a href="#">94</a>
Word (program) . . . . .	<a href="#">33</a>
word processor	
HTML conversion settings . .	<a href="#">83</a> , <a href="#">110</a>
wrapfig (package) . . . . .	<a href="#">469</a>
<b>X</b>	
xcolor (package) . . . . .	<a href="#">471</a>
xdyFilename (option) . . . . .	<a href="#">69</a> , <a href="#">97</a>
XeLaTeX	
detection . . . . .	<a href="#">101</a>
file & section names . . . . .	<a href="#">193</a>
\XeLaTeX . . . . .	<a href="#">6674</a>
XeLaTeX (program) [requirement] . . .	<a href="#">36</a>
\XeTeX . . . . .	<a href="#">6674</a>
xfrac (package) . . . . .	<a href="#">106</a> , <a href="#">474</a>
xifthen (package) . . . . .	<a href="#">104</a>
xindy	
customizing . . . . .	<a href="#">69</a>
xindy (program) . . . . .	<a href="#">69</a>
xparse (package) . . . . .	<a href="#">103</a>
xstring (package) . . . . .	<a href="#">104</a>
<b>Z</b>	
zref (package) . . . . .	<a href="#">105</a>