



# The **lwarp** package

v0.27 — 2017/04/04

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

© 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.

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>19</b>
<b>1</b>	<b>Updates</b>	<b>19</b>
<b>2</b>	<b>Introduction</b>	<b>22</b>
2.1	Supported packages and features . . . . .	23
<b>3</b>	<b>Alternatives</b>	<b>28</b>
3.1	Internet class . . . . .	28
3.2	TeX4ht . . . . .	28
3.3	Translators . . . . .	28
3.4	AsciiDoc . . . . .	29
3.5	Pandoc . . . . .	29
3.6	Word Processors . . . . .	29
3.7	Commercial Systems . . . . .	30
3.8	Comparisons . . . . .	30
<b>4</b>	<b>Installation</b>	<b>31</b>
4.1	Installing the lwarp package . . . . .	31
4.2	Installing the lwarpmk utility . . . . .	34
4.2.1	Using a local copy of lwarpmk . . . . .	35
4.3	Installing additional utilities . . . . .	36
<b>5</b>	<b>Tutorial</b>	<b>38</b>
5.1	Starting a new project . . . . .	38
5.2	Compiling the print version with lwarpmk . . . . .	42
5.3	Compiling the HTML version with lwarpmk . . . . .	43
5.4	Generating the SVG images . . . . .	44
5.5	Using MathJax for math . . . . .	45
5.6	Changing the CSS style . . . . .	46
5.7	Customizing the HTML output . . . . .	47
5.8	Using latexmk . . . . .	51
5.9	Using XeLaTeX or LuaLaTeX . . . . .	52
5.10	Cleaning auxiliary files . . . . .	53
5.11	Cleaning auxiliary and output files . . . . .	53
5.12	Processing multiple projects in the same directory . . . . .	53
5.13	Using the make utility . . . . .	53
<b>6</b>	<b>Additional details</b>	<b>54</b>
6.1	Font and UTF-8 support . . . . .	54
6.2	lwarp package loading and options . . . . .	55
6.3	Selecting the operating system . . . . .	55
6.4	Selecting actions for print or HTML output . . . . .	55
6.5	Commands to be placed into the warpprint environment . . . . .	56

6.6	Commands for a successful HTML conversion . . . . .	57
6.7	Title page . . . . .	58
6.8	HTML page meta descriptions . . . . .	59
6.9	CSS . . . . .	60
6.10	Special cases and limitations . . . . .	60
6.10.1	Text formatting . . . . .	60
6.10.2	Footnotes and page notes . . . . .	60
6.10.3	Math . . . . .	61
6.10.4	Ntheorem . . . . .	62
6.10.5	Graphics . . . . .	63
6.10.6	Xcolor . . . . .	63
6.10.7	Tabular . . . . .	63
6.10.8	Save Boxes . . . . .	65
6.10.9	Minipages . . . . .	65
6.10.10	Mdframed . . . . .	65
6.10.11	float, trivfloat, and/or algorithmicx together . . . . .	66
6.10.12	caption and subcaption packages . . . . .	66
6.10.13	subfig package . . . . .	66
6.10.14	floatrow package . . . . .	67
6.10.15	abstract package . . . . .	67
6.10.16	cleveref and varioref packages . . . . .	67
6.10.17	verse and memoir . . . . .	67
6.10.18	siunitx package . . . . .	68
6.10.19	newclude package . . . . .	68
6.10.20	newtxmath package . . . . .	68
6.10.21	babel package . . . . .	69
6.10.22	enumitem package . . . . .	69
6.10.23	enumerate package . . . . .	69
<b>7</b>	<b>Expanding the code base</b>	<b>69</b>
7.1	Creating an lwarp version of a package . . . . .	70
7.2	Testing lwarp . . . . .	70
<b>8</b>	<b>Troubleshooting</b>	<b>71</b>
8.1	Using the <code>lwarp.sty</code> package . . . . .	71
8.1.1	Debug tracing output . . . . .	73
8.2	Compiling the <code>lwarp.dtx</code> file . . . . .	73
<b>9</b>	<b>Implementation</b>	<b>73</b>
<b>10</b>	<b>Stack depths</b>	<b>74</b>
<b>11</b>	<b>Source Code</b>	<b>76</b>
<b>12</b>	<b>Detecting the T<sub>E</sub>X Engine — pdf<sub>l</sub>atex, lua<sub>l</sub>atex, x<sub>e</sub>latex</b>	<b>77</b>
<b>13</b>	<b>Unicode Input Characters</b>	<b>77</b>

<b>14</b>	<b>Handling package options</b>	<b>77</b>
<b>15</b>	<b>Misplaced packages</b>	<b>80</b>
<b>16</b>	<b>Required packages</b>	<b>83</b>
<b>17</b>	<b>Loading Packages</b>	<b>88</b>
<b>18</b>	<b>Copying a file</b>	<b>90</b>
<b>19</b>	<b>Debugging messages</b>	<b>91</b>
<b>20</b>	<b>Remembering original formatting macros</b>	<b>91</b>
<b>21</b>	<b>Operating-System portability</b>	<b>92</b>
21.1	Common portability code . . . . .	92
21.2	Unix, Linux, and Mac OS . . . . .	93
21.3	MS-Windows . . . . .	93
<b>22</b>	<b>Stacks</b>	<b>93</b>
22.1	Assigning depths . . . . .	94
22.2	Closing actions . . . . .	94
22.3	Closing depths . . . . .	95
22.4	Pushing and popping the stack . . . . .	95
<b>23</b>	<b>Data arrays</b>	<b>97</b>
<b>24</b>	<b>HTML entities</b>	<b>97</b>
<b>25</b>	<b>HTML filename generation</b>	<b>98</b>
<b>26</b>	<b>Homepage link</b>	<b>100</b>
<b>27</b>	<b>\PrintStack diagnostic tool</b>	<b>101</b>
<b>28</b>	<b>Closing stack levels</b>	<b>101</b>
<b>29</b>	<b>HTML tags, spans, divs, elements</b>	<b>102</b>
29.1	Mapping L <sup>A</sup> T <sub>E</sub> X Sections to HTML Sections . . . . .	102
29.2	HTML tags . . . . .	103
29.3	Block tags and comments . . . . .	104
29.4	Div class and element class . . . . .	105
29.5	Single-line elements . . . . .	106
29.6	HTML5 semantic elements . . . . .	107
29.7	High-level block and inline classes . . . . .	107
29.8	Closing HTML tags . . . . .	108
<b>30</b>	<b>Paragraph handling</b>	<b>109</b>

<b>31</b>	<b>Paragraph start/stop handling</b>	<b>112</b>
<b>32</b>	<b>Page headers and footers</b>	<b>115</b>
<b>33</b>	<b>CSS</b>	<b>116</b>
<b>34</b>	<b>HTML meta description</b>	<b>116</b>
<b>35</b>	<b>Footnotes</b>	<b>117</b>
35.1	Regular page footnotes . . . . .	117
35.2	Minipage footnotes . . . . .	117
35.3	Titlepage thanks . . . . .	117
35.4	Regular page footnote implementation . . . . .	118
<b>36</b>	<b>Marginpars</b>	<b>119</b>
<b>37</b>	<b>Splitting HTML files</b>	<b>119</b>
<b>38</b>	<b>Sectioning</b>	<b>124</b>
38.1	Book class commands . . . . .	124
38.2	Sectioning support macros . . . . .	125
38.3	\section and friends . . . . .	130
<b>39</b>	<b>Starting a new file</b>	<b>131</b>
<b>40</b>	<b>Starting HTML output</b>	<b>134</b>
<b>41</b>	<b>Ending HTML output</b>	<b>137</b>
<b>42</b>	<b>Titles and the titling package</b>	<b>138</b>
42.1	Setting the title, etc. . . . .	139
42.2	Changes to HTML titlepage and titlingpage . . . . .	142
42.3	Printing the title, etc. in HTML . . . . .	143
42.4	Printing the title, etc. in print form . . . . .	145
42.5	\maketitle for print output . . . . .	146
42.6	\maketitle for HTML output . . . . .	149
<b>43</b>	<b>Abstract</b>	<b>152</b>
<b>44</b>	<b>Quote and verse</b>	<b>152</b>
44.1	Citations and attributions . . . . .	152
44.2	Quotes, quotations . . . . .	153
44.3	Verse . . . . .	153
<b>45</b>	<b>Verbatim</b>	<b>154</b>
<b>46</b>	<b>Fancyvrb</b>	<b>154</b>

<b>47</b>	<b>Theorems</b>	<b>160</b>
<b>48</b>	<b>Lists</b>	<b>161</b>
48.1	Itemize . . . . .	161
48.2	Enumerate . . . . .	162
48.3	Description . . . . .	163
<b>49</b>	<b>Tabular</b>	<b>164</b>
49.1	Token lookahead . . . . .	166
49.2	Booleans . . . . .	166
49.3	Handling & and ! . . . . .	167
49.4	Handling \\ . . . . .	169
49.5	Variables . . . . .	170
49.6	Parsing @, >, <, ! columns . . . . .	171
49.7	Parsing 'l', 'c', or 'r' columns . . . . .	173
49.8	Parsing 'p', 'm', or 'b' columns . . . . .	174
49.9	Parsing 'D' columns . . . . .	174
49.10	Parsing the column specifications . . . . .	174
49.11	Starting a new row . . . . .	178
49.12	Data opening tag . . . . .	179
49.13	Midrules . . . . .	180
49.14	Multicolumns . . . . .	181
49.14.1	Parsing multicolumns . . . . .	181
49.14.2	High-level multicolumn interface . . . . .	184
49.14.3	Longtable captions . . . . .	185
49.14.4	\tabulatedatamulticolumntag . . . . .	187
49.15	Multirow . . . . .	188
49.16	Utility macros inside a table . . . . .	188
49.17	Checking for a new table cell . . . . .	188
49.18	\mrowcell . . . . .	190
49.19	New \tabular definition . . . . .	190
49.20	Array . . . . .	192
<b>50</b>	<b>Cross-references</b>	<b>193</b>
50.1	Setup . . . . .	193
50.2	Zref setup . . . . .	195
50.3	Labels . . . . .	196
50.4	References . . . . .	197
50.5	Hyper-references . . . . .	200
<b>51</b>	<b>Floats</b>	<b>203</b>
51.1	Float captions . . . . .	204
51.1.1	Caption inside a float environment . . . . .	205
51.1.2	Caption and LOF linking and tracking . . . . .	205
<b>52</b>	<b>Table of Contents, LOF, LOT</b>	<b>209</b>

52.1	Reading and printing the TOC . . . . .	209
52.2	High-level TOC commands . . . . .	211
52.3	Side TOC . . . . .	212
52.4	Low-level TOC line formatting . . . . .	213
<b>53</b>	<b>Index</b>	<b>215</b>
<b>54</b>	<b>Math</b>	<b>217</b>
54.1	Inline and display math . . . . .	219
54.2	MathJax support . . . . .	221
54.3	Equation environment . . . . .	224
54.4	AMS Math environments . . . . .	225
54.4.1	Support macros . . . . .	225
54.4.2	Environment patches . . . . .	227
<b>55</b>	<b>Lateximages</b>	<b>231</b>
<b>56</b>	<b>center, flushleft, flushright</b>	<b>237</b>
<b>57</b>	<b>Siunitx</b>	<b>238</b>
<b>58</b>	<b>Graphics</b>	<b>238</b>
58.1	<code>\graphicspath</code> . . . . .	239
58.2	Length conversions and graphics options . . . . .	239
58.3	<code>\includegraphics</code> . . . . .	242
58.4	<code>\rotatebox</code> , <code>\scalebox</code> , <code>\reflectbox</code> . . . . .	246
58.5	Null functions . . . . .	248
<b>59</b>	<b>Cleverref</b>	<b>249</b>
<b>60</b>	<b>Picture</b>	<b>250</b>
<b>61</b>	<b>Boxes and Minipages</b>	<b>251</b>
61.1	Counters and lengths . . . . .	251
61.2	Footnote handling . . . . .	252
61.3	Minipage handling . . . . .	252
61.4	<code>Parbox</code> , <code>makebox</code> , <code>framebox</code> , <code>fbox</code> , <code>raisebox</code> . . . . .	256
<b>62</b>	<b>Direct formatting</b>	<b>257</b>
<b>63</b>	<b>Skips, spaces, font sizes</b>	<b>260</b>
<b>64</b>	<b><code>\phantomsection</code></b>	<b>266</b>
<b>65</b>	<b><code>\LaTeX</code> and other logos</b>	<b>266</b>
65.1	HTML logos . . . . .	267
65.2	Print logos . . . . .	268

<b>66</b>	<b>\AtBeginDocument, \AtEndDocument</b>	<b>269</b>
<b>67</b>	<b>lwarpmk.conf</b>	<b>270</b>
<b>68</b>	<b>Trademarks</b>	<b>271</b>
<b>2</b>	<b>lwarp-newproject.sty</b>	<b>272</b>
<b>69</b>	<b>lwarp-newproject</b>	<b>272</b>
69.1	project_html.tex . . . . .	274
69.2	project.lwarpmkconf . . . . .	274
69.3	lwarp.css . . . . .	275
69.4	lwarp_sagebrush.css . . . . .	299
69.5	lwarp_formal.css . . . . .	304
69.6	sample_project.css . . . . .	308
69.7	lwarp_html.xdy . . . . .	309
69.8	lwarp_mathjax.txt . . . . .	309
69.9	Lwarpmk option . . . . .	311
<b>3</b>	<b>lwarp-abstract.sty</b>	<b>321</b>
<b>70</b>	<b>Abstract</b>	<b>321</b>
<b>4</b>	<b>lwarp-afterpage.sty</b>	<b>323</b>
<b>71</b>	<b>Afterpage</b>	<b>323</b>
<b>5</b>	<b>lwarp-algorithmicx.sty</b>	<b>324</b>
<b>72</b>	<b>Algorithmicx</b>	<b>324</b>
<b>6</b>	<b>lwarp-alltt.sty</b>	<b>324</b>
<b>73</b>	<b>Alltt</b>	<b>325</b>
<b>7</b>	<b>lwarp-amsthm.sty</b>	<b>326</b>
<b>74</b>	<b>AMSthm</b>	<b>326</b>



---

<b>8</b>	<b>lwarp-bookmark.sty</b>	<b>328</b>
75	Bookmark	328
<b>9</b>	<b>lwarp-booktabs.sty</b>	<b>330</b>
76	Booktabs	330
<b>10</b>	<b>lwarp-ccaption.sty</b>	<b>330</b>
77	Ccaption	330
<b>11</b>	<b>lwarp-changepage.sty</b>	<b>331</b>
78	Changepage	331
<b>12</b>	<b>lwarp-cutwin.sty</b>	<b>332</b>
79	Cutwin	332
<b>13</b>	<b>lwarp-dcolumn.sty</b>	<b>332</b>
80	Dcolumn	333
<b>14</b>	<b>lwarp-draftwatermark.sty</b>	<b>334</b>
81	Draftwatermark	334
<b>15</b>	<b>lwarp-ellipsis.sty</b>	<b>334</b>
82	Ellipsis	334
<b>16</b>	<b>lwarp-emptypage.sty</b>	<b>335</b>
83	Emptypage	335
<b>17</b>	<b>lwarp-endnotes.sty</b>	<b>336</b>
84	Endnotes	336

---

<b>18</b>	<b>lwarp-enumerate.sty</b>	<b>337</b>
85	Enumerate	337
<b>19</b>	<b>lwarp-epigraph.sty</b>	<b>338</b>
86	Epigraph	338
<b>20</b>	<b>lwarp-eso-pic.sty</b>	<b>339</b>
87	Eso-pic	339
<b>21</b>	<b>lwarp-everypage.sty</b>	<b>340</b>
88	Everypage	340
<b>22</b>	<b>lwarp-extramarks.sty</b>	<b>340</b>
89	Extramarks	340
<b>23</b>	<b>lwarp-fancyhdr.sty</b>	<b>341</b>
90	Fancyhdr	341
<b>24</b>	<b>lwarp-float.sty</b>	<b>342</b>
91	Float and \newfloat	342
<b>25</b>	<b>lwarp-floatflt.sty</b>	<b>344</b>
92	Floatflt	344
<b>26</b>	<b>lwarp-floatrow.sty</b>	<b>345</b>
93	Floatrow	345
<b>27</b>	<b>lwarp-fontenc.sty</b>	<b>351</b>
94	Fontenc	351

---

<b>28</b>	<b>lwarp-fontspec.sty</b>	<b>351</b>
95	Fontspec	351
<b>29</b>	<b>lwarp-footmisc.sty</b>	<b>351</b>
96	Footmisc	351
<b>30</b>	<b>lwarp-footnote.sty</b>	<b>353</b>
97	Footnote	353
<b>31</b>	<b>lwarp-footnotehyper.sty</b>	<b>354</b>
98	Footnotehyper	354
<b>32</b>	<b>lwarp-framed.sty</b>	<b>355</b>
99	Framed	355
<b>33</b>	<b>lwarp-ftnright.sty</b>	<b>358</b>
100	Ftnright	358
<b>34</b>	<b>lwarp-geometry.sty</b>	<b>358</b>
101	Geometry	358
<b>35</b>	<b>lwarp-graphics.sty</b>	<b>359</b>
102	Graphics	359
<b>36</b>	<b>lwarp-graphicx.sty</b>	<b>359</b>
103	Graphicx	359
<b>37</b>	<b>lwarp-hyperref.sty</b>	<b>360</b>
104	Hyperref	360

---

<b>38</b>	<b>lwarp-indentfirst.sty</b>	<b>363</b>
105	Indentfirst	363
<b>39</b>	<b>lwarp-inputenc.sty</b>	<b>363</b>
106	Inputenc	363
<b>40</b>	<b>lwarp-keyfloat.sty</b>	<b>364</b>
107	Keyfloat	364
<b>41</b>	<b>lwarp-layout.sty</b>	<b>364</b>
108	Layout	365
<b>42</b>	<b>lwarp-letterspace.sty</b>	<b>365</b>
109	Letterspace	365
<b>43</b>	<b>lwarp-lettrine.sty</b>	<b>366</b>
110	Lettrine	366
<b>44</b>	<b>lwarp-lips.sty</b>	<b>367</b>
111	Lips	367
<b>45</b>	<b>lwarp-listings.sty</b>	<b>368</b>
112	Listings	368
<b>46</b>	<b>lwarp-longtable.sty</b>	<b>372</b>
113	Longtable	372
<b>47</b>	<b>lwarp-lscape.sty</b>	<b>374</b>
114	Lscape	374

---

<b>48</b>	<b>lwarp-ltcaption.sty</b>	<b>374</b>
115	Ltcaption	374
<b>49</b>	<b>lwarp-marginfix.sty</b>	<b>375</b>
116	Marginfix	375
<b>50</b>	<b>lwarp-marginnote.sty</b>	<b>376</b>
117	Marginnote	376
<b>51</b>	<b>lwarp-mcaption.sty</b>	<b>376</b>
118	Mcaption	376
<b>52</b>	<b>lwarp-mdframed.sty</b>	<b>377</b>
119	Mdframed	377
<b>53</b>	<b>lwarp-microtype.sty</b>	<b>384</b>
120	Microtype	384
<b>54</b>	<b>lwarp-mparhack.sty</b>	<b>385</b>
121	Mparhack	385
<b>55</b>	<b>lwarp-multicol.sty</b>	<b>386</b>
122	Multicol	386
<b>56</b>	<b>lwarp-multirow.sty</b>	<b>388</b>
123	Multirow	388
<b>57</b>	<b>lwarp-nameref.sty</b>	<b>388</b>
124	Nameref	388

---

<b>58</b>	<b>lwarp-needspace.sty</b>	<b>389</b>
125	Needspace	389
<b>59</b>	<b>lwarp-newclude.sty</b>	<b>389</b>
126	Newclude	389
<b>60</b>	<b>lwarp-newunicodechar.sty</b>	<b>390</b>
127	Newunicodechar	390
<b>61</b>	<b>lwarp-nextpage.sty</b>	<b>390</b>
128	Nextpage	390
<b>62</b>	<b>lwarp-nowidow.sty</b>	<b>390</b>
129	Nowidow	391
<b>63</b>	<b>lwarp-ntheorem.sty</b>	<b>392</b>
130	Ntheorem	392
<b>64</b>	<b>lwarp-pagenote.sty</b>	<b>403</b>
131	Pagenote	403
<b>65</b>	<b>lwarp-parskip.sty</b>	<b>404</b>
132	Parskip	404
<b>66</b>	<b>lwarp-placeins.sty</b>	<b>404</b>
133	Placeins	404
<b>67</b>	<b>lwarp-ragged2e.sty</b>	<b>405</b>
134	Ragged2e	405

---

<b>68</b>	<b>lwarp-rotating.sty</b>	<b>406</b>
135	Rotating	406
<b>69</b>	<b>lwarp-setspace.sty</b>	<b>407</b>
136	Setspace	407
<b>70</b>	<b>lwarp-showidx.sty</b>	<b>407</b>
137	Showidx	408
<b>71</b>	<b>lwarp-showkeys.sty</b>	<b>408</b>
138	Showkeys	408
<b>72</b>	<b>lwarp-sidecap.sty</b>	<b>409</b>
139	Sidecap	409
<b>73</b>	<b>lwarp-sidenotes.sty</b>	<b>410</b>
140	Sidenotes	410
<b>74</b>	<b>lwarp-soul.sty</b>	<b>412</b>
141	Soul	412
<b>75</b>	<b>lwarp-subfig.sty</b>	<b>414</b>
142	Subfig	414
<b>76</b>	<b>lwarp-tabularx.sty</b>	<b>420</b>
143	Tabularx	420
<b>77</b>	<b>lwarp-textpos.sty</b>	<b>421</b>
144	Textpos	421

<b>78</b>	<b>lwarp-theorem.sty</b>	<b>422</b>
145	Theorem	422
<b>79</b>	<b>lwarp-threeparttable.sty</b>	<b>426</b>
146	Threeparttable	426
<b>80</b>	<b>lwarp-tikz.sty</b>	<b>427</b>
147	Tikz	427
<b>81</b>	<b>lwarp-titleps.sty</b>	<b>428</b>
148	Titleps	428
<b>82</b>	<b>lwarp-titlesec.sty</b>	<b>430</b>
149	Titlesec	430
<b>83</b>	<b>lwarp-titletoc.sty</b>	<b>431</b>
150	Titletoc	431
<b>84</b>	<b>lwarp-titling.sty</b>	<b>432</b>
151	Titling	432
<b>85</b>	<b>lwarp-tocloft.sty</b>	<b>433</b>
152	Tocloft	433
<b>86</b>	<b>lwarp-trivfloat.sty</b>	<b>438</b>
153	Trivfloat	438
153.1	Combining \newfloat, \trivfloat, and algorithmicx . . . . .	438



---

<b>87</b>	<b>lwarp-ulem.sty</b>	<b>440</b>
154	Ulem	440
<b>88</b>	<b>lwarp-verse.sty</b>	<b>442</b>
155	Verse	442
<b>89</b>	<b>lwarp-wallpaper.sty</b>	<b>444</b>
156	Wallpaper	444
<b>90</b>	<b>lwarp-wrapfig.sty</b>	<b>445</b>
157	Wrapfig	445
<b>91</b>	<b>lwarp-xcolor.sty</b>	<b>447</b>
158	Xcolor	447
<b>92</b>	<b>lwarp-xfrac.sty</b>	<b>450</b>
159	Xfrac	450
	<b>Change History and Index</b>	<b>453</b>

## List of Figures

1	tutorial.tex listing . . . . .	39
---	--------------------------------	----

## List of Tables

1	L <sup>A</sup> T <sub>E</sub> X–HTML generation — lwarp package — Supported functions	23
2	Additional supported packages . . . . .	26
3	Required software programs . . . . .	32
4	Files created in a new project . . . . .	41
5	Section depths and HTML headings . . . . .	74
6	Tabular baseline . . . . .	175
7	Tabular HTML column conversions . . . . .	176
8	Cross-referencing data structures . . . . .	194
9	Float data structures . . . . .	203

## 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 [453](#).

### v0.27:

- Improved documentation for MacOS install.
- Fix for `microtype` with `XYLATEX` and `LuaATEX`.
- 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.
- `LATEX` 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 `XYLATEX` and `LuaATEX` 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.
- `\HTMLFileName` may now be empty, allowing filenames without a prefix. `Lwarp` no longer automatically appends a `-` character. For existing projects, add a `-` to the end of `\HTMLFileName`.





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

#### v0.19:



- `MATHJAX` now may be used to display math via the `mathjax` option. See sections 6.2 and 6.10.3. To use MathJax with a pre-existing project, copy or link the file `lwarp_mathjax.txt` to the project's directory.
- `\rule` added, supporting width, height, raise, `\textcolor`.
- `\LateximageFontSizeName` provides user-adjustable font size for math and 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 59.
- `\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 `XYLATEX` and `LuaATEX`. 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.

A special `lwarp-newproject` package is used to set up the auxiliary files necessary for the HTML conversion. Included is a parallel version of the user's source document, `<sourcename>-html.tex`, which selects HTML output and then inputs the user's own source. This process allows both the printed and HTML versions to co-exist side-by-side, each with their own auxiliary files.

When requesting packages during HTML conversion, `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 23.)

## 2.1 Supported packages and features

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

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

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

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

## lwarp Supported Functions — continued

Category	Status
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.
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. $\text{\LaTeX}$ 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> .
<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>multirow</code> :	Emulated.



## lwarp Supported Functions — continued

Category	Status
<code>longtable:</code>	Emulated. Converted to a tabular. Captions supported. Extra headings and <code>\kill</code> lines must be enclosed in <code>\warpprintonly{}</code>
<code>booktabs:</code>	Emulated. <code>\toprule</code> and <code>\bottomrule</code> form black rules, <code>\midrule</code> forms silver rules, as demonstrated on this table. <code>\cmidrule</code> , demonstrated at this line, does not use width or trim options.
<code>threeparttable:</code>	Emulated
<code>graphics, graphicx:</code>	Emulated. <code>\includegraphics</code> supports <code>width</code> , <code>height</code> , <code>origin</code> , <code>angle</code> , and <code>scale</code> tags, and adds <code>class</code> . References to PDF files are changed to SVG, other image types are accepted as well. <code>\rotatebox</code> and <code>\scalebox</code> are supported as well as HTML can handle.
<code>rotating:</code>	Emulated. All objects are displayed unrotated.
<code>Lists:</code>	Supported
<code>enumitem:</code>	Supported, although spacing is still controlled by CSS.
<code>Environments:</code>	Standard L <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, mdframed:</code>	Supported
<code>multicol:</code>	Emulated, with CSS3. Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported.
<code>siunitx:</code>	Supported except for <code>per-mode=fraction</code> .
<code>xfrac:</code>	Supported

## lwarp Supported Functions — continued

Category	Status
Direct formatting:	<code>\emph</code> , <code>\textsuperscript</code> , <code>\textbf</code> , etc are supported. <code>\bfseries</code> , etc. are not yet supported. <code>lettrine</code> , <code>ulem</code> , and <code>soul</code> are supported.
Ordinals:	<code>nth</code> , <code>fmtcount</code> , and <code>engord</code> are supported.
Text ligatures:	Ligatures for symbols are supported. Ligatures for f, q, t are intentionally turned off because many simpler browsers do not display them correctly. Modern full-featured browsers re-create these ligatures on-the-fly.
Horizontal space:	HTML output for thin-unbreakable, unbreakable, <code>\enskip</code> , <code>\quad</code> , <code>\qquad</code> , <code>\hspace</code> .
Rules:	<code>\rule</code> with width, height, raise, text color.
HTML reserved characters:	<code>\&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 are simply nullfied during HTML output. Others are not affected by the output mode and thus work as-is.

Table 2: Additional supported packages

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

---

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** **L<sup>A</sup>T<sub>E</sub>XML**: <http://dlmf.nist.gov/LaTeXML/>

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

Prog **LaTeX2HTML** **L<sup>A</sup>T<sub>E</sub>X2HTML**: <http://www.latex2html.org/>  
and <http://ctan.org/pkg/latex2html>.

Prog **TeX2page** **T<sub>E</sub>X2page**: <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.
5. Renew the cache:

```
Enter ⇒ mktexlsr — or —
```

```
Enter ⇒ texhash
```

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 lwarp-newproject package:** Accompanies lwarp, and used to create configuration files.

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

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

**xindy:** The xindy package is used by lwarp to create indexes. On a MiK<sub>T</sub>E<sub>X</sub> system this may have to be acquired separately, but it is part of the regular installer as of mid 2015.

**latexmk:** Optionally used by lwarpmk to compile L<sup>A</sup>T<sub>E</sub>X code. On a MiK<sub>T</sub>E<sub>X</sub> system, Perl may need to be installed first.

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

**Poppler PDF utilities:**

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

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

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

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

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

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

For Windows, see:

<https://sourceforge.net/projects/poppler-win32/> and:  
<http://blog.alivate.com.au/poppler-windows/>

**Perl:**

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

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

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

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

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

---



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 T<sub>E</sub>X 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 ⇒ `pdflatex lwarp.dtx` (several times)

6. Create the .sty files:

Enter ⇒ `pdflatex lwarp.ins`

7. Copy the .sty files somewhere such as the T<sub>E</sub>X Live local tree found in the previous CTAN TDS section, under the subdirectory:

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

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

`<texlocal>/doc/local/lwarp`

9. Renew the cache:

Enter ⇒ `mktextlsr` — or —

Enter ⇒ `texhash`

Or, for Windows 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.

**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.

Just testing!

## 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 TeX Live or MiKTeX system you may use

```
Enter ⇒ kpsewhich lwarpmk.lua
```

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

4. Create `lwarpmk`:

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

- (a) Locate the TeX Live binaries:

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

This will be something like:

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

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

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

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

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

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

- (c) Make the link executable:

```
Enter ⇒ chmod 0755 lwarpmk
```

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

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

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

- (a) Locate the 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-newproject` package:  
`\usepackage[lwarpmk]{lwarp-newproject}`
2. When the tutorial is compiled with `pdflatex`, the file `lwarpmk.lua` will be generated along with the other configuration files.
3. `lwarpmk.lua` may be used for this project:

**Unix:**

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

**Windows:**

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

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

Or:

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

### 4.3 Installing additional utilities

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

Enter the following in a command line. If each programs' version is displayed, then that utility is already installed. See table 3 on page 32.

Enter ⇒ `luatex -version`

Enter ⇒ `xindy -version`

Enter ⇒ `latexmk -version`

Enter ⇒ `perl -version`

Enter ⇒ `pdfcrop -version`

Enter ⇒ `pdftotext -v`

Enter ⇒ `pdfseparate -version`

Enter ⇒ `pdftocairo -v`

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

The 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/pdftocrop>

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 32.

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 ⇒ 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 ⇒ pdflatex tutorial.tex (several times)
```

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

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

(Two of the new files are configuration files for the helper program `lwarpmk`. Whenever a print version of the document is created while the `lwarp-newproject` package is used, the configuration files for `lwarpmk` are updated to record the operating system,  $\text{\LaTeX}$  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

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

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

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

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

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

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

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

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

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

\section{Some math}

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

\printindex

\end{document}
```



Table 4: Files created in a new project

- tutorial.pdf:** The PDF output from L<sup>A</sup>T<sub>E</sub>X.
- 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 `tutorial.aux`, `tutorial.log`. When an HTML version of the document is created, `_html` versions of the auxiliary files will also be generated.
- lwarpmk.conf:** A configuration file for `lwarpmk`, which is used to automate the compilation of PDF or HTML versions of the document.
- tutorial.lwarpmkconf:** Another configuration file used by `lwarpmk`, which is only useful if you wish to have several projects residing in the same directory.
- .css files:** `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\_html.xdy:** Used by `lwarp` while creating an index. This file should not need to be modified by the user.
- lwarp\_mathjax.txt:** Inserted into the HTML files when `MathJax` is used to display math. This file should not need to be modified by the user.
- comment.cut:** A temporary file used by `lwarp` to conditionally process blocks of text. This file may be ignored.

---

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

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

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

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

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

## 5.2 Compiling the print version with `lwarpmk`

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

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

1. Re-compile the print version:

Enter  $\Rightarrow$  `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.

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.

### 5.3 Compiling the HTML version with `lwarpmk`

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

1. Compile the HTML version:

```
Enter ⇒ lwarpmk html
```

- (a) `lwarpmk` uses  $\text{\LaTeX}$  to process `tutorial_html.tex` to create `tutorial_html.pdf`.
- (b) `pdftotext` is then used to convert into the file `tutorial_html.html`. This file is a plain-text file containing HTML tags and content for the entire document.
- (c) `lwarpmk` manually splits `tutorial_html.html` into individual HTML files according to the HTML settings. For this tutorial, the result is `tutorial.html` (the home page), along with `First-chapter.html`<sup>4</sup>, `Some-math.html`, and the document's index in `_Index.html`.<sup>5</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 not yet been generated. Math may be displayed as SVG images or by a MathJax script, as seen in sections 5.4 and 5.5.

3. Force a recompile:

```
Enter ⇒ lwarpmk again
```

```
Enter ⇒ lwarpmk html
```

```
Enter ⇒ lwarpmk print
```

4. Process the HTML index and recompile:

```
Enter ⇒ lwarpmk htmlindex
```

```
Enter ⇒ lwarpmk html
```

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

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

<sup>4</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>5</sup>`index.html` is commonly used as a homepage, so the document index is in `_Index.html`.

## 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 ⇒ lwarpmk limages
```

```
Enter ⇒ lwarpmk html
```

2. Move to the tutorial's math page and reload.
3. The math images are displayed using the same font and formatting as the printed version.
4. Copy/paste a math expression into a text editor to see the  $\text{\LaTeX}$  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, change  
`\usepackage{lwarp}`  
to  
`\usepackage[mathjax]{lwarp}`
2. Recompile  
  
Enter  $\Rightarrow$  `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

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

The styles provided by `lwarp` include:


**lwarp.css:** A default style if **\NewCSS** is not used. This style is comparable to a plain L<sup>A</sup>T<sub>E</sub>X document. To set this style, you may use **\NewCSS{lwarp.css}**, or no **\NewCSS** 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 **\NewCSS** entry in the tutorial `lwarpmk.html`, and then reload the webpage.

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

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

`<project>.css` has an entry which loads `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 use `lwarpmk clearall` to clear all the HTML, PDF, and auxiliary files. Also, if `\HomeHTMLFilename`, `\HTMLFileName`, or `\UseLatexmk` are changed, it is necessary to reenale the `lwarp-newproject` package and then recompile the print version in order to recreate the configuration files for `lwarpmk`.

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

<code>\HomeHTMLFilename</code>	<b>\HomeHTMLFilename:</b> Filename of the homepage, without the “.html” suffix. Defaults to the <code>\BaseJobname</code> . A common setting is: <code>\newcommand{HomeHTMLFilename}{index}</code> causing the homepage to be the file <code>index.html</code> . Underscores are allowed in <code>\HomeHTMLFilename</code> and <code>\HTMLFileName</code> , but usually must be escaped elsewhere, such as <code>\item [\href{file\_name.pdf}{text}] \</code>
filename underscore	
<code>\HTMLFileName</code>	<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.
<code>\UseLatexmk</code>	<b>\UseLatexmk:</b> A macro which controls whether <code>lwarp</code> uses <code>latexmk</code> to compile the document. This setting is written to <code>lwarpmk</code> ’s configuration files. Defaults to false. Set to true with: <code>\newcommand*{\UseLatexmk}{true}</code>

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

Ctr <code>tocdepth</code>	<b>tocdepth:</b> Sectioning depth of the table of contents. See section 10 for a list of L <sup>A</sup> T <sub>E</sub> X stack depths.
Ctr <code>SideTOCDepth</code>	<b>SideTOCDepth:</b> Sectioning depth of the sideTOC. Defaults to 1, causing the sidetoc to show sections but not subsections.
sideTOC	Each subpage of the website has its own small table of contents on the side (the “sideTOC”). Its depth is set by <code>SideTOCDepth</code> . This sideTOC is only shown if the web page is wide enough. When using a narrow web browser window, “responsive web design” is used to show the sideTOC at the top of the page and a link back to “Home” at the bottom. It is recommended to set: <code>SideTOCDepth = FileDepth</code> , or <code>SideTOCDepth = FileDepth+1</code> If <code>SideTOCDepth &lt; FileDepth</code> , web pages will be inaccessible via the sideTOC.
	
Ctr <code>FileDepth</code>	<b>FileDepth:</b> 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` is prefixed.

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.)

---

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

---

#### Named HTML sections, no prefix:

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

---

```
\newcommand{\HomeHTMLFileName}{index}
\newcommand{\HTMLFileName}{}
```

---



---

```
\usepackage{lwrap}
\booltrue{FileSectionNames}
```

---

### Named HTML sections, with prefix:

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

---

```
\newcommand{\HomeHTMLFileName}{mywebsite}
\newcommand{\HTMLFileName}{mywebsite-}
\usepackage{lwrap}
\booltrue{FileSectionNames}
```

---

<code>\abstractname</code>	<b>\abstractname:</b> The name of the abstract. This may also be over-written by the babel package. Defaults to “Abstract”.
<code>\MetaLanguage</code>	<b>\MetaLanguage:</b> The HTML language meta header. Defaults to en-US.
<code>\SetFirstPageTop</code>	<b>\SetFirstPageTop:</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>\SetPageTop</code>	<b>\SetPageTop:</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>\SetPageBottom</code>	<b>\SetPageBottom:</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 section break:

<code>\tableofcontents</code>	<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 before any sectioning command which causes a file break:

<code>\NewCSS</code>	<b>\NewCSS:</b> <code>{\langle filename.css \rangle}</code> 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 <code>\NewCSS</code> command. If <code>\NewCSS</code> is not used, a default plain style is used to mimic printed $\text{\LaTeX}$ output. <code>lwrap_sagebrush.css</code> is a semi-fancy colored style as shown in this tutorial. Change it to <code>lwrap_formal.css</code> for a more formal look, or comment out the <code>\NewCSS</code> command to see the default. <code>\NewCSS</code> may be used before each file break to set the CSS for individual pages of the website.
----------------------	---

---

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

**Placed in the document wherever necessary:**

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

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

`\warpprintonly`      **\warpprintonly:** `{\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, at "HTML FILENAME AND LATEXMK SETTINGS", insert the line:

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

2. Directly recompile the printed version of the document.

⚠ NOT `lwarpmk`!

```
Enter ⇒ pdflatex tutorial.tex
```

```
(Or xelatex or lualatex)
```

`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 LaTeX.

1. Remove the auxiliary files for the project:

```
Enter ⇒ lwarpmk cleanall
```

2. Be sure that

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

is still enabled in `tutorial.tex`.

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

```
Enter ⇒ xelatex tutorial.tex -or-
```

```
Enter ⇒ lualatex tutorial.tex
```

When the recompile occurs and `lwarp-newproject` is loaded, the configuration files for `lwarpmk` are modified to remember which TeX engine was used. X<sub>Y</sub>LaTeX or LuaLaTeX will be used for future runs of `lwarpmk`.

4. To recompile the document:

```
Enter ⇒ lwarpmk print -and-
```

```
Enter ⇒ lwarpmk html
```

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

## 5.10 Cleaning auxiliary files

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

```
Enter ⇒ lwarpmk clean
```

## 5.11 Cleaning auxiliary and output files

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

```
Enter ⇒ lwarpmk cleanall
```

## 5.12 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
```

## 5.13 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



While using `pdflatex`, if no font-related package is specified, the default bit-mapped Computer Modern font is used, so simply add

```
usepackage{lmodern}
```

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

```
\usepackage{dejavu}
```

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



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	<code>cmap</code>	– <code>cmap</code>
Pkg	<code>textcomp</code>	– <code>textcomp</code>
Pkg	<code>microtype</code> <code>ligatures</code>	– <code>microtype</code> is automatically used by <code>lwarp</code> 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 $\text{\TeX}$ 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.

## 6.2 lwarp package loading and options

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

Pkg	<code>lwarp</code>	Load the <code>lwarp</code> package immediately after the font and UTF-8 setup commands.
Opt lwarp	<code>warpprint</code>	Select the <code>warpprint</code> option to generate print output (default), or the <code>warpHTML</code> option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual <code>pdflatex</code> , etc. When <code>lwarp-newproject</code> is loaded in print mode, it creates <code>&lt;project&gt;_html.tex</code> , which sets the <code>warpHTML</code> 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 lwarp	<code>warpHTML</code>	
Opt lwarp	<code>mathsvg</code>	For math display, select <code>mathsvg</code> (default), or <code>mathjax</code> . For more information about the math options, see section 6.10.3.
Opt lwarp	<code>mathjax</code>	

## 6.3 Selecting the operating system

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

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

## 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  $\text{\LaTeX}$  print-formatted PDF generation, or to HTML generation.

For most of built-in  $\text{\LaTeX}$  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 8: [Troubleshooting](#).

## 6.6 Commands for a successful HTML conversion

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

**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  $x^x$

**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 8: [Troubleshooting](#).

**6.7 Title page**

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

---

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

---

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

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

`\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

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

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

## 6.9 CSS

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

File `project.css`

File `sample_project.css`

---

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

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

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

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

---

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

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

## 6.10 Special cases and limitations

### 6.10.1 Text formatting

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

### 6.10.2 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.

### 6.10.3 Math

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

**SVG math option** For SVG math, math is rendered as usual by  $\text{\LaTeX}$  into the initial PDF file using the current font<sup>6</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}
```

**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.

**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

---

<sup>6</sup>See section 159 regarding fonts and fractions.

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

For an example of using custom  $\text{\LaTeX}$  macros with MathJax, see page 219.

### 6.10.4 Ntheorem

#### ⚠ Font control

This conversion is not total. Font control is via CSS, and the custom  $\text{\LaTeX}$  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.


### 6.10.5 Graphics

`\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  $\text{\LaTeX}$ , so expect some ugly results for scaling and rotating.

### 6.10.6 Xcolor

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

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

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

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

### 6.10.7 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.
- \newcolumnntype is ignored; unknown column types are set to l.
- tabularx ignores the width, but X columns do produce paragraph columns or multicolumns.
- Multirow and multicolumn cannot be used at the same time. (No rectangular holes wider than one column or taller than one row.)
- For multirow, insert \mrowcell into any empty multi-row cells. This will be a null function for the print output, and is a placeholder for parsing the table for HTML output.
- If a multirow reaches to the bottom of a table, and \bottomrule does not go over to that edge, try adding a line of empty cells below the \bottomrule. This may be a browser bug.
- If a \midrule is desired after the last row, an additional row of blank cells must be used.
- Multiple paragraphs in one cell of a p, b, m column must have \newline between paragraphs.
- \cmidrule does not support width or trim options due to CSS limitations.
- For longtable, place headings and footings which do not apply to HTML inside \warpprintonly{ }.
- For \toprule and \bottomrule, when combined with a warpprint or warpHTML environment, if a “misplaced \noalign” error occurs, change  
This & That \endhead  
to  
\warpprintonly{This & That \endhead}  
and likewise with the other \end headings. Keep the \endfirsthead row unchanged, as it is still relevant to HTML output.
- For S columns (from the siunitx package), while producing print output, anything non-numeric must be placed inside { } braces, including commands such as \multirow. While producing HTML output, though, anything placed inside braces is not seen by lwarp’s tabular handling algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:  
\warpprintonly{1 & 2 & {\multirow{2}{2cm}{Text}} & 3 \\}  
\warpHTMLonly{1 & 2 & \multirow{2}{2cm}{Text} & 3 \\}

⚠ \multirow &  
\multicolumn

⚠ \multirow

\multirow with rules

rule at last row

⚠ paragraphs

\cmidrule width, trim  
longtable headings

⚠ \warpprintonly

⚠ S columns




### 6.10.8 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.

### 6.10.9 Minipages

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

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


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

**in a span** There is limited support for minipages inside an HTML `<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.

### 6.10.10 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`.

For title font, use

```
frametitlefont=\textbf,
```

**font** 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.

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

 **package conflicts** If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 153.1.


#### 6.10.12 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`.


#### 6.10.13 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.

#### 6.10.14 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.

#### 6.10.15 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.

#### 6.10.16 cleveref and varioref packages

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

#### 6.10.17 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>	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>\TMLleftmargini</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>\TMLleftmargini</code> is if there is a wide <code>\flagverse</code> in use, such as the word “Chorus”, in which case the value of <code>\TMLleftmargini</code> should be set to a wide enough length to contain “Chorus”. The default is wide enough for a stanza number.
---	--

Horizontal spacing relies on `pdftotext`’s ability to discern the `-layout` 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.

#### 6.10.18 siunitx package

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

#### 6.10.19 newclue package

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

Ex:

---

```

\documentclass{article}
... <font setup>
\usepackage{newclue}
\usepackage[warpedHTML]{lwarp}
...

```

---

#### 6.10.20 newtxmath package

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

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

5. `\usepackage{newtxmath}`

6. ...

### 6.10.21 babel package

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

△ **French** 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}{:~}`

### 6.10.22 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.

### 6.10.23 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 Expanding the code base

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 60 for an example of the `picture` environment.

To create a custom HTML block or inline CSS class, see section 29.7.

## 7.1 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 `TEX` 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 130 (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.

## 7.2 Testing lwarp

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

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

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

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

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

## 8 Troubleshooting

### 8.1 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 6.10: [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.

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

Use `XYLaTeX` or `LuaLaTeX`.

**Floats out of sequence:**

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

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

**Print document contains HTML tags:**

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

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

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

**Images are appearing in strange places:**

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

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

**Plain-looking document:**

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

**Broken fragments of HTML:**

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

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

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

**Un-matched conditional compiles:**

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



### 8.1.1 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.

## 8.2 Compiling the `lwarp.dtx` file

**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}`

## 9 Implementation

This package is perhaps best described as a large collection of smaller individual technical challenges, in many cases solved through a number of ~~crude 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.

Table 5: Section depths and HTML headings

Section	L <sup>A</sup> T <sub>E</sub> X 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

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.

## 10 Stack depths

Stacks are created to track depth inside the L<sup>A</sup>T<sub>E</sub>X document structure. This depth is translated to HTML headings as shown in table 5. “Depth” here is not depth in the traditional computer-science stack-usage sense, but rather a representation of the nesting depth inside the L<sup>A</sup>T<sub>E</sub>X document structure.

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

Support for the `memoir` package will require the addition of a `book` level, which will 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.

## 11 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.

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

```

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

```

## 13 Unicode Input Characters

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

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

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

```

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

```

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

```

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

```

## 14 Handling package options

Pkg **etoolbox** Provides `\ifbool` and other functions.

```

22 \RequirePackage{etoolbox}[2011/01/03]
23 % requires v2.6 for \BeforeBeginEnvironment, etc.

```

Pkg	<b>ifplatform</b>	Provides <code>\ifwindows</code> to try to automatically detect Windows OS.
		24 <code>\RequirePackage{ifplatform}% sense op-system platform</code>
Pkg	<b>comment</b>	Provides conditional code blocks.
		25 <code>\RequirePackage{comment}</code>
		26 <code>\excludecomment{testing}</code>
Pkg	<b>kvoptions</b>	Allows key/value package options.
		27 <code>\RequirePackage{kvoptions}</code>
		28 <code>\SetupKeyvalOptions{family=LWR,prefix=LWR@}</code>
Bool	<b>warpingprint</b>	Set to true/false depending on the package option selections for print/HTML output
Bool	<b>warpingHTML</b>	and mathsvg/mathjax:
Bool	<b>mathjax</b>	29 <code>\newbool{warpingprint}</code>
		30 <code>\newbool{warpingHTML}</code>
		31 <code>\newbool{mathjax}</code>
	<b>\warpprintonly</b>	<code>{\langle contents \rangle}</code>
		Only process the contents if producing printed output.
		32 <code>\newcommand{\warpprintonly}[1]{\ifbool{warpingprint}{#1}{}}</code>
	<b>\warpHTMLonly</b>	<code>{\langle contents \rangle}</code>
		Only process the contents if producing HTML output.
		33 <code>\newcommand{\warpHTMLonly}[1]{\ifbool{warpingHTML}{#1}{}}</code>
Env	<b>warpall</b>	Anything in the <code>warpall</code> environment will be generated for print or HTML outputs.
		34 <code>\includecomment{warpall}</code>
Env	<b>warpprint</b>	Anything in the <code>warpprint</code> environment will be generated for print output only.
Opt lwarp	<b>warpprint</b>	If the <code>warpprint</code> option is given, boolean <code>warpingprint</code> is true and boolean <code>warpingHTML</code> is false, and may be used for <code>\ifbool</code> tests.
		35 <code>\DeclareVoidOption{warpprint}{%</code>
		36 <code>\PackageInfo{lwarp}{Using option 'warpprint'}</code>
		37 <code>\includecomment{warpprint}%</code>
		38 <code>\excludecomment{warpHTML}%</code>

```

39 \booltrue{warpingprint}%
40 \boolfalse{warpingHTML}%
41 }

```

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

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

```

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

```

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

```

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

```

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

```

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

```

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

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

```

57 \DeclareStringOption[\jobname]{BaseJobname}

```

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

```

58 \includecomment{warpprint}%
59 \excludecomment{warpHTML}%
60 \booltrue{warpingprint}%
61 \boolfalse{warpingHTML}%
62 \boolfalse{mathjax}%

```

```

63
64
65 % \ExecuteOptions{warpprint,mathsvg}
66
67 \ProcessKeyvalOptions*\relax

```

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

```

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

```

```

\@nameedef {\langle name \rangle} {\langle defn \rangle}

```

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

```

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

```

## 15 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-⟨package name⟩.sty`

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

```

\LWR@loadafter {\langle package name \rangle} Error if this package was loaded before lwarp.

```

```

70 \newcommand*\LWR@loadafter[1]{%
71 \ifpackageloaded{#1}
72 {
73 \PackageError{lwarp}
74 {Package #1, or one which uses #1, must be loaded after lwarp}
75 {Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{lwarp}.
76 Package #1 may also be loaded by something else, which must also be moved
77 after lwarp.}
78 }
79 {}
80 }

```

```

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

```

```

81 \newcommand*\LWR@loadbefore[1]{%
82 \PackageError{lwarp}

```



```

83 {Package #1 must be loaded before lwarp}
84 {Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{lwarp}.}
85 }

```

`\LWR@loadnever`  $\{\langle badpackagename \rangle\} \{\langle replacementpkgnam \rangle\}$

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

```

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

```

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`:

```

92 \LWR@loadafter{abstract}
93 \LWR@loadafter{afterpage}
94 \LWR@loadafter{algorithmicx}
95 \LWR@loadafter{alltt}
96 \LWR@loadafter{amsthm}
97 \LWR@loadafter{bookmark}
98 \LWR@loadafter{booktabs}
99 \LWR@loadafter{ccaption}
100 \LWR@loadafter{changepage}
101 \LWR@loadafter{cutwin}
102 \LWR@loadafter{dcolumn}
103 \LWR@loadafter{draftwatermark}
104 \LWR@loadafter{ellipsis}
105 \LWR@loadafter{emptypage}
106 \LWR@loadafter{enumerate}
107 \LWR@loadafter{epigraph}
108 \LWR@loadafter{eso-pic}
109 \LWR@loadafter{everypage}
110 \LWR@loadafter{extramarks}
111 \LWR@loadafter{fancyhdr}
112 \LWR@loadafter{floatrow}
113 \LWR@loadafter{float}
114 \LWR@loadafter{floatflt}
115 \LWR@loadafter{ftnright}
116 \LWR@loadafter{geometry}
117 % \LWR@loadafter{graphics}% pre-loaded by xunicode
118 % \LWR@loadafter{graphicx}% pre-loaded by xunicode
119 \LWR@loadafter{hyperref}
120 \LWR@loadafter{indentfirst}

```

```
121 \LWR@loadafter{keyfloat}
122 \LWR@loadafter{layout}
123 \LWR@loadafter{letterspace}
124 \LWR@loadafter{lettrine}
125 \LWR@loadafter{lips}
126 \LWR@loadafter{listings}
127 \LWR@loadafter{longtable}
128 \LWR@loadafter{lscape}
129 \LWR@loadafter{ltcaption}
130 \LWR@loadafter{marginfix}
131 \LWR@loadafter{marginnote}
132 \LWR@loadafter{mcaption}
133 \LWR@loadafter{mdframed}
134 \LWR@loadafter{microtype}
135 \LWR@loadafter{mparhack}
136 %\LWR@loadafter{multicol}% loaded by ltxdoc
137 \LWR@loadafter{multirow}
138 \LWR@loadafter{nameref}
139 \LWR@loadafter{needspace}
140 \LWR@loadafter{newtxmath}
141 \LWR@loadafter{nextpage}
142 \LWR@loadafter{nowidow}
143 \LWR@loadafter{ntheorem}
144 \LWR@loadafter{pagenote}
145 \LWR@loadafter{parskip}
146 \LWR@loadafter{placeins}
147 \LWR@loadafter{ragged2e}
148 \LWR@loadafter{rotating}
149 \LWR@loadafter{setspace}
150 \LWR@loadafter{showidx}
151 \LWR@loadafter{showkeys}
152 \LWR@loadafter{sidecap}
153 \LWR@loadafter{sidenotes}
154 \LWR@loadafter{soul}
155 \LWR@loadafter{subfig}
156 \LWR@loadafter{tabularx}
157 \LWR@loadafter{textpos}
158 \LWR@loadafter{theorem}
159 \LWR@loadafter{threeparttable}
160 \LWR@loadafter{tikz}
161 \LWR@loadafter{titleps}
162 \LWR@loadafter{titlesec}
163 \LWR@loadafter{titletoc}
164 \LWR@loadafter{tocloft}
165 \LWR@loadafter{trivfloat}
166 \LWR@loadafter{ulem}
167 \LWR@loadafter{varioref}
168 \LWR@loadafter{verse}
169 \LWR@loadafter{wallpaper}
170 \LWR@loadafter{wrapfig}
```

```

171 \LWR@loadafter{xcolor}
172 \LWR@loadafter{xfrac}

```

## 16 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:** 173 `\begin{warpall}`

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

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

```

174 \RequirePackage{iftex}
175 \newif\ifxetexorluatex
176 \ifXeTeX
177     \xetexorluatextrue
178 \else
179     \ifLuaTeX
180         \xetexorluatextrue
181     \else
182         \xetexorluatexfalse
183     \fi
184 \fi

185 \end{warpall}

```

**for HTML output:** 186 `\begin{warpHTML}`

```

187 \ifxetexorluatex
188 % ^^A \usepackage[no-math]{fontspec}

```

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

```

189 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
190 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
191 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
192 \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.

```

193 \RequirePackage {microtype}
194
195 \microtypesetup{
196 protrusion=false,
197 expansion=false,
198 tracking=false,
199 kerning=false,
200 spacing=false}
201
202 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}

203 \fi

204 \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.
- Will also use a scriptsize font.
- Also uses extra space at the margin to avoid HTML tag overflow off the page.
- Will also force a new page before environments.

**for HTML output:**

```

205 \begin{warpHTML}
206 \RequirePackage[paperheight=190in,paperwidth=20in,%
207 left=2in,right=12in,%
208 top=1in,bottom=1in,%
209 ]{geometry}
210 \end{warpHTML}

```

**for HTML & PRINT:**

```

211 \begin{warpall}

```

Pkg **xparse**

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

```

212 \RequirePackage{xparse}

213 \end{warpall}

```

```
for HTML output: 214 \begin{warpHTML}

    Pkg  expl3

    LATEX3 programming

    215 \RequirePackage{expl3}

Pkg  gettitlestring

    Used to emulate \nameref.

    216 \RequirePackage{gettitlestring}

Pkg  everyhook

    everyhook is used to patch paragraph handling.

    217 \RequirePackage{everyhook}
    218 \end{warpHTML}

for HTML & PRINT: 219 \begin{warpall}

    Pkg  fancyvrb

    Used for Verbatim, verse.

    220 \RequirePackage{fancyvrb}

    221 \end{warpall}

for HTML output: 222 \begin{warpHTML}

    Pkg  xifthen

    223 \RequirePackage{xifthen}

Pkg  xstring

    224 \RequirePackage{xstring}

Pkg  makeidx

    225 \RequirePackage{makeidx}
    226 \makeindex

Pkg  calc

    227 \RequirePackage{calc}
```

Pkg **refcount**

```
228 \RequirePackage{refcount}
```

Pkg **newfloat**

```
229 \RequirePackage{newfloat}
```

Pkg **caption**

```
230 \RequirePackage{caption}
```

Pkg **enumitem**

enumitem is patched to support `\newlist` with HTML.

```
231 \RequirePackage{enumitem}
232 \setlist[itemize]{leftmargin=0em}
233 \setlist[enumerate]{leftmargin=0em}
234 \setlist[description]{leftmargin=0em}
```

```
235 \end{warpHTML}
```

for HTML & PRINT: 236 \begin{warpall}

Pkg **titling**

Used for `\maketitle` and the title page. See section 42.

```
237 \RequirePackage{titling}
```

```
238 \end{warpall}
```

for HTML output: 239 \begin{warpHTML}

Pkg **zref**

Used for cross-references.

```
240 \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.

```
241 \PassOptionsToPackage{leqno,fleqn}{amsmath}
242 \RequirePackage{amsmath}
```

Pkg **environ**

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

```
243 \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.

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

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

```
245 \let\LWR@origpagestyle\pagestyle
246 \renewcommand*{\pagestyle}[1]{}
```

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

```
247 \let\LWR@origthispagestyle\thispagestyle
248 \renewcommand*{\thispagestyle}[1]{}
```

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

```
249 \let\LWR@origpagenumbering\pagenumbering
250 \renewcommand*{\pagenumbering}[1]{}
```

Pkg **xfrac**

Patched for HTML use. See section 159.

```
251 \RequirePackage{xfrac}
```

Used to convert lengths for image width/height options.

```
252 \RequirePackage{printlen}
```

```
253 \end{warpHTML}
```

## 17 Loading Packages

for HTML output: 254 \begin{warpHTML}

Remember the original \RequirePackage:

```
255 \let\LWR@origRequirePackage\RequirePackage
```

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

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

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

Note that argument 4 is passed directly to \StrBetween.

```
257 \newcommand*\LWR@findword[3][,]{%
258   \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
259 }
```

\LWR@lookforpackagename [*index*] If this is a package name, re-direct it to the lwarp version by renaming it lwarp- followed by the original name.

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

Find the n'th package name from the list:

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

See if the package name was found:

```
262 \IfStrEq{\LWR@strresult}{}
263 {}% no filename
264 {}% yes filename
```

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

```
265 \IfFileExists{lwarp-\LWR@strresult.sty}
266 {% latex_html_ file found
267 \StrSubstitute
268 {\LWR@requirepackagenames}
269 {\LWR@strresult}
270 {lwarp-\LWR@strresult}\LWR@requirepackagenames]
271 }
272 {}% no latex_html_* file
273 }% yes filename
274 }
```



`\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.

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

Redirect up to nine names:

```
276 \renewcommand*{\LWR@requirepackagenames}{#2}
277 \LWR@lookforpackagename{1}
278 \LWR@lookforpackagename{2}
279 \LWR@lookforpackagename{3}
280 \LWR@lookforpackagename{4}
281 \LWR@lookforpackagename{5}
282 \LWR@lookforpackagename{6}
283 \LWR@lookforpackagename{7}
284 \LWR@lookforpackagename{8}
285 \LWR@lookforpackagename{9}
```

`\RequirePackage` depending on the options and version:

```
286 \IfValueTF{#1}
287 {% options given
288 \IfValueTF{#3}% version given?
289 {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}[#3]}
290 {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}
291 }
292 {% no options given
293 \IfValueTF{#3}% version given?
294 {\LWR@origRequirePackage{\LWR@requirepackagenames}[#3]}
295 {\LWR@origRequirePackage{\LWR@requirepackagenames}}
296 }
297 }
298 \let\usepackage\RequirePackage
```

`\LWR@ProvidesPackagePass` {*⟨pkgnamē⟩*} [*⟨version⟩*]

Uses the original package, including options.

```
299 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
300 \PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
301 \IfValueTF{#2}
302 {\ProvidesPackage{lwarp-#1}[#2]}
303 {\ProvidesPackage{lwarp-#1}}
304 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
305 \ProcessOptions\relax
306
307 \IfValueTF{#2}
```

```

308 {\LWR@origRequirePackage{#1}[#2]}
309 {\LWR@origRequirePackage{#1}}
310 }

```

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

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

```

311 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
312 \PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
313 \IfValueTF{#2}
314 {\ProvidesPackage{lwarp-#1}[#2]}
315 {\ProvidesPackage{lwarp-#1}}
316 \DeclareOption*{}
317 \ProcessOptions\relax
318 }

319 \end{warpHTML}

```

## 18 Copying a file

for HTML output: 320 \begin{warpHTML}

`\LWR@copyfile`  $\{\langle source filename \rangle\} \{\langle destination filename \rangle\}$

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

```

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

```

---

```
337 \end{warpHTML}
```

## 19 Debugging messages

Bool `LWR@tracinglwarp` True if tracing is turned on.

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

`\tracinglwarp` Turns on the debug tracing messages.

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

`\LWR@traceinfo`  $\{ \langle text \rangle \}$  If tracing is turned on, writes the text to the .log file.

```
340 \newcommand{\LWR@traceinfo}[1]{%
341 \ifbool{LWR@tracinglwarp}%
342 {%
343 \typeout{*** lwarp: #1}%
344 % \PackageInfo{lwarp}{#1 : }%
345 }%
346 {}%
347 }
```

## 20 Remembering original formatting macros

for HTML output: `348 \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.

```
349 \let\LWR@origtextit\textit
350 \let\LWR@origtextbf\textbf
351 \let\LWR@origtexttt\texttt
352 \let\LWR@origtextsc\textsc
353 \let\LWR@origtextsf\textsf
354 \let\LWR@origtextrm\textrm
355 \let\LWR@origbfseries\bfseries
356 \let\LWR@origrmfamily\rmfamily
357 \let\LWR@origttfamily\ttfamily
358 \let\LWR@orignormalfont\normalfont
359
```

---

```

360 \let\LWR@origraggedright\raggedright
361 \let\LWR@origonecolumn\onecolumn
362
363 \let\LWR@origtextsuperscript\textsuperscript
364 \let\LWR@origtextsubscript\textsubscript
365
366 \let\LWR@origscriptsize\scriptsize
367
368 \let\LWR@orignewpage\newpage
369
370 \let\LWR@origminipage\minipage
371 \let\LWR@origendminipage\endminipage
372
373 \let\LWR@orignewline\newline
374
375 \let\LWR@origitem\item
376
377 \let\LWR@origpar\par
378
379
380 \let\LWR@origfootnote\footnote
381 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
382
383 \let\LWR@origclearpage\clearpage
384 \let\clearpage\relax
385 \let\cleardoublepage\relax

386 \end{warpHTML}

```

## 21 Operating-System portability

lwarp tries to detect which operating system is being used.

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

for HTML & PRINT: 387 \begin{warppall}

### 21.1 Common portability code

Bool	usingOSwindows	Set if <code>\warpOSwindows</code> .
------	----------------	--------------------------------------

```

388 \newbool{usingOSwindows}
389 \boolfalse{usingOSwindows}

```

## 21.2 Unix, Linux, and Mac OS

`\OSPathSymbol` Symbol used to separate directories in a path.

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

## 21.3 MS-Windows

For MS-Windows:

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

```
391 \newcommand*{\warpOSwindows}
392 {
393 \booltrue{usingOSwindows}
394 \renewcommand*{\OSPathSymbol}{\@backslashchar}
395 }
```

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

```
396 \ifwindows
397 \warpOSwindows
398 \fi
```

```
399 \end{warpall}
```

## 22 Stacks

for HTML output: `400 \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@closedeptione`, etc.

## 22.1 Assigning depths

initial depths for empty stack entries:

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

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

```
402 \newcommand*\LWR@depthfinished}{-4}
403 \newcommand*\LWR@depthpart}{-1}
404 \newcommand*\LWR@depthchapter}{0}
405 \newcommand*\LWR@depthsection}{1}
406 \newcommand*\LWR@depthsubsection}{2}
407 \newcommand*\LWR@depthsubsubsection}{3}
408 \newcommand*\LWR@depthparagraph}{4}
409 \newcommand*\LWR@depthsubparagraph}{5}
```

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

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

used by `\item`:

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

## 22.2 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.

```
412 \newcommand*\LWR@closeone}{}% top of the stack
413 \newcommand*\LWR@closetwo){}
414 \newcommand*\LWR@closethree){}
415 \newcommand*\LWR@closefour){}
416 \newcommand*\LWR@closefive){}
417 \newcommand*\LWR@closesix){}
418 \newcommand*\LWR@closeseven){}
419 \newcommand*\LWR@closeeight){}
420 \newcommand*\LWR@closenine){}
421 \newcommand*\LWR@closeten){}
422 \newcommand*\LWR@closeeleven){}
423 \newcommand*\LWR@closetwelve){}
```

## 22.3 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}
```

---

```
424 \newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
425 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
426 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
427 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
428 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
429 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
430 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
431 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
432 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
433 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}
434 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
435 \newcommand*{\LWR@closedephtwelve}{\LWR@depthnone}
```

## 22.4 Pushing and popping the stack

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

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

```
436 \NewDocumentCommand{\pushclose}{m m}
437 {
438 \let\LWR@closetwelve\LWR@closeeleven
439 \let\LWR@closeeleven\LWR@closeten
440 \let\LWR@closeten\LWR@closenine
441 \let\LWR@closenine\LWR@closeeight
442 \let\LWR@closeeight\LWR@closeseven
443 \let\LWR@closeseven\LWR@closesix
444 \let\LWR@closesix\LWR@closefive
445 \let\LWR@closefive\LWR@closefour
446 \let\LWR@closefour\LWR@closethree
```

```

447 \let\LWR@closethree\LWR@closetwo
448 \let\LWR@closetwo\LWR@closeone
449 \let\LWR@closeone#1
450 \let\LWR@closedepthtwelve\LWR@closedeptheleven
451 \let\LWR@closedeptheleven\LWR@closedepthten
452 \let\LWR@closedepthten\LWR@closedepthnine
453 \let\LWR@closedepthnine\LWR@closedeptheight
454 \let\LWR@closedeptheight\LWR@closedepthseven
455 \let\LWR@closedepthseven\LWR@closedepthsix
456 \let\LWR@closedepthsix\LWR@closedepthfive
457 \let\LWR@closedepthfive\LWR@closedepthfour
458 \let\LWR@closedepthfour\LWR@closedepththree
459 \let\LWR@closedepththree\LWR@closedepthtwo
460 \let\LWR@closedepthtwo\LWR@closedepthone
461 \let\LWR@closedepthone#2
462 }

```

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

```

463 \newcommand*{\popclose}
464 {
465 \let\LWR@closeone\LWR@closetwo
466 \let\LWR@closetwo\LWR@closethree
467 \let\LWR@closethree\LWR@closefour
468 \let\LWR@closefour\LWR@closefive
469 \let\LWR@closefive\LWR@closesix
470 \let\LWR@closesix\LWR@closeseven
471 \let\LWR@closeseven\LWR@closeeight
472 \let\LWR@closeeight\LWR@closenine
473 \let\LWR@closenine\LWR@closeten
474 \let\LWR@closeten\LWR@closeeleven
475 \let\LWR@closeeleven\LWR@closetwelve
476 \let\LWR@closedepthone\LWR@closedepthtwo
477 \let\LWR@closedepthtwo\LWR@closedepththree
478 \let\LWR@closedepththree\LWR@closedepthfour
479 \let\LWR@closedepthfour\LWR@closedepthfive
480 \let\LWR@closedepthfive\LWR@closedepthsix
481 \let\LWR@closedepthsix\LWR@closedepthseven
482 \let\LWR@closedepthseven\LWR@closedeptheight
483 \let\LWR@closedeptheight\LWR@closedepthnine
484 \let\LWR@closedepthnine\LWR@closedepthten
485 \let\LWR@closedepthten\LWR@closedeptheleven
486 \let\LWR@closedeptheleven\LWR@closedepthtwelve
487 }

488 \end{warpHTML}

```



## 23 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: 489 \begin{warpHTML}

```
\LWR@setexparray {\langle name \rangle} {\langle index \rangle} {\langle contents \rangle}
```

```
490 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
491 \expandafter\edef\csname #1#2\endcsname{\expandonce#3}%
492 }
```

```
\LWR@getexparray {\langle name \rangle} {\langle index \rangle}
```

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

```
494 \end{warpHTML}
```

## 24 HTML entities

for HTML output: 495 \begin{warpHTML}

HTML entites and HTML Unicode entities:

```
496 \let\LWR@origampersand\&
```

```
\HTMLentity {\langle entitytag \rangle}
```

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

```
\HTMLunicode {\langle hex_unicode \rangle}
```

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

```
\&
```

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

```

\textless
\textgreater
500 \let\LWR@origtextless\textless
501 \renewcommand*{\textless}{\HTMLentity{lt}}
502
503 \let\LWR@origtextgreater\textgreater
504 \renewcommand*{\textgreater}{\HTMLentity{gt}}

505 \end{warpHTML}

```

## 25 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:** 506 `\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.

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

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

```
508 \providecommand*{\HTMLFileName}{}
```

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

```
509 \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.

```

510 \newcommand*{\SetHTMLFileNumber}[1]{%
511 \setcounter{LWR@htmlfilenumber}{#1}%
512 }

```

Bool FileSectionNames Selects how to create HTML file names.

Defaults to use section names in the filenames.

```
513 \newbool{FileSectionNames}
514 \booltrue{FileSectionNames}

515 \end{warpall}
```

for HTML output: 516 \begin{warpHTML}

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

```
517 \newcounter{LWR@htmlfilenumber}
518 \setcounter{LWR@htmlfilenumber}{0}
```

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

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

```
519 \newcommand*{\LWR@htmlsectionfilename}[1]{%
520 \LWR@traceinfo{LWR@htmlsectionfilename A}%
```

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

```
521 \LWR@traceinfo{about to assign temp}%
522 \edef\LWR@tempone{#1}%
523 \LWR@traceinfo{about to compare with ??}%
524 \ifthenelse{\equal{\LWR@tempone}{??}}{%
525 {%
526 \LWR@traceinfo{found ??}%
527 }{%
528 \LWR@traceinfo{not found ??}%
529 }%
530 \LWR@traceinfo{about to compare with zero or empty}%
531 \ifthenelse{%
532 \equal{\LWR@tempone}{0}}{%
533 \OR \equal{\LWR@tempone}{}}%
534 \OR \equal{\LWR@tempone}{??}}%
535 }%
536 {%
537 \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFileName.html}%
538 \HomeHTMLFileName.html%
539 }%
```

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`:

```

540 {%
541 \LWR@traceinfo{\LWR@htmlsectionfilename C \LWR@tempone}%
542 \ifthenelse{%
543 \equal{\HTMLFileName}{ } \AND \equal{\LWR@tempone}{Index} \OR \equal{\LWR@tempone}{index}%
544 }%
545 {%
546 \LWR@traceinfo{prefixing the index name with an underscore.}%
547 \_#1.html}%

```

Otherwise, create a filename with the chosen prefix:

```

548 {\HTMLFileName#1.html}%
549 }%
550 \LWR@traceinfo{\LWR@htmlsectionfilename Z}%
551 }

```

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

Prints the filename for the given label

```

552 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
553 \LWR@traceinfo{\LWR@htmlrefsectionfilename A: '#1!}%
554 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
555 \LWR@traceinfo{\LWR@htmlrefsectionfilename B}%
556 }

557 \end{warpHTML}

```

## 26 Homepage link

**for HTML output:** 558 `\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.

```

559 \newcommand*{\LinkHome}{%
560 \LWR@subhyperrefclass{%
561 \HomeHTMLFileName.html}%
562 {Home}{linkhome}%
563 }

```

`\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.

```
564 \newcommand*{\LWR@topnavigation}{
565 \LWR@htmlclassline{nav}{topnavigation}{\LinkHome}
566 }
```

`\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.

```
567 \newcommand*{\LWR@botnavigation}{
568 \LWR@htmlclassline{nav}{botnavigation}{\LinkHome}
569 }
```

```
570 \end{warpHTML}
```

## 27 \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: 571 `\begin{warpHTML}`

`\PrintStack` Prints the closedepth stack.

```
572 \newcommand*{\PrintStack}{
573 \LWR@startpars
574 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
575 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
576 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
577 \LWR@closedephten{} \LWR@closedeptheleven{} \LWR@closedephtwelve{}
578 }
```

```
579 \end{warpHTML}
```

## 28 Closing stack levels

for HTML output: 580 `\begin{warpHTML}`

Close one nested level:

```

581 \newcommand*{\LWR@closeoneprevious}{%
582
583 \LWR@closeone{
584
585 \popclose{
586 }

```

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

```

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

```

Close any pending paragraph:

```

588 \LWR@stoppars

```

Close anything nested deeper than the desired depth:

```

589 \whiledo{\not\(\LWR@closedepthone<#1\)}{\LWR@closeoneprevious}
590 }

```

```

591 \end{warpHTML}

```

## 29 HTML tags, spans, divs, elements

for HTML output: 592 \begin{warpHTML}

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

```

593 \newcommand*{\LWR@tagpart}{h2}
594 \newcommand*{\LWR@tagpartend}{/h2}
595 \newcommand*{\LWR@tagchapter}{h3}
596 \newcommand*{\LWR@tagchapterend}{/h3}
597 \newcommand*{\LWR@tagsection}{h4}
598 \newcommand*{\LWR@tagsectionend}{/h4}
599 \newcommand*{\LWR@tagsubsection}{h5}
600 \newcommand*{\LWR@tagsubsectionend}{/h5}
601 \newcommand*{\LWR@tagsubsubsection}{h6}
602 \newcommand*{\LWR@tagsubsubsectionend}{/h6}
603 \newcommand*{\LWR@tagparagraph}{span class="paragraph">{}}
604 \newcommand*{\LWR@tagparagraphend}{/span}
605 \newcommand*{\LWR@tagsubparagraph}{span class="subparagraph">{}}
606 \newcommand*{\LWR@tagsubparagraphend}{/span}
607
608 \newcommand*{\LWR@tagregularparagraph}{p}

```


## 29.2 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.

```
609
610 \newcommand*\LWR@htmltagc[1]{%
611 {%
612 \protect\LWR@origttfamily%
613 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
614 }%
615 }
```


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.

```
616 \newenvironment*LWR@nestspan{
617 {%
618 \addtocounter{LWR@spandepth}{1}%
619 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{-}{-}%
620 }%
621 {\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.

```
622 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
623 \LWR@ensuredoingapar%
624 \LWR@htmltagc{#1}%
625 \begin{LWR@nestspan}%
626 #2%
627 \LWR@htmltagc{/#1}%
628 \end{LWR@nestspan}%
629 }
```

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

```
630 \NewDocumentCommand{\LWR@htmlspanclass}{m o +m}{%
631 \LWR@ensuredoingapar%
```

```

632 \LWR@subhtmllementclass{span}{#1}{#2}%
633 \begin{LWR@nestspan}%
634 #3%
635 \LWR@htmltagc{/span}%
636 \end{LWR@nestspan}%
637 }

```

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

print an HTML tag: `<tag>`

```

638 \newcommand*{\LWR@htmltagb}[1]{%
639 \LWR@htmltagc{#1}%
640 \endgroup%
641 }
642
643 \newcommand*{\LWR@htmltag}{%
644 \begingroup\catcode'\_ =12
645 \LWR@htmltagb%
646 }

```

### 29.3 Block tags and comments

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

```

\LWR@htmlopencomment
\LWR@htmlclosecomment
647 \newcommand*{\LWR@htmlopencomment}{%
648 % \LWR@origtexttt{\LWR@origtextbf{\LWR@origtextless{!}{-}{-}}}%
649 \begingroup\LWR@origttfamily\LWR@origtextless{!}{-}{-}\endgroup%
650 }
651
652 \newcommand*{\LWR@htmlclosecomment}{%
653 % \LWR@origtexttt{\LWR@origtextbf{{-}{-}\LWR@origtextgreater{}}}%
654 \begingroup\LWR@origttfamily{-}{-}\LWR@origtextgreater{}\endgroup%
655 }

```

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

```

656 \newcommand{\LWR@htmlcomment}[1]{%
657 \LWR@htmlopencomment}%
658 \LWR@origtextrm{#1}%
659 \LWR@htmlclosecomment}%

```



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

```

660 \newcommand{\LWR@htmlblockcommentb}[1]
661 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
662
663 \newcommand{\LWR@htmlblockcomment}
664 {%
665 \begingroup\catcode'\_ =12%
666 \LWR@htmlblockcommentb%
667 }
```

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

```

668 \newcommand*\LWR@htmlblocktag}[1]{%
669 \LWR@stoppars%
670 \LWR@htmltag{#1}%
671 \LWR@startpars%
672 }
```

`\LWR@htmlblocktagcomment`  $\{\langle tag \rangle\} \{\langle HTML comment \rangle\}$

```

673 \NewDocumentCommand{\LWR@htmlblocktagcomment}{m +m}{%
674 \LWR@stoppars%
675 \LWR@htmltag{#1}\LWR@htmlcomment{#2}\LWR@orignewline%
676 \LWR@startpars%
677 }
```

## 29.4 Div class and element class

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

Factored and reused in several places.

```

678 \NewDocumentCommand{\LWR@subhtmllementclass}{m m o}{%
679 \IfValueTF{#3}%
680 {% option
681 \ifthenelse{\equal{#3}{}}{%
682 {\LWR@htmltag{#1 class="#2"}}% empty option
683 {\LWR@htmltag{#1 class="#2" style="#3"}}% non-empty option
684 }% option
685 {\LWR@htmltag{#1 class="#2"}}% no option
686 }
```

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

```

687 \NewDocumentCommand{\LWR@htmlelementclass}{m m o}{%
688 \LWR@stoppars%
689 \LWR@subhtmlelementclass{#1}{#2}[#3]%
690 \LWR@startpars%
691 }

```

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

```

692 \newcommand*{\LWR@htmlelementclassend}[2]{%
693 \LWR@stoppars%
694 \LWR@htmltag{/#1}%
695 \LWR@htmlcomment{End of #1 ‘‘#2’’}%
696 \LWR@startpars%
697 }

```

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

```

698 \NewDocumentCommand{\LWR@htmldivclass}{m o}{%
699 \LWR@htmlelementclass{div}{#1}[#2]%
700 }

```

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

```

701 \newcommand*{\LWR@htmldivclassend}[1]{%
702 \LWR@htmlelementclassend{div}{#1}%
703 }

```

## 29.5 Single-line elements

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

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

```

704 \NewDocumentCommand{\LWR@htmlelementclassline}{m m o +m}{%
705 \LWR@stoppars
706 \LWR@subhtmlelementclass{#1}{#2}[#3]%
707 #4%
708 \LWR@htmltag{/#1}
709 \LWR@startpars
710 }

```

## 29.6 HTML5 semantic elements

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

```
711 \newcommand*\LWR@htmlelement}[1]{%
712 \LWR@htmlblocktag{#1}
713 }
```

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

```
714 \newcommand*\LWR@htmlelementend}[1]{%
715 \LWR@stoppars
716 \LWR@htmltag{/#1}
717 \LWR@startpars
718 }
719
720 \end{warpHTML}
```

## 29.7 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 [62](#).

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

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

**for HTML output:**

```
721 \begin{warpHTML}
722 \NewDocumentEnvironment{BlockClass}{m o}
723 {
724 \LWR@htmldivclass{#1}[#2]
725 }
726 {
727 \LWR@htmldivclassend{#1}
728 }
729 \end{warpHTML}
```

**for PRINT output:**

```
730 \begin{warpprint}
731 \NewDocumentEnvironment{BlockClass}{m o}{}{}
732 \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: 733 \begin{warpHTML}  
 734 \newcommand{\BlockClassSingle}[2]{%  
 735 \LWR@html@element@classline{div}{#1}{#2}%  
 736 }  
 737 \end{warpHTML}

for PRINT output: 738 \begin{warpprint}  
 739 \newcommand{\BlockClassSingle}[2]{#2}  
 740 \end{warpprint}

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

for HTML output: 741 \begin{warpHTML}  
 742 \NewDocumentCommand{\InlineClass}{m o +m}{%  
 743 \LWR@html@spanclass{#1}{#2}{#3}%  
 744 }  
 745 \end{warpHTML}

for PRINT output: 746 \begin{warpprint}  
 747 \NewDocumentCommand{\InlineClass}{m o +m}{#3}  
 748 \end{warpprint}

## 29.8 Closing HTML tags

for HTML output: 749 \begin{warpHTML}

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

```
750 \newcommand*{\LWR@printclosepart}
751   {\LWR@htmlcomment{Closing part}}
752 \newcommand*{\LWR@printclosechapter}
753   {\LWR@htmlcomment{Closing chapter}}
754 \newcommand*{\LWR@printclosesection}
755   {\LWR@htmlcomment{Closing section}}
756 \newcommand*{\LWR@printclosesubsection}
757   {\LWR@htmlcomment{Closing subsection}}
758 \newcommand*{\LWR@printclosesubsubsection}
759   {\LWR@htmlcomment{Closing subsubsection}}
760 \newcommand*{\LWR@printcloseparagraph}
761   {\LWR@htmlcomment{Closing paragraph}}
762 \newcommand*{\LWR@printclosesubparagraph}
763   {\LWR@htmlcomment{Closing subparagraph}}
```

Lists require closing HTML tags:

```

764 \newcommand*{\LWR@printcloselistitem}
765     {\LWR@htmltag{/li}}
766 \newcommand*{\LWR@printclosedescitem}
767     {\LWR@htmltag{/dd}}
768 \newcommand*{\LWR@printcloseitemize}
769     {\LWR@htmltag{/ul}}
770 \newcommand*{\LWR@printcloseenumerate}
771     {\LWR@htmltag{/ol}}
772 \newcommand*{\LWR@printclosedescription}
773     {\LWR@htmltag{/dl}}

774 \end{warpHTML}

```

### 30 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 31 has high-level commands which allow paragraph-tag generation to start/stop. Even when allowed (`\LWR@doingstartpars`), tags are not generated until a `LATEX` paragraph is being used (`\LWR@doingapar`). `LWR@lateximagedepth` is used to prevent nesting tags inside a `lateximage`. `LWR@spandepth` is used to prevent nesting paragraph tags inside a paragraph, which became important inside `\fbox` commands and other spans.

**for HTML output:** 775 `\begin{warpHTML}`

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

```

776 \newcounter{LWR@spandepth}
777 \setcounter{LWR@spandepth}{0}

```

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

```

778 \newbool{LWR@doingstartpars}
779 \boolfalse{LWR@doingstartpars}

```

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

```

780 \newbool{LWR@doingapar}
781 \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:

```

782 \newcommand*{\LWR@ensuredoingapar}{%
783 \ifbool{\LWR@doingstartpars}%
784 {\global\booltrue{\LWR@doingapar}}%
785 {}%
786 }

```

`\LWR@openparagraph`

```

787 \newcommand*{\LWR@openparagraph}
788 {%

```

See if paragraph handling is enabled:

```

789 \ifbool{\LWR@doingstartpars}%
790 {% handling pars

```

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

```

791 \ifthenelse{%
792 \cnttest{\value{\LWR@lateximagedepth}}{>}{0} \OR%
793 \cnttest{\value{\LWR@spandepth}}{>}{0}%
794 }{% 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>`.

```

795 {}% no nested par tags

```

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

```

796 {% yes nest par tags
797 \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:

```

798 \ifnumcomp{\@listdepth}{>}{0}%
799 {%

```

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

```
800 \LWR@orighspace{1in}%
801 }{}%
```

Now have started a paragraph.

```
802 \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.)

```
803 \let\par\LWR@closeparagraph%
804 }% end of yes nest par tags
805 }% end of handling pars
806 {}% not handling pars
807 }
```

\LWR@closeparagraph

```
808 \newcommand*{\LWR@closeparagraph}
809 {%
```

See if paragraph handling is enabled:

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

If currently in paragraph mode:

```
811 {% handling pars
```

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

```
812 \ifthenelse{%
813 \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
814 \cnttest{\value{LWR@spandepth}}{>}{0}%
815 }%
```

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

```
816 {% no nested par tags
817 \ifthenelse{%
818 \cnttest{\value{LWR@spandepth}}{>}{0}%
819 \AND%
820 \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
821 }%
822 {%
823 \ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}%
```

---

```

824 }%
825 {}%
826 }% no nested par tags

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

827 {% yes nest par tags

Print a closing tag:

828 \unskip%
829 \LWR@htmltagc{/\LWR@tagregularparagraph}%

No longer doing a paragraph:

830 \global\boolfalse{LWR@doingapar}%
831 % Disable the special \env{minipage} \& \cs{hspace} interaction
832 % until a new minipage is found:
833 % \begin{macrocode}
834 \global\boolfalse{LWR@minipagethispar}%
835 }% end of yes nest par tags
836 }% end of handling pars

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

837 {% not handling pars
838 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}{%
839 {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
840 {}%
841 }% not handling pars

Finish with regular paragraph processing

842 \LWR@origpar%
843 }

844 \end{warpHTML}

```

## 31 Paragraph start/stop handling

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

Section 30 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: 845 \begin{warpHTML}

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

```
846 \newcommand*{\LWR@startpars}%
847 {%
```

See if currently handling HTML paragraphs:

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

If already in paragraph mode, do nothing.

```
849 {}%
```

If not currently in paragraph mode:

```
850 {%
```

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

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

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

```
852 \let\par\LWR@closeparagraph
853
854 }% an intentionally blank line
```

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

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

No <par> tag yet to undo:

```
856 \global\boolfalse{\LWR@doingapar}%
857 }
```

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

```
858 \newcommand*{\LWR@stoppars}%
859 {%
```

See if currently handling HTML paragraphs:

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

if currently in an HTML paragraph:

```
861 {%
```

Print a closing tag:

```
862 \unskip%
```

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

No longer have an open HTML paragraph:

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

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

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

```
866
```

```
867 }% an intentionally blank line
```

If was not in an HTML paragraph:

```
868 {}%
```

See if currently allowing HTML paragraphs:

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

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

```
870 {%
```

```
871 \ClearPreHook{par}%
```

```
872 }%
```

Else: do nothing

```
873 {}%
```

no longer in paragraph mode

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

no `<p>` tag to undo:

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

```
876 }
```

```
877 \end{warpHTML}
```

## 32 Page headers and footers

for HTML & PRINT: 878 \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.

```
879 \newcommand{\LWR@firstpagetop}{ } % for the home page alone
880 \newcommand{\LWR@pagetop}{ } % for all other pages
881 \newcommand{\LWR@pagebottom}{ }
882
883 \newcommand{\LWR@setfirstpagetopb}[1]{%
884 \renewcommand{\LWR@firstpagetop}{#1}
885 \catcode'\_ =8
886 }
```

\SetFirstPageTop {*{text and logos}*}

```
887 \newcommand{\SetFirstPageTop}{%
888 \catcode'\_ =12
889 \LWR@setfirstpagetopb
890 }
```

```
891 \newcommand{\LWR@setpagetopb}[1]{%
892 \renewcommand{\LWR@pagetop}{#1}
893 \catcode'\_ =8
894 }
```

\SetPageTop {*{text and logos}*}

```
895 \newcommand{\SetPageTop}{%
896 \catcode'\_ =12
897 \LWR@setpagetopb
898 }
```

```
899 \newcommand{\LWR@setpagebottomb}[1]{%
900 \renewcommand{\LWR@pagebottom}{#1}
901 \catcode'\_ =8
902 }
```

\SetPageBottom {*{text and logos}*}

```
903 \newcommand{\SetPageBottom}{%
904 \catcode'\_ =12
905 \LWR@setpagebottomb
906 }
```

```
907 \end{warpall}
```

## 33 CSS

**for HTML output:** 908 \begin{warpHTML}

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

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

**\NewCSS**  $\{ \langle new-css-filename.css \rangle \}$

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

```
910 \newcommand*{\LWR@newcssb}[1]{%
911 \renewcommand*{\LWR@currentcss}{#1}
912 \catcode'\_ =8
913 }
914
915 \newcommand*{\NewCSS}{
916 \catcode'\_ =12
917 \LWR@newcssb
918 }
919 \end{warpHTML}
```

**for PRINT output:** 920 \begin{warpprint}  
 921 \newcommand\*{\NewCSS}[1]{}  
 922 \end{warpprint}

## 34 HTML meta description

**for HTML output:** 923 \begin{warpHTML}

**\LWR@currentHTMLdescription** The HTML meta description to use.

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

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

```
924 \newcommand{\LWR@currentHTMLdescription}{}
```

```
\NewHTMLdescription {⟨New HTML meta description.⟩}
```

Assigns the HTML file's description meta tag

```
925 \newcommand{\NewHTMLdescription}[1]{%
926 \renewcommand{\LWR@currentHTMLdescription}{#1}
927 }
928
929 \end{warpHTML}
```

```
for PRINT output: 930 \begin{warpprint}
931 \newcommand{\NewHTMLdescription}[1]{%
932 \end{warpprint}
```

## 35 Footnotes

`lwrap` uses native  $\text{\LaTeX}$  footnote code, although with its own `\box` to avoid the  $\text{\LaTeX}$  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.

### 35.1 Regular page footnotes

In HTML documents, footnotes are placed at the bottom of the web page using the  $\text{\LaTeX}$  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 [35.4](#) for the implementation.

### 35.2 Minipage footnotes

See section [61.2](#) for minipage footnotes.

### 35.3 Titlepage thanks

See section [42.6](#) for titlepage footnotes.

## 35.4 Regular page footnote implementation

for HTML output: 933 \begin{warpHTML}

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

934 \newbox\LWR@footnotes

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

935 \long\def\@makefntext#1{\textsuperscript{\@thefnmark} #1}

936

937 \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.

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

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

940 \protected@edef\@currentlabel{%

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

942 }% \@currentlabel

943 \color@begingroup%

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

945 \@makefntext{#1}%

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

947 \color@endgroup%

948 }% vbox

949 }%

950

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

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

953 \unvbox\@mpfootins

954 \reset@font\footnotesize

955 \hsize\columnwidth

956 \@parboxrestore

957 \protected@edef\@currentlabel

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

959 \color@begingroup

960 \@makefntext{%

961 \ignorespaces#1%

962 }%

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

963 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%

964 {}%

965 {\LWR@htmltagc{/LWR@tagregularparagraph}}%

```
966 \color@endgroup}%
967 }
```

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

```
968 \newcommand*{\LWR@printpendingfootnotes}{%
969 \ifvoid\LWR@footnotes\else
970 \begin{BlockClass}{footnotes}
971 \LWR@origmedskip
972 \unvbox\LWR@footnotes
973 \setbox\LWR@footnotes=\vbox{}
974 \end{BlockClass}
975 \fi
976 }

977 \end{warpHTML}
```

## 36 Marginpars

for HTML output: 978 \begin{warpHTML}

```
\marginpar [\langle left \rangle] {\langle right \rangle}
```

```
979 \renewcommand{\marginpar}[2] [] {%
980 \LWR@htmlspanclass{marginpar}{#2}%
981 }
```

```
982 \end{warpHTML}
```

## 37 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: 983 \begin{warppall}

Ctr `FileDepth` `{\langle section depth \rangle}` determines how deeply to break into new HTML files, similar to

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

```
984 \newcounter{FileDepth}
985 \setcounter{FileDepth}{-5}
```

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

```
986 \newbool{CombineHigherDepths}
987 \booltrue{CombineHigherDepths}
```

```
988 \end{warpall}
```

for HTML output: 989 `\begin{warpHTML}`

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

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

`\LWR@thisnewfilename` The filename being sanitized.

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

`\LWR@filenameno blanks`  $\{\langle filename \rangle\}$

Convert blanks into dashes, removes short words, store result in `\LWR@thisfilename`.



Be sure that this does not result in filename collisions! Use the optional TOC caption entry parameter for formatting. Remember to `\protect` L<sup>A</sup>T<sub>E</sub>X commands which appear in section names and TOC captions.

```
992 \newcommand*{\LWR@filenameno blanks}[1]{%
993 \begingroup
```

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

```
994 \renewcommand*{\HTMLunicode}[1]{%
995 \renewcommand*{\HTMLentity}[1]{##1}
996 \renewcommand*{\LWR@htmltagc}[1]{%
997 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}
```

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

```
998 \renewcommand*{\&}{and}
999 \renewcommand{\textit}[1]{##1}
1000 \renewcommand{\textsc}[1]{##1}
```



```

1001 \renewcommand{\textsl}[1]{##1}
1002 \renewcommand{\textbf}[1]{##1}
1003 \renewcommand{\texttt}[1]{##1}
1004 \renewcommand{\textsf}[1]{##1}
1005 \renewcommand{\textrm}[1]{##1}
1006 \renewcommand{\textsuperscript}[1]{##1}
1007 \renewcommand{\textsubscript}[1]{##1}

```

Replaces common symbols and short words with hyphens:

```

1008 \edef\LWR@thisnewfilename{#1}
1009 \fullexpandarg

```

Convert spaces into hyphens:

```

1010 \StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]

```

Convert punctuation into hyphens:

```

1011 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
1012 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
1013 \StrSubstitute{\LWR@thisnewfilename}%
1014 {\LWR@origampersand}{-}[\LWR@thisnewfilename]
1015 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
1016 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
1017 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
1018 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
1019 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
1020 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
1021 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
1022 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
1023 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
1024 \StrSubstitute{\LWR@thisnewfilename}%
1025 {\textless}{-}[\LWR@thisnewfilename]
1026 \StrSubstitute{\LWR@thisnewfilename}%
1027 {\textgreater}{-}[\LWR@thisnewfilename]
1028 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
1029 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
1030 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
1031 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
1032 \StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
1033 \StrSubstitute{\LWR@thisnewfilename}%
1034 {\textbackslash}{-}[\LWR@thisnewfilename]
1035 \StrSubstitute{\LWR@thisnewfilename}{^}{-}[\LWR@thisnewfilename]
1036 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
1037 %      "~{" for babel
1038 \StrSubstitute{\LWR@thisnewfilename}{[}{-}[\LWR@thisnewfilename]
1039 \StrSubstitute{\LWR@thisnewfilename}{]}{-}[\LWR@thisnewfilename]
1040 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]

```

Convert short words:

```

1041 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
1042 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
1043 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
1044 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
1045 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
1046 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
1047 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
1048 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
1049 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
1050 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
1051 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
1052 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
1053 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
1054 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
1055 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
1056 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
1057 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
1058 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

1059 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
1060 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
1061 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
1062 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
1063 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
1064 %      emdash
1065 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
1066 %      endash
1067 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
1068 \endgroup
1069 }

```

`\LWR@newhtmlfile`  $\{ \langle section\ name \rangle \}$

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.

```
1070 \newcommand*{\LWR@newhtmlfile}[1]{
```

At the bottom of the ending file:

```

1071 \LWR@html elementclassend{section}{textbody}
1072
1073 \LWR@printpendingfootnotes

```

```

1074
1075 \LWR@htmlelement{footer}
1076
1077 \LWR@pagebottom
1078
1079 \LWR@htmlelementend{footer}

```

No bottom navigation if are finishing the home page: Presumably has a table-of-contents printed.

```

1080 \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{ }

```

End of this HTML file:

```

1081 \LWR@stoppars
1082 \LWR@htmltag{/body}\LWR@orignewline
1083 \LWR@htmltag{/html}\LWR@orignewline
1084
1085 \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.

```

1086 \ifbool{FileSectionNames}%
1087 {\LWR@filenamenoblanks{#1}}
1088 {\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.

```

1089 \LWR@htmlblockcomment{%
1090 \LWR@origtextttt{|Start file|}%
1091 \LWR@htmlsectionfilename{\LWR@thisfilename}|}%
1092 }

```

At the top of the starting file:

```

1093 \LWR@stoppars
1094
1095 \LWR@filestart{ -- #1}% there is an EMDash in front of the #1
1096
1097 \LWR@topnavigation
1098
1099 \LWR@htmlelement{header}
1100
1101 \LWR@pagetop

```

```

1102
1103 \LWR@htmlelementend{header}
1104
1105 \ifcvoid{thetitle}{\LWR@printthetitle}
1106
1107 \LWR@sidetoc
1108
1109 \LWR@htmlelementclass{section}{textbody}
1110
1111 \LWR@stoppars
1112
1113 \setcounter{LWR@latestautopage}{\value{page}}
1114 }

1115 \end{warpHTML}

```

## 38 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:** 1116 \begin{warpHTML}

### 38.1 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.



```

1117 \newbool{LWR@mainmatter}
1118 \DeclareDocumentCommand{\mainmatter}{\LWR@mainmatter}{\LWR@mainmatter}%
1119 \booltrue{LWR@mainmatter}%
1120 }

```

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

```
1121 \DeclareDocumentCommand{\frontmatter}{}{%
1122 \boolfalse{LWR@mainmatter}%
1123 }
```

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

```
1124 \DeclareDocumentCommand{\backmatter}{}{%
1125 \boolfalse{LWR@mainmatter}
1126 }
```

## 38.2 Sectioning support macros

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

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

```
1127 \newcommand*{\LWR@sectionnumber}[1]{%
1128 \InlineClass{sectionnumber}{#1} %
1129 }
```

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

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

Create an autosection tag.

```
1130 \newcommand*{\LWR@createautosec}[1]{%
1131 \LWR@htmltag{#1 id="autosec-\thepage"}{}%
1132 }
```

**\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.

```
1133 \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.

```
1134 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
1135 \LWR@closeprevious{#1}%
```

Push a new section depth:

```
1136 \LWR@pushoneclose{#1}{#2}%
```

```
1137 }
```

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.

```
1138 \newcounter{LWR@prevFileDepth}
```

```
1139 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

`\LWR@section` \* [*TOC name*] {*<name>*} {*<sectiontype>*}

The common actions for the high-level sectioning commands.

```
1140 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
```

```
1141 \LWR@traceinfo{LWR@section}%
```

```
1142 \LWR@stoppars%
```

Cancel special `minipage` horizontal space interaction:

```
1143 \global\boolfalse{LWR@minipagethispar}%
```

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

```
1144 \LWR@orignewpage%
```

```
1145
```

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

```
1146 \LWR@traceinfo{LWR@section: testing about to start a new HTML file}%
```

```
1147 \IfBooleanTF{#1}{\% not starred
```

```
1148 \ifthenelse{%
```

```
1149 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}%
```

```
1150 \AND%
```

```
1151 \(%
```

```
1152 \NOT\boolean{CombineHigherDepths}\OR%
```

```

1153 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}%
1154 \)%
1155 }%

```

If so: start a new HTML file:

```

1156 {% new file
1157 \LWR@traceinfo{LWR@section: new HTML file}%

```

See if there was an optional TOC name entry:

```

1158 \IfNoValueTF{#2}%

```

If no optional entry

```

1159 {\LWR@newhtmlfile{#3}}%

```

If yes an optional entry

```

1160 {\LWR@newhtmlfile{#2}}%
1161 }% new file

```

Else: No new HTML file:

```

1162 {}% not new file
1163 }% not starred

```

Remember this section's name for \nameref:

```

1164 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
1165 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%

```

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

```

1166
1167 \LWR@htmlcomment{Opening #4 ‘‘#3’’{}}
1168

```

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

```

1169 \ifthenelse{%
1170 \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
1171 }%
1172 {\LWR@startpars}
1173 {}

```

Create the opening tag with an autosec:

```

1174 \LWR@createautosec{\csuse{LWR@tag#4}}%

```

If not starred, step counter and add to TOC:

```
1175 \IfBooleanTF{#1}%
1176 {}% starred
1177 {}% not starred
```

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

```
1178 \ifthenelse{%
1179 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
1180 }%
1181 {}% if secnumdepth
```

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

```
1182 \LWR@traceinfo{LWR@section: about to test main matter}%
1183 \ifbool{LWR@mainmatter}%
1184 {%
1185 \LWR@traceinfo{LWR@section: yes mainmatter}%
1186 \refstepcounter{#4}%
```

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

```
1187 \LWR@traceinfo{LWR@section: about to addcontentsline}%
1188 \addcontentsline{toc}{#4}%
1189 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
1190 \IfValueTF{#2}{#2}{#3}%
1191 \LWR@traceinfo{LWR@section: finished addcontentsline}%
1192 }% end of if main matter
```

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

```
1193 {}% not main matter
1194 \LWR@traceinfo{LWR@section: no main matter}%
1195 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
1196 }% end of not main matter
1197 }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```
1198 {}%
1199 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
1200 }%
```

For part, print the section type:

```
1201 \ifbool{LWR@mainmatter}%
1202 {%
```



```

1203 \ifthenelse{%
1204 \(\cnttest{\csuse{LWR@depth#4}}{<=}%
1205 {\value{secnumdepth}}\)\ \AND
1206 \(\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\)
1207 }%
1208 {\csuse{#4name}~{}}%
1209 {}%

```

Print the section number:

```

1210 \LWR@traceinfo{LWR@section: about to print section number}%
1211 \ifthenelse{%
1212 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
1213 }%
1214 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
1215 {}%
1216 \LWR@traceinfo{LWR@section: finished print section number}%
1217 }{}%
1218 }% end of not starred

```

Print the section name:

```

1219 #3

```

close the heading tag, such as /H2

```

1220 \LWR@htmltag{\csuse{LWR@tag#4end}}%

```

Generate a L<sup>A</sup>T<sub>E</sub>X label:

```

1221 \label{autopage-\thepage}%

```

Start paragraph handing unless is an inline paragraph or subparagraph:

```

1222 \ifthenelse{%
1223 \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}}%
1224 {\LWR@startpars}%
1225 {}%

```

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.

```

1226 \IfBooleanTF{#1}{% not starred
1227 \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%

```

```

1228 }% not starred
1229 \LWR@traceinfo{LWR@section: done}%
1230 }

```

### 38.3 \section and friends

`\part` \* [*TOC name*] {*name*}

```

1231 \@ifundefined{chapter}
1232 {}
1233 {%
1234 \DeclareDocumentCommand{\part}{s o m}{%
1235 \LWR@stoppars%
1236
1237 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
1238
1239 \LWR@section{#1}{#2}{#3}{part}%
1240 }
1241 }

```

`\chapter` \* [*TOC name*] {*name*}

```

1242 \@ifundefined{chapter}
1243 {}
1244 {%
1245 \DeclareDocumentCommand{\chapter}{s o m}{%
1246 \LWR@traceinfo{chapter #3}%
1247 \LWR@stoppars%
1248 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
1249
1250 \LWR@traceinfo{chapter: about to LWR@section}%
1251 \LWR@section{#1}{#2}{#3}{chapter}%
1252 \LWR@traceinfo{chapter: done}%
1253 }
1254 }

```

`\section` \* [*TOC name*] {*name*}

```

1255 \DeclareDocumentCommand{\section}{s o m}{%
1256 \LWR@stoppars%
1257
1258 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
1259
1260 \LWR@section{#1}{#2}{#3}{section}%
1261 }

```

`\subsection * [(<TOC name>)] {<name>}`

```
1262 \DeclareDocumentCommand{\subsection}{s o m}{%
1263 \LWR@stoppars%
1264
1265 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
1266
1267 \LWR@section{#1}{#2}{#3}{subsection}%
1268 }
```

`\subsubsection * [(<TOC name>)] {<name>}`

```
1269 \DeclareDocumentCommand{\subsubsection}{s o m}{%
1270 \LWR@stoppars%
1271
1272 \LWR@startnewdepth{\LWR@depthsubsubsection}%
1273 {\LWR@printclosesubsubsection}%
1274
1275 \LWR@section{#1}{#2}{#3}{subsubsection}%
1276 }
```

`\paragraph * [(<TOC name>)] {<name>}`

```
1277 \DeclareDocumentCommand{\paragraph}{s o m}{%
1278 \LWR@stoppars%
1279
1280 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
1281
1282 \LWR@section{#1}{#2}{#3}{paragraph}%
1283 }
```

`\subparagraph * [(<TOC name>)] {<name>}`

```
1284 \DeclareDocumentCommand{\subparagraph}{s o m}{%
1285 \LWR@stoppars%
1286
1287 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
1288
1289 \LWR@section{#1}{#2}{#3}{subparagraph}%
1290 }

1291 \end{warpHTML}
```

## 39 Starting a new file

for HTML output:

```
1292 \begin{warpHTML}
```

`\MetaLanguage` Default language for the HTML lang tag.

```
1293 \newcommand*{\MetaLanguage}{en-US}
```

`\LWR@filestart`  $\{\langle title\_suffix \rangle\}$

Creates the opening HTML tags.

```
1294 \newcommand*{\LWR@filestart}[1]{
```

Locally temporarily disable direct-formatting commands:

```
1295 \begingroup
1296 \renewcommand{\textit}[1]{##1}% not used in filenames
1297 \renewcommand{\textsc}[1]{##1}
1298 \renewcommand{\textsl}[1]{##1}
1299 \renewcommand{\textbf}[1]{##1}
1300 \renewcommand{\texttt}[1]{##1}
1301 \renewcommand{\textsf}[1]{##1}
1302 \renewcommand{\textrm}[1]{##1}
1303 \renewcommand{\textsuperscript}[1]{##1}
1304 \renewcommand{\textsubscript}[1]{##1}
1305 \renewcommand*{\HTMLUnicode}[1]{ }
1306 \renewcommand*{\HTMLentity}[1]{ }
1307 \RenewDocumentCommand{\LWR@htmlspanclass}{m o +m}{##3}
1308 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}
```

Create the page's HTML header:

```
1309 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
```

The language is user-adjustable:

```
1310 \LWR@htmltag{html lang="\MetaLanguage"{}\LWR@orignewline
```

Start of the meta data:

```
1311 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
1312 \LWR@htmltag{meta charset="UTF-8" /\LWR@orignewline
```

lwarp is the generator:

```

1313 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /}%
1314   \LWR@orignewline

```

If there is a description, add it now:

```

1315 \ifdefempty{\LWR@currentHTMLdescription}{-}{-}{%
1316 \LWR@htmltag{%
1317 meta name="description" content="\LWR@currentHTMLdescription" /}%
1318   \LWR@orignewline
1319 }%

```

Mobile-friendly viewport:

```

1320 \LWR@htmltag{meta name="viewport" %
1321 content="width=device-width, initial-scale=1.0" /}%
1322   \LWR@orignewline

```

IE patch:

```

1323 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
1324 \LWR@htmltag{%
1325 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{}%
1326 \LWR@htmltag{/script}\LWR@orignewline
1327 \LWR@htmltag{![endif]{-}{-}}\LWR@orignewline

```

The page's title:

```

1328 \ifcvoid{thetitle}{-}{-}{%
1329 \LWR@htmltag{title}\thetitle#1\LWR@htmltag{/title}\LWR@orignewline%
1330 }%

```

The page's stylesheet:

```

1331 \LWR@htmltag{%
1332 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
1333 \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.

```

1334 \ifbool{mathjax}%
1335 {%
1336 \boolfalse{LWR@verbtags}
1337 \VerbatimInput{lwarp_mathjax.txt}%
1338 \booltrue{LWR@verbtags}
1339 \LWR@stoppars
1340 }% end of mathjax
1341 {}%

```

End of the header:

```
1342 \LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
1343 \LWR@htmltag{body}\LWR@orignewline
1344 \endgroup
1345 }
```

```
1346 \end{warpHTML}
```

## 40 Starting HTML output

**for HTML output:** 1347 \begin{warpHTML}

`\LWR@LwarpStart` Executed at the beginning of the entire document.

```
1348 \newcommand*{\LWR@LwarpStart}
1349 {%
```

Expand and detokenize `\HomeHTMLFileName` and `\HTMLFileName`:

```
1350 \edef\LWR@strresult{\HomeHTMLFileName}
1351 \edef\HomeHTMLFileName{\detokenize\expandafter{\LWR@strresult}}
1352 \edef\LWR@strresult{\HTMLFileName}
1353 \edef\HTMLFileName{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn:

```
1354 \LWR@origonecolumn%
```

Reduce chance of line overflow in verbatim environments:

```
1355 \LWR@origscriptsize%
```

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

```
1356 \LWR@origraggedright%
1357 \let\\\LWR@endoffline%
```

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

```
1358 \linespread{1.3}%
```

For `pdftotext` to reliably identify paragraph splits:

```
1359 \setlength{\parindent}{0pt}
1360 \setlength{\parskip}{2ex}
```

For the `lateximages` record file:

```
1361 \immediate\openout\LWR@file=lateximages.txt
```

Removes space after the caption in the HTML:

```
1362 \setlength{\belowcaptionskip}{-3ex}
```

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

```
1363 \renewcommand{\ps@plain}{} 
```

`\centering` Not used in the HTML environment:  
`\raggedleft`  
`\raggedright` 1364 `\renewcommand*{\centering}{}`   
1365 `\renewcommand*{\raggedleft}{}`   
1366 `\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:

```
1367 \let\LWR@origtabular\tabular
1368 \let\LWR@origendtabular\endtabular
1369 \let\tabular\LWR@tabular
1370 \let\endtabular\endLWR@tabular
```

Float captions:

```
1371 \let\LWR@origcaption\caption
```

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

```
1372 \let\LWR@origltx@label\ltx@label
1373 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
1374 \let\TeX\LWR@TeX
1375 \let\LaTeX\LWR@LaTeX
1376 \let\LuaTeX\LWR@LuaTeX
```

```

1377 \let\LuaLaTeX\LWR@LuaLaTeX
1378 \let\XeTeX\LWR@XeTeX
1379 \let\XeLaTeX\LWR@XeLaTeX
1380 \let\ConTeXt\LWR@ConTeXt

```

Graphics:

```

1381 \let\rotatebox\LWR@rotatebox
1382 \let\scalebox\LWR@scalebox
1383 \let\reflectbox\LWR@reflectbox

```

Not yet started any paragraph handling:

```

1384 \global\boolfalse\LWR@doingapar}
1385 \global\boolfalse\LWR@doingstartpars}

```

Start a new HTML file and a header:

```

1386 \LWR@filestart{}
1387 \LWR@htmltag{header}\LWR@orignewline
1388 \LWR@startpars
1389 \LWR@firstpagetop
1390 \LWR@stoppars
1391 \LWR@htmltag{/header}\LWR@orignewline
1392 \LWR@htmltag{section class="textbody"{} }
1393 \LWR@origpagestyle{empty}

```

Document and page settings:

```

1394 \mainmatter
1395 \LWR@origpagenumbering{arabic}

```

Set default titlepage thanks footnote marks. See section [42.6](#).

```

1396 \if@titlepage
1397   \thanksmarkseries{arabic}
1398 \else
1399   \thanksmarkseries{fnsymbol}
1400 \fi

```

Initial default patch for fancyvrb:

```

1401 \fvset{frame=none}%

```

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

```

1402 \catcode'\&=\active

```



Allow HTML paragraphs to begin:

```
1403 \LWR@startpars
1404 }

1405 \end{warpHTML}
```

## 41 Ending HTML output

**for HTML output:** 1406 \begin{warpHTML}

`\LWR@requesttoc`  $\{\langle boolean \rangle\}$   $\{\langle suffix \rangle\}$  Requests that a toc, lof, or lot be generated.

```
1407 \newcommand*{\LWR@requesttoc}[2]{%
1408 \ifbool{#1}
1409 {
1410 \expandafter\newwrite\csuse{tf@#2}
1411 \immediate\openout \csuse{tf@#2} \jobname.#2\relax
1412 }{}
1413 }
```

`\LWR@LwarpEnd` Final stop of all HTML output:

```
1414 \newcommand*{\LWR@LwarpEnd}
1415 {
1416 \LWR@stoppars
1417 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
1418 \LWR@html-elementclassend{section}{textbody}
```

Print any pending footnotes:

```
1419 \LWR@printpendingfootnotes
```

Create the footer:

```
1420 \LWR@html-element{footer}
1421
1422 \LWR@pagebottom
1423
1424 \LWR@html-elementend{footer}
```

No bottom navigation if are finishing the home page:  
Presumably has a table-of-contents printed.

```
1425 \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}
```

```
1426 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
1427 \LWR@htmltag{/body}\LWR@orignewline
1428 \LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
1429 \LWR@orignewpage
```

For `lateximage` commands:

```
1430 \immediate\closeout\LWR@file
1431 }
```

```
1432 \end{warpHTML}
```

## 42 Titles and the titling package

Supports and extends the titling package.


Additional functions include `\published` and `\subtitle`, and the `\author` command has an additional `\affiliation` command to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using `\theauthor` in the main text.

The `titling` package maintains the definitions of `\thetitle`, `\theauthor`, etc., after the title has been typeset. These commands are to be used to refer to the document's title and author, etc., in the main text. These definitions have the `\thanks` and `\affiliation` removed, and for author the `\and` is replaced to generate a simple inline list of authors separated by commas.

`\printtitle`, `\printauthor`, etc., are provided for use inside the `titlepage` or `titlingpage` environments, and these retain the `\thanks` and `\affiliation`.

Several additional hooks are provided in addition to `titling`:

`\maketitlehookaa` **\maketitlehookaa:** Between “published” and the title.

<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.
 <b>No footnotes!</b>	Inside a <code>\titlepage</code> or <code>\titlingpage</code> environment, use <code>\thanks</code> for footnotes, do not use <code>\footnote</code> .  At the end of the <code>titlingpage</code> environment, footnote marks are forced to reset to zero.  Inside a <code>titlingpage</code> environment with the <code>article</code> document class, thanks marks will be <code>fnsymbol</code> instead of <code>arabic</code> . <code>arabic</code> is usually used when inside <code>titlepage</code> environments where the title page is on its own page, but is not automatically used inside a <code>titlingpage</code> environment.  To force the thanks marks to be <code>arabic</code> :
	<hr/> <pre> \begin{titlingpage} \thanksmarkseries{arabic} ... </pre> <hr/>

## 42.1 Setting the title, etc.

The following provide setting commands for both HTML and print outputs.

<code>\published</code>	<code>\@title</code> , <code>\@subtitle</code> , <code>\@author</code> , etc. store the values as originally assigned, including any <code>\thanks</code> , <code>\and</code> , or <code>\affiliation</code> . These are low-level macros intended to be used by other macros only inside a <code>titlepage</code> or <code>titlingpage</code> , and are used by <code>\maketitle</code> . The author is printed inside a single-column table, which becomes multiple single-column tables if multiples authors are included.
<code>\title</code>	
<code>\subtitle</code>	
<code>\author</code>	
<code>\date</code>	
<code>\printpublished</code>	<code>\printtitle</code> , <code>\printsubtitle</code> , etc. are user-level macros intended to be used in <code>titlepage</code> and <code>titlingpage</code> environments in cases where <code>\maketitle</code> is not desired. These commands preserve the <code>\thanks</code> , etc., and should not be used in the main text. The author is printed inside a single-column table, which becomes multiple single-column tables if multiples authors are included.
<code>\printtitle</code>	
<code>\printsubtitle</code>	
<code>\printauthor</code>	
<code>\printdate</code>	

`\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:

`\SetPageBottom` `{\text}`

---

```
\SetPageBottom{
\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: 1433 \begin{warpHTML}
                  1434 \newrobustcmd{\affiliation}[1]{\\ \InlineClass{affiliation}{#1}}
                  1435 \end{warpHTML}
```

```
for PRINT output: 1436 \begin{warpprint}
                  1437 \newrobustcmd{\affiliation}[1]{\\ \textsc{\small#1}}
                  1438 \end{warpprint}
```

The following are based on the original titling code:

for HTML & PRINT: 1439 \begin{warpall}

\author {*text*}

Redefined to nullify \affiliation, etc. before printing the authors inline.

\@author retains the entire author with its \thanks, while \theauthor will have \thanks removed and \and simplified.

```
1440 \renewcommand{\author}[1]{%
1441 \gdef\@author{#1}
1442 \begingroup
1443   \renewcommand{\thanks}[1]{ }
1444   \renewcommand{\and}{\unskip, }
1445   \renewcommand{\thanksmark}[1]{ }
1446   \renewcommand{\thanksgap}[1]{ }
1447   \renewcommand{\affiliation}[1]{ }
1448   \protected@xdef\theauthor{#1}
1449 \endgroup}
```

\published {*text*}

```
1450 \newcommand{\published}[1]{%
1451 \gdef\@published{#1}
1452 \begingroup
1453   \renewcommand{\thanks}[1]{ }
1454   \renewcommand{\thanksmark}[1]{ }
1455   \renewcommand{\thanksgap}[1]{ }
1456   \protected@xdef\thepublished{#1}
1457 \endgroup
1458 }
1459 \newcommand{\@published}{ }
1460 \newcommand{\thepublished}{ }
```

\subtitle {*text*}

```
1461 \newcommand{\subtitle}[1]{%
1462 \gdef\@subtitle{#1}
1463 \begingroup
1464   \renewcommand{\thanks}[1]{ }
1465   \renewcommand{\thanksmark}[1]{ }
1466   \renewcommand{\thanksgap}[1]{ }
1467   \protected@xdef\thesubtitle{#1}
1468 \endgroup
1469 }
1470 \newcommand{\@subtitle}{ }
1471 \newcommand{\thesubtitle}{ }
```

```
1472 \end{warpall}
```

## 42.2 Changes to HTML titlepage and titlingpage

for HTML output: 1473 \begin{warpHTML}

Env **titlepage** Sets up a titlepage div with a L<sup>A</sup>T<sub>E</sub>X PDF minipage inside.

```
1474 \renewenvironment*{titlepage}
1475     {\BlockClass{titlepage}\LWR@subminipage}
1476     {\LWR@endsubminipage\endBlockClass}
```

Env **titlingpage**

```
1477 \renewenvironment*{titlingpage}
1478 {%
```

Start an HTML titlepage div:

```
1479 \begin{titlepage}
```

Prepare for a custom version of \maketitle inside the titlingpage:

```
1480 \LWR@maketitlesetup
1481 \let\maketitle\LWR@titlingmaketitle
1482 }
1483 {
```

At the end of the environment, end the HTML titlepage div:

```
1484 \end{titlepage}
```

Reset the footnote counter:

```
1485 \@bscontmark
1486 }
```

```
1487 \end{warpHTML}
```

for HTML & PRINT: 1488 \begin{warpall}

\printthanks Forces the \thanks to be printed.

This is necessary in a titlingpage environment when \maketitle was not used.

```
1489 \newcommand*{\printthanks}{\@thanks}
```

Env **titlingpage** At the end of the titlingpage for both print and HTML, reset footnote markers to zero.

```
1490 \AtEndEnvironment{titlingpage}{\@bscontmark}
```

```
1491 \end{warpall}
```

## 42.3 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: 1492 \begin{warpHTML}

Patch the pre/post title/author/date to add HTML tags, then initialize:

```
1493 \newcommand{\prepublished}[1]{%
1494 \def\@bspublished{\BlockClass{published}#1}%
1495 }
1496
1497 \newcommand{\postpublished}[1]{%
1498 \def\@bspublished{#1\endBlockClass}%
1499 }
1500
1501 \renewcommand{\pretitle}[1]{%
1502 \def\@bspretitle{#1\LWR@stoppars\LWR@htmltag{h1}}%
1503 }
1504
1505 \renewcommand{\posttitle}[1]{%
1506 \def\@bsposttitle{\LWR@htmltag{/h1}\LWR@startpars#1}%
1507 }
1508
1509 \newcommand{\presubtitle}[1]{%
1510 \def\@bspresubtitle{\BlockClass{subtitle}#1}%
1511 }
1512
1513 \newcommand{\postsubtitle}[1]{%
1514 \def\@bspostsubtitle{#1\endBlockClass}%
1515 }
1516
1517 \renewcommand{\preauthor}[1]{%
1518 \def\@bspauthor{\BlockClass{author}#1}%
1519 }
1520
1521 \renewcommand{\postauthor}[1]{%
1522 \def\@bspauthor{#1\endBlockClass}%
1523 }
```

```

1524
1525 \renewcommand{\predate}[1]{%
1526 \def\@bspredate{#1\BlockClass{titledate}}%
1527 }
1528
1529 \renewcommand{\postdate}[1]{%
1530 \def\@bspostdate{\endBlockClass#1}%
1531 }
1532
1533 \prepublished{\begin{center}}
1534 \postpublished{\par\end{center}}
1535
1536 \pretitle{\begin{center}}
1537 \posttitle{\par\end{center}}
1538
1539 \presubtitle{\begin{center}}
1540 \postsubtitle{\par\end{center}}
1541
1542 \preauthor{\begin{center}}%
1543 \begin{tabular}[t]{c}%
1544 }
1545 \postauthor{\end{tabular}\par\end{center}}
1546
1547 \predate{\begin{center}}
1548 \postdate{\par\end{center}}

```

#### \printpublished

```

1549 \newcommand*\printpublished{%
1550 \ifthenelse{\equal{\thepublished}{}}{
1551 {}
1552 {
1553 \begin{BlockClass}{published}
1554 \@published
1555 \end{BlockClass}
1556 }
1557 }

```

#### \printtitle

```

1558 \newcommand*\printtitle{
1559 {
1560 \LWR@stoppars
1561 \LWR@htmltag{h1}%
1562 \@title%
1563 \LWR@htmltag{/h1}
1564 \LWR@startpars
1565 }

```



`\LWR@printthetitle` A private version which prints the title without footnotes, used to title each HTML page.

```
1566 \newcommand*{\LWR@printthetitle}
1567 {
1568 \LWR@stoppars
1569 \LWR@htmltag{h1}%
1570 \thetitle%
1571 \LWR@htmltag{/h1}
1572 \LWR@startpars
1573 }
```

`\printssubtitle`

```
1574 \newcommand*{\printssubtitle}{
1575 \ifthenelse{\equal{\thesubtitle}{}}{
1576 {}
1577 {
1578 \begin{BlockClass}{subtitle}
1579 \@subtitle
1580 \end{BlockClass}
1581 }
1582 }
```

`\printauthor`

```
1583 \newcommand*{\printauthor}{
1584 \begin{BlockClass}{author}
1585 \begin{tabular}{c}\@author\end{tabular}
1586 \end{BlockClass}
1587 }
```

`\printdate`

```
1588 \newcommand*{\printdate}{%
1589 \begin{BlockClass}{titledate}
1590 \@date
1591 \end{BlockClass}
1592 }
```

```
1593 \end{warpHTML}
```

## 42.4 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: 1594 \begin{warpprint}

\printpublished

1595 \newcommand\*{\printpublished}{\Large\scshape\@published}}

\printtitle

1596 \newcommand\*{\printtitle}{\Huge\@title}}

\printsubtitle

1597 \newcommand\*{\printsubtitle}{\Large\itshape\@subtitle\bigskip}}

\printauthor

1598 \newcommand\*{\printauthor}

1599 {\large\begin{tabular}[t]{c}\@author\end{tabular}}}

\printdate

1600 \newcommand\*{\printdate}{\small\textit{\@date}}}

## 42.5 \maketitle for print output

\maketitle From the titling package, patched to add the publisher and subtitle.

```

1601 \providecommand{\maketitle}{}
1602 \if@titlepage
1603   \renewcommand{\maketitle}{\begin{titlepage}%
1604     \let\footnotesize\small
1605     \let\footnoterule\relax
1606     \let \footnote \thanks
1607     \@bsmarkseries
1608     \def\@makefnmark{\rlap{\@textsuperscript{%
1609       \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
1610     \long\def\@makefntext##1{\makethanksmark ##1}
1611     \null\vfil
1612     \vskip 60\p@
1613     \vspace*{\droptitle}
1614     \maketitlehooka
1615     \ifcseempty{\@published}
1616 {}

```

```

1617 {{\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
1618     {\@bsprettitle \@title \@bsposttitle}
1619     \ifcsempy{\@subtitle}
1620 {}
1621 {\maketitlehookaaaa{\@bspresubtitle \@subtitle \@bspostsubttile}}
1622     \maketitlehookb
1623     {\@bspreauthor \@author \@bspostauthor}
1624     \maketitlehookc
1625     {\@bspredate \@date \@bspostdate}
1626     \maketitlehookd
1627     \par
1628     \@thanks
1629     \vfil\null
1630     \end{titlepage}%
1631     \@bscontmark % \setcounter{footnote}{0}%
1632 %%% \@bsmttitleempty
1633 } % end titlepage defs
1634 \else
1635 \renewcommand{\maketitle}{\par
1636 \begingroup
1637 \bbsmarkseries
1638 \def\@makefnmark{\rlap{\@textsuperscript{%
1639 \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
1640 \long\def\@makefntext##1{\makethanksmark ##1}
1641 \if@twocolumn
1642 \ifnum \col@number=\@ne
1643 \maketitle
1644 \else
1645 \twocolumn[\maketitle]%
1646 \fi
1647 \else
1648 \newpage
1649 \global\@topnum\z@
1650 \maketitle
1651 \fi
1652 \thispagestyle{plain}\@thanks
1653 \endgroup
1654 \@bscontmark % \setcounter{footnote}{0}%
1655 %%% \@bsmttitleempty
1656 } % end non-titlepage
1657
1658 \def\@maketitle{%
1659 \newpage
1660 \null
1661 \vskip 2em%
1662 \vspace*{\droptitle}
1663 \maketitlehooka
1664 \ifcsempy{\@published}
1665 {}
1666 {{\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}

```

```

1667     {\@bspretitle \@title \@bsposttitle}
1668     \ifcsemtty{@subtitle}
1669 {}
1670 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
1671     \maketitlehookb
1672     {\@bspreadauthor \@author \@bspostauthor}
1673     \maketitlehookc
1674     {\@bspredate \@date \@bspostdate}
1675     \maketitlehookd
1676     \par
1677     \vskip 1.5em}
1678 \fi
1679
1680 \providecommand{\maketitlehookaa}{}
1681
1682 \providecommand{\maketitlehookaaa}{}
1683
1684 \newcommand{\prepublished}[1]{%
1685 \def\@bspublished{#1}%
1686 }
1687
1688 \newcommand{\postpublished}[1]{%
1689 \def\@bspublished{#1}%
1690 }
1691
1692 \newcommand{\presubtitle}[1]{%
1693 \def\@bspresubtitle{#1}%
1694 }

```

`\presubtitle` Hook after printing the subtitle.

```

1695 \newcommand{\postsubtitle}[1]{%
1696 \def\@bspublished{#1}%
1697 }

```

Initial settings:

```

1698 \if@titlepage
1699 \prepublished{
1700 \vspace*{-\baselineskip}\vspace*{-\medskipamount}\vspace*{-2em}
1701 \begin{center}}
1702 \postpublished{\par\end{center}\vskip 2em}
1703
1704 \presubtitle{\unskip\begin{center}\unskip}
1705 \postsubtitle{\par\end{center}\vskip 2em}
1706 \else
1707 \prepublished{\begin{center}}
1708 \postpublished{\par\end{center}\vskip 0.5em}
1709

```

```

1710 \presubtitle{\begin{center}\unskip}
1711 \postsubtitle{\par\end{center}\vskip 0.5em}
1712 \fi

1713 \end{warpprint}

```

## 42.6 \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 35 for other kinds of footnotes.

See `\thanksmarkseries{series}`, below, to set the style of the footnote marks.

for HTML output: 1714 \begin{warppHTML}

\LWR@maketitlesetup Patches \thanks macros to use L<sup>A</sup>T<sub>E</sub>X minipage footnotes.

```
1715 \newcommand*\LWR@maketitlesetup{%
```

Select which kind of footnote marks to use:

```

1716 \@bsmarkseries
1717 \@mpbsmarkseries

```

Redefine the footnote mark:

```
1718 \def\@makefnmark{\textsuperscript{\thefootnote}}
```

```

\thefootnote ⇒ \nameuse{arabic}{footnote}, or
\thefootnote ⇒ \nameuse{fnsymbol}{footnote}

```

Redefine the footnote text:

```
1719 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
1720 \makethanksmark \LWR@origspace{1in}
```

`\makethanksmark`  $\Rightarrow$  `\thanksfootmark`  $\Rightarrow$  `\tamark`  $\Rightarrow$   
`\@thefnmark`  $\Rightarrow$  `\itshape` a (or similar)

Print the text:

```
1721 ##1%
1722 }%
1723 }
```

`\@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`.

```
1724 \def\@fnsymbol#1{\ensuremath{\ifcase#1\or *\or \dagger\or \ddagger\or
1725   \mathsection\or \mathparagraph\or \text{\HTMLunicode{2016}}\or
1726   **\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.

```
1727 \renewcommand*\maketitle}{%
```

An HTML titlepage div is used for all classes.

```
1728 \begin{titlepage}
```

Set up special patches:

```
1729 \LWR@maketitlesetup
```

Typeset the title, etc:

```
1730 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
1731 \@thanks
```

Close the HTML titlepage div:

```
1732 \end{titlepage}
```

Reset the footnote counter:

```
1733 \@bscontmark
1734 }
```

`\@maketitle` Typesets the title, etc. for HTML:

```

1735 \DeclareDocumentCommand{\@maketitle}{-}{%
1736 \maketitlehooka
1737 \ifcempty{@published}
1738 {}
1739 {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
1740 {\@bsprettitle \@title \@bsposttitle}
1741 \ifcempty{@subtitle}
1742 {}
1743 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
1744 \maketitlehookb
1745 {\@bspreauthor \@author \@bspostauthor}
1746 \maketitlehookc
1747 {\@bspredate \@date \@bspostdate}
1748 \maketitlehookd
1749 }

1750 \providecommand{\maketitlehookaa}{}
1751 \providecommand{\maketitlehookaaa}{}

```

`\LWR@titlingmaketitle` `\maketitle` for use inside an HTML titlingpage environment.

```

1752 \newcommand*{\LWR@titlingmaketitle}{%

```

Typeset the title, etc:

```

1753 \@maketitle

```

Immediately generate any `\thanks` footnotes:

```

1754 \@thanks
1755 }

```

`\thanksmarkseries` `{\series}`

Sets the type of footnote marks used by `\thanks`, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc. Modified to use the L<sup>A</sup>T<sub>E</sub>X PDF minipage which is included with the title page.

```

1756 \renewcommand{\thanksmarkseries}[1]{%
1757 \def\@mpbmarkseries{%
1758 \renewcommand*{\thempfootnote}{\@nameuse{#1}{mpfootnote}}}%
1759 \def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
1760 }

1761 \end{warpHTML}

```

## 43 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: 1762 \begin{warpHTML}

\abstractname User-redefinable title for the abstract.

                Also over-written by the babel package.

1763 \providecommand*{\abstractname}{Abstract}

Env abstract

1764 \DeclareDocumentEnvironment{abstract}{}
1765 {
1766 \BlockClass{abstract}
1767 \BlockClassSingle{abstracttitle}{\abstractname}
1768 }
1769 {
1770 \endBlockClass
1771 }

1772 \end{warpHTML}
```

## 44 Quote and verse

### 44.1 Citations and attributions

`\attribution` for use inside quote, quotation, verse:

ex: `\attribution{author name} --- \citetitle{book name}`

```
for HTML output: 1773 \begin{warpHTML}
1774 \newcommand{\attribution}[1]{%
1775 \InlineClass{attribution}{--\,#1}}% emdash
1776 \end{warpHTML}
```

```
for PRINT output: 1777 \begin{warpprint}
1778 \newcommand{\attribution}[1]{\textsc{--\,#1}}
1779 \end{warpprint}
```

`\citetitle` for use inside quote, quotation, verse:



for HTML output: 1780 `\begin{warpHTML}`  
 1781 `\newcommand{\citetitle}[1]{%`  
 1782 `\InlineClass{citetitle}{--\,#1}}% emdash`  
 1783 `\end{warpHTML}`

for PRINT output: 1784 `\begin{warpprint}`  
 1785 `\newcommand{\citetitle}[1]{\textsl{---\,#1}}`  
 1786 `\end{warpprint}`

## 44.2 Quotes, quotations

for HTML output: 1787 `\begin{warpHTML}`

Env quote

```
1788 \renewenvironment*{quote}
1789 {\LWR@htmlblocktag{blockquote}}
1790 {\LWR@htmlblocktag{/blockquote}}
1791
1792 \renewenvironment*{quotation}
1793 {\LWR@htmlblocktag{blockquotation}}
1794 {\LWR@htmlblocktag{/blockquotation}}
1795 \end{warpHTML}
```

## 44.3 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}
```

---

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`

Len `\leftmargini`

Len `\TMLvleftskip`

Len `\TMLleftmargini`

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

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

**for HTML & PRINT:** 1796 `\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.

```
1797 \newlength{\TMLvleftskip}
1798 \setlength{\TMLvleftskip}{1em}
```

Len `\TMLleftmargini` Sets `\leftmargini` inside a `verse` environment in HTML.

```
1799 \newlength{\TMLleftmargini}
1800 \setlength{\TMLleftmargini}{4.5em}

1801 \end{warpall}
```

## 45 Verbatim

**for HTML output:** 1802 `\begin{warpHTML}`

Env `verbatim`

```
1803 \AfterEndPreamble{
1804 \AtBeginEnvironment{verbatim}{\LWR@atbeginverbatim{verbatim}\unskip\vspace*{--\baselineskip}}
1805 \AfterEndEnvironment{verbatim}{\unskip\vspace*{--\baselineskip}\LWR@afterendverbatim}
1806 }

1807 \end{warpHTML}
```

## 46 Fancyvrb

**for HTML & PRINT:** 1808 `\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.

```
1809 \newlength{\VerbatimHTMLWidth}
1810 \setlength{\VerbatimHTMLWidth}{4in}
1811 \end{warpall}
```

**for HTML output:** 1812 \begin{warpHTML}

Bool    LWR@verbtags    Used to temporarily turn off verbatim tags while doing VerbatimInput in the HTML head.

```
1813 \newbool{LWR@verbtags}
1814 \booltrue{LWR@verbtags}
```

For \VerbatimFootnotes:

```
1815 \renewcommand{\VerbatimFootnotes}{
1816 \PackageError{lwarp}
1817 {Verbatim footnotes are not yet supported by lwarp.}
1818 {This may be improved some day.}
1819 }
```

\LWR@atbeginverbatim    {\class}

Encloses a verbatim environment with the given CSS class.

```
1820 \newcommand*{\LWR@atbeginverbatim}[1]
1821 {%
```

Avoid excessive space between lines:

```
1822 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
1823 \LWR@stoppars%
```

Create a new `pre` of the given class:

```
1824 \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.

```
1825 \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:

```
1826 \let\hspace\LWR@orighspace%
1827 }
```

`\LWR@afterendverbatim` Finishes enclosing a verbatim environment.

```
1828 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the `pre`:

```
1829 \unskip%
```

At the end of the environment, close the `pre`:

```
1830 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}}
1831
1832 }{}%
```

Resume regular paragraph handling:

```
1833 \LWR@startpars%
1834 }
```

`\LWR@Verbatimclass` Holds the class of the following verbatim.

```
1835 \newcommand*{\LWR@Verbatimclass}{fancyvrb}
```

Env `VerbatimClass` `{\class}` [*Verbatim options*]

Creates a `Verbatim` enclosed in a `DIV` of the given class.

```
1836 \NewDocumentEnvironment{VerbatimClass}{m 0{}}
1837 {%
1838 \renewcommand*{\LWR@Verbatimclass}{#1}%
1839 \LWR@origVerbatim[#2]%
1840 }
1841 {\endVerbatim}
```

After the preamble is loaded, after any patches to `Verbatim`:

```
1842 \AfterEndPreamble{
```

Remember the original definition of `Verbatim`:

```
1843 \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.

```
1844 \renewcommand*{\Verbatim}{%
1845 \renewcommand*{\LWR@Verbatimclass}{fancyvrb}%
1846 \LWR@origVerbatim%
1847 }
```

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.

```
1848 \newcommand*{\LWR@fvstartnone}{%
1849 \LWR@traceinfo{fvstartnone}%
1850 \ifbool{LWR@verbtags}{\hbox to\z@{\LWR@htmltagc{div class="\LWR@Verbatimclass"}}}{}%
1851 \hbox to\z@{\LWR@atbeginverbatim{verbatim}}}%
1852 }
1853
1854 \newcommand*{\LWR@fvendnone}{%
1855 \LWR@traceinfo{fvendnone}%
1856 \hbox to\z@{\LWR@afterendverbatim}%
1857 \ifbool{LWR@verbtags}{\hbox to\z@{\LWR@htmltagc{/div}}}{}%
1858 }
1859
1860 \newcommand*{\LWR@fvstartsingle}{%
1861 \LWR@traceinfo{fvstartsingle}%
1862 \LWR@fvstartnone%
1863 \FV@BeginListFrame@Single%
1864 }
1865
1866 \newcommand*{\LWR@fvendsingle}{%
1867 \LWR@traceinfo{fvendsingle}%
1868 \FV@endListFrame@Single%
1869 \LWR@fvendnone%
1870 }
1871
1872 \newcommand*{\LWR@fvstartline}{%
1873 \LWR@traceinfo{fvstartline}%
1874 \LWR@fvstartnone%
1875 \FV@BeginListFrame@Lines%
1876 }
1877
1878 \newcommand*{\LWR@fvendline}{%
1879 \LWR@traceinfo{fvendline}%
1880 \FV@endListFrame@Lines%
1881 \LWR@fvendnone%
1882 }
```

The following patches select the start/left/right/end behaviors depending on `frame`.  
Original code is from the `fancyvrb` package.

```

1883 \def\FV@Frame@none{%
1884 \let\FV@BeginListFrame\LWR@fvstartnone%
1885 \let\FV@LeftListFrame\relax%
1886 \let\FV@RightListFrame\relax%
1887 \let\FV@EndListFrame\LWR@fvendnone}
1888
1889 \def\FV@Frame@single{%
1890 \let\FV@BeginListFrame\LWR@fvstartsingle%
1891 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
1892 \let\FV@RightListFrame\FV@RightListFrame@Single%
1893 \let\FV@EndListFrame\LWR@fvendsingle}
1894
1895 \def\FV@Frame@lines{%
1896 \let\FV@BeginListFrame\LWR@fvstartline%
1897 \let\FV@LeftListFrame\relax%
1898 \let\FV@RightListFrame\relax%
1899 \let\FV@EndListFrame\LWR@fvendline}
1900
1901 \def\FV@Frame@topline{%
1902 \let\FV@BeginListFrame\LWR@fvstartline%
1903 \let\FV@LeftListFrame\relax%
1904 \let\FV@RightListFrame\relax%
1905 \let\FV@EndListFrame\LWR@fvendnone}
1906
1907 \def\FV@Frame@bottomline{%
1908 \let\FV@BeginListFrame\LWR@fvstartnone%
1909 \let\FV@LeftListFrame\relax%
1910 \let\FV@RightListFrame\relax%
1911 \let\FV@EndListFrame\LWR@fvendline}
1912
1913 \def\FV@Frame@leftline{%
1914 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
1915 \ifx\FancyVerbFillColor\relax%
1916 \let\FV@FrameFillLine\relax%
1917 \else%
1918 \@tempdima\FV@FrameRule\relax%
1919 \multiply\@tempdima-\tw@%
1920 \edef\FV@FrameFillLine{%
1921 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}}%
1922 \kern-\number\@tempdima sp}}%
1923 \fi%
1924 \let\FV@BeginListFrame\LWR@fvstartnone%
1925 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
1926 \let\FV@RightListFrame\relax%
1927 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the `fancyvrb` package.

```

1928 \def\FV@SingleFrameLine#1{%
1929   \hbox to\z@{%
1930     \kern\leftmargin
1931     \ifnum#1=\z@\relax
1932       \let\FV@Label\FV@LabelBegin
1933     \else
1934       \let\FV@Label\FV@LabelEnd
1935     \fi
1936     \ifx\FV@Label\relax
1937 %       \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
1938     \else
1939       \ifnum#1=\z@
1940 %         \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
1941         \ifx\FV@LabelPositionTopLine\relax
1942 \else
1943       \LWR@htmltagc{div class="fancyvrblabel"}
1944 \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
1945 \LWR@htmltagc{/div}
1946 \fi
1947       \else
1948 %       \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
1949       \ifx\FV@LabelPositionBottomLine\relax
1950 \else
1951       \LWR@htmltagc{div class="fancyvrblabel"}
1952 \LWR@origtextrm{\FV@LabelEnd}
1953 \LWR@htmltagc{/div}
1954 \fi
1955       \fi
1956     \fi
1957   \fi
1958   \hss
1959 }
1960 }

```

Processes each line, adding optional line numbers. Original code is from the `fancyvrb` package.

```

1961 \def\FV@ListProcessLine#1{%
1962   \hbox to \hsize{%
1963     \kern\leftmargin
1964     \hbox to \VerbatimHTMLWidth {%
1965       \ifcvoid\FV@LeftListNumber}{-}{\kern 2.5em}%
1966 \FV@LeftListNumber%
1967 %       \FV@LeftListFrame
1968       \FancyVerbFormatLine{#1}%
1969       \hss%

```

```

1970 %          \FV@RightListFrame
1971          \FV@RightListNumber%
1972 }%
1973          \hss% required to avoid underfull hboxes
1974 }
1975 }

```

Env **BVerbatim**

```

1976 \AtBeginEnvironment{BVerbatim}
1977 {
1978 \LWR@atbeginverbatim{bverbatim}
1979
1980 }
1981
1982 \AfterEndEnvironment{BVerbatim}
1983 {
1984 \leavevmode\par\vspace{-\baselineskip}
1985 \LWR@afterendverbatim
1986 }

```

Env **LVerbatim** No changes required.

End of the modifications to make at the end of the preamble:

```

1987 } % \AfterEndPreamble

```

**\UseVerbatim**  $\{\langle text \rangle\}$

No changes required.

```

1988 \end{warpHTML}

```

## 47 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 **theorem**.
- 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: 1989 \begin{warpHTML}

\@begintheorem {<name>} {<number>}

```
1990 \renewcommand{\@begintheorem}[2]{%
1991 \BlockClass{theoremcontents}
1992 \InlineClass{theoremlabel}{#1\ #2\ }
1993 }
```

\@opargbegintheorem {<name>} {<number>} {<oparg>}

```
1994 \renewcommand{\@opargbegintheorem}[3]{%
1995 \BlockClass{theoremcontents}
1996 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
1997 }
```

\@endtheorem


```
1998 \renewcommand*\@endtheorem{%
1999 \endBlockClass% theoremcontents
2000 }
```

```
2001 \end{warpHTML}
```

## 48 Lists

If using babel with French, use

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

 **French** 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.

### 48.1 Itemize

for HTML output: 2002 \begin{warpHTML}

```
2003 \let\LWR@origitem\item
```

\LWR@itemizeitem [*label*]

Handles `\item` inside an `itemize` or `enumerate`.

See `\LWR@openparagraph` where extra `\hspace` is used to leave room for the label while inside a list during paragraph construction.

```

2004 \newcommand*{\LWR@itemizeitem}{%
2005 \LWR@stoppars%
2006 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloseitem{}}%
2007 \LWR@htmltag{li}%
2008 \LWR@startpars%
2009 \LWR@origitem%
2010 }
```

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)*]

```

2011 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
2012
2013 \newcommand*{\LWR@itemizestart}{%
2014 \LWR@stoppars%
2015 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
2016 \LWR@htmltag{ul style="list-style-type:none">{}}%
2017 \LWR@startpars%
2018 \let\item\LWR@itemizeitem%
2019 }
2020
2021 \AtEndEnvironment{itemize}{\LWR@itemizeend}
2022
2023 \newcommand*{\LWR@itemizeend}{%
2024 \LWR@stoppars%
2025 \LWR@closeprevious{\LWR@depthlistitem}%
2026 \LWR@closeoneprevious{}}%
2027 \LWR@startpars%
2028 }
```

## 48.2 Enumerate

An HTML unordered list is used with customized L<sup>A</sup>T<sub>E</sub>X-generated labels.

Env **enumerate** [*enumitem options*]

```

2029 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
2030
2031 \newcommand*{\LWR@enumeratestart}{%
2032 \LWR@stoppars%
2033 \LWR@pushhonestclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
2034 \LWR@htmltag{ul style="list-style-type:none">{}}%
2035 \LWR@startpars%
2036 \let\item\LWR@itemizeitem%
2037 }
2038
2039
2040 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
2041
2042 \newcommand*{\LWR@enumerateend}{%
2043 \LWR@stoppars%
2044 \LWR@closeprevious{\LWR@depthlistitem}%
2045 \LWR@closeoneprevious{}}%
2046 \LWR@startpars%
2047 }

```

### 48.3 Description

**\LWR@descitem** [*label*] Handles an `\item` inside a description.

```

2048 \newcommand*{\LWR@descitem}[1][]%
2049 {%
2050 \LWR@stoppars%
2051 \LWR@setlatestname{#1}%
2052 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem{}}%
2053 \LWR@origitem[]%

```

Be sure the label doesn't print to the left of the rest of the file:

```

2054 \LWR@orighspace{1in}
2055 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
2056 \LWR@orignewline%
2057 \LWR@htmltag{dd}%
2058 \LWR@startpars%
2059 }

```

Env **description** [*enumitem options*]

```

2060 \AtBeginEnvironment{description}{\LWR@descriptionstart}
2061

```

```

2062 \newcommand*{\LWR@descriptionstart}{%
2063 \LWR@stoppars%
2064 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}%
2065 \LWR@htmltag{dl}%
2066 \LWR@startpars%
2067 \let\item\LWR@descitem%
2068 }
2069
2070 \AtEndEnvironment{description}{\LWR@descriptionend}
2071
2072 \newcommand*{\LWR@descriptionend}{%
2073 \LWR@stoppars%
2074 \LWR@closeprevious{\LWR@depthlistitem}%
2075 \LWR@closeoneprevious{}}%
2076 \LWR@startpars%
2077 }

```

`\newlist`     $\langle name \rangle$     $\langle type \rangle$     $\langle maxdepth \rangle$

`\renewlist`    $\langle name \rangle$     $\langle type \rangle$     $\langle maxdepth \rangle$

For `enumitem` lists, new lists must have the start and end actions assigned to the new environment. Renewed lists already have their actions assigned, and thus need no changes.

```

2078 \let\LWR@orignewlist\newlist
2079
2080 \renewcommand*{\newlist}[3]{%
2081 \LWR@orignewlist{#1}{#2}{#3}%
2082 \AtBeginEnvironment{#1}{\csuse{\LWR@#2start}}%
2083 \AtEndEnvironment{#1}{\csuse{\LWR@#2end}}%
2084 }
2085 \end{warpHTML}

```

## 49 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

Limitations:

- [column types](#)
- Vertical rules are not yet supported.
  - `*` in a column specification is not used (so far). Repeat the column type the correct number of times.

- Only one each of @, !, >, and < may be used at each column, and they are used in that order.
- \newcolumnntype is ignored; unknown column types are set to 1.
- tabularx ignores the width, but X columns do produce paragraph columns or multicolumns.

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

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

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

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

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

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

- For longtable, place headings and footings which do not apply to HTML inside \warpprintonly{ }.

- For \toprule and \bottomrule, when combined with a warpprint or warpHTML environment, if a “misplaced \noalign” error occurs, change  
This & That \endhead  
to  
\warpprintonly{This & That \endhead}  
and likewise with the other \end headings. Keep the \endfirsthead row unchanged, as it is still relevant to HTML output.

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

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

⚠ \multirow &  
\multicolumn

⚠ \multirow

\multirow with rules

rule at last row

⚠ paragraphs

\cmidrule width, trim

longtable headings

⚠ \warpprintonly

⚠ S columns

## 49.1 Token lookahead

Used by `\LWR@futurenonSPACElet` to look at the next token.

for HTML output: 2086 `\begin{warphTML}`

`\LWR@mynexttoken`

2087 `\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`

```
2088 \def\LWR@futurenonSPACElet#1{\def\LWR@cs{#1}%
2089 \afterassignment\LWR@fnslone\let\nexttoken= }
2090 \def\LWR@fnslone{\expandafter\futurelet\LWR@cs\LWR@fnsltwo}
2091 \def\LWR@fnsltwo{%
2092 \expandafter\ifx\LWR@cs\@sptoken\let\next=\@BTfnslthree%
2093 \else\let\next=\nexttoken\fi\next}
2094 \def\@BTfnslthree{\afterassignment\LWR@fnslone\let\next= }
```

`\LWR@getmynexttoken` Looks ahead and copies the next token into `\LWR@mynexttoken`.

```
2095 \newcommand*{\LWR@getmynexttoken}{%
2096 % nothing must follow this next line
2097 \LWR@futurenonSPACElet\LWR@mynexttoken\LWR@tabledatacolumnntag
2098 }
```

## 49.2 Booleans

Bool `LWR@startedrow` True if should print a row tag before this column.

```
2099 \newbool{LWR@startedrow}
2100 \boolfalse{LWR@startedrow}
```

Bool `LWR@doinghline` True if the next row will have an hline above it.

```
2101 \newbool{LWR@doinghline}
2102 \boolfalse{LWR@doinghline}
```

Bool LWR@doingtbrule True if the next row will have a top/bottom rule above it.

```
2103 \newbool{LWR@doingtbrule}
2104 \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.

```
2105 \newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty multi-row cell, and thus there is no data tag to close.

```
2106 \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.

```
2107 \newbool{LWR@intabularmetadata}
2108 \boolfalse{LWR@intabularmetadata}
```

### 49.3 Handling & and !

For technical discussion regarding problems redefining \&, See:

<http://tex.stackexchange.com/questions/11638/where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860>

\LWR@closetabledatacell If LWR@skippingmrowcell then there is no data tag to close. Otherwise, close any paragraphs, then close the data tag.

```
2109 \newcommand*{\LWR@closetabledatacell}{%
2110 \global\booltrue{LWR@intabularmetadata}%
2111 \ifbool{LWR@exitingtabular}{}%
2112 {% not exiting tabular
2113 \ifbool{LWR@skippingmrowcell}{}%
2114 {% not skippingmrowcell
```

Insert any < then any @ and ! column contents:

```
2115 \unskip%
2116 \LWR@getexparray{LWR@colafterspec}{\theLWR@tablecolspos}%
2117 \LWR@getexparray{LWR@colatspec}{\theLWR@tablecolspos}%
2118 \LWR@getexparray{LWR@colbangspec}{\theLWR@tablecolspos}%
```

Close paragraphs:

```
2119 \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
2120 \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell:

```
2121 \unskip\LWR@htmltag{/td}\LWR@orignewline%
2122 }% not skipping mrowcell
2123 }% not exiting tabular
2124 \global\boolfalse{LWR@skippingmrowcell}%
2125 }
```

`LWR@tabulardepth` tracks whether `&` is being used inside a `tabular`.

```
2126 \newcounter{LWR@tabulardepth}
2127 \setcounter{LWR@tabulardepth}{0}
2128
```

When not used inside a `tabular`, `&` performs its original function as recorded here ( with catcode 4 ).

```
2129 \def\LWR@origampmacro{&}
```

See below for why the group is used.

```
2130 \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.

```
2131 \catcode'\&=\active
2132
2133 \gdef&{%
2134 \ifthenelse{\cnttest{\value{LWR@tabulardepth}}{>}{0}}{%
2135 {%
```

If not skipping a multirow cell, close the current data cell.

```
2136 \unskip%
2137 \LWR@closetabledatacell%
```

Move to the next column.

```
2138 \addtocounter{LWR@tablecolspos}{1}%
```

Look at the next token to decide multi or single column data tag.



```

2139 \LWR@getmynexttoken%
2140 }%

```

If not inside a tabular, performs the original action:

```

2141 {\LWR@origampmacro}%
2142 }
2143 \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.

## 49.4 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendoffline

Throws away options \\[dim] or \\\*

\LWR@tabularendoffline

```

2144 \NewDocumentCommand{\LWR@tabularendoffline}{s o}
2145 {%
2146 \LWR@closetabledatacell%

```

Finish the previous row:

```

2147 \LWR@htmltag{/tr}\LWR@orignewline
2148 \global\booltrue{\LWR@intabularmetadata}

```

Not yet started a table row:

```

2149 \global\boolfalse{\LWR@startedrow}

```

Additional setup:

```

2150 \global\boolfalse{\LWR@doinghline}%
2151 \global\boolfalse{\LWR@doingtbrule}%
2152 \LWR@clearmidrules%

```

Start at first column:

```

2153 \setcounter{\LWR@tablecolspos}{1}

```

Look at the next token to decide between single column data tag or a special case:

```
2154 \LWR@getmynexttoken%
2155 }
```

## 49.5 Variables

```
2156 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
2157 \newcommand*{\LWR@pposition}{}
2158 \newcommand*{\LWR@pleft}{}
2159 \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.

```
2160 \newcommand*{\LWR@tablecolspec}{}

```

`\LWR@strresult` Holds the result of `Str` functions.

```
2161 \newcommand*{\LWR@strresult}{}

```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

```
2162 \newcommand*{\LWR@origcolspec}{}

```

`Ctr` `LWR@tablecolwidth` Holds the width of the table specification.

(This is not the total # columns.)

```
2163 \newcounter{LWR@tablecolwidth}

```

`Ctr` `LWR@tablecolspos` Where are currently looking into the table column specification.

```
2164 \newcounter{LWR@tablecolspos}

```

`Ctr` `LWR@tabletotalcols` Holds the final number of table columns.

```
2165 \newcounter{LWR@tabletotalcols}

```

`Ctr` `LWR@tabletotalcolsnext` Holds the next column while parsing. Is one more than `LWR@tabletotalcols`.

```
2166 \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 23.
- LWR@colbangspec** A data array of specifications for ! columns. The leftmost's index is `leftedge`, the others are counter values. See section 23.
- LWR@colbeforespec** A data array of specifications for > columns.
- LWR@colafterspec** A data array of specifications for < columns.

## 49.6 Parsing @, >, <, ! columns

`\LWR@parseatcolumn` Handles @{text} columns.

```
2167 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
2168 \LWR@traceinfo{at column}%
2169 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into `\LWR@strresult`, expanding once:

```
2170 \LWR@traceinfo{about to read the next token:}%
2171 \expandarg%
2172 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]
2173 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
2174 \LWR@traceinfo{have now read the next token}%
2175 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}{
2176 {% left edge of the table:
2177 \LWR@traceinfo{at the left edge}%
2178 \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@strresult}%
2179 }%
2180 {% not at the left edge:
2181 \LWR@traceinfo{not at the left edge}%
2182 \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
2183 \LWR@traceinfo{at \theLWR@tabletotalcols: %
2184 \LWR@getexparray{LWR@colatspec}{\theLWR@tabletotalcols}}!}%
2185 \let\LWR@strresult\relax%
2186 \booltrue{LWR@validtablecol}%
2187 }%
2188 }
```

\LWR@parsebangcolumn

```
2189 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
2190 \LWR@traceinfo{bang column}%
```

```
2191 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into \LWR@strresult, expanding once:

```
2192 \LWR@traceinfo{about to read the next token:}%
```

```
2193 \expandarg%
```

```
2194 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]
```

```
2195 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```
2196 \LWR@traceinfo{have now read the next token}%
```

```
2197 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}
```

```
2198 {% left edge of the table:
```

```
2199 \LWR@traceinfo{at the left edge}%
```

```
2200 \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@strresult}%
```

```
2201 }%
```

```
2202 {% not at the left edge:
```

```
2203 \LWR@traceinfo{not at the left edge}%
```

```
2204 \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
```

```
2205 \LWR@traceinfo{bang \theLWR@tabletotalcols: \LWR@colbangspec(\theLWR@tabletotalcols)!}%
```

```
2206 }%
```

```
2207 \let\LWR@strresult\relax%
```

```
2208 \booltrue{LWR@validtablecol}%
```

```
2209 }
```

\LWR@parsebeforecolumn Handles >{text} columns.

```
2210 \newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
2211 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token, expanding once into \LWR@strresult:

```
2212 \expandarg%
```

```
2213 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
```

```
2214 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```

2215 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\LWR@strresult}%
2216 \let\LWR@strresult\relax%
2217 \booltrue{LWR@validtablecol}%
2218 }

```

`\LWR@parseaftercolumn` Handles <{text} columns.

```

2219 \newcommand*{\LWR@parseaftercolumn}{%

```

Move to the next token after the '<':

```

2220 \addtocounter{LWR@tablecolspos}{1}%

```

Read the next token, expanding once into `\LWR@strresult`:

```

2221 % \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]
2222 \expandarg%
2223 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
2224 \fullexpandarg%

```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```

2225 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
2226 \let\LWR@strresult\relax%
2227 \booltrue{LWR@validtablecol}%
2228 }

```

`\LWR@parseskipcolumn` Handles columns to skip, such as the vertical bar.

```

2229 \newcommand*{\LWR@parseskipcolumn}{%
2230 \booltrue{LWR@validtablecol}%
2231 }

```

## 49.7 Parsing 'l', 'c', or 'r' columns

`\LWR@parsenormalcolumn` *{\thiscolumn}*

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```

2232 \newcommand*{\LWR@parsenormalcolumn}[1]{%
2233 \appto\LWR@tablecolspec{#1}%
2234 \addtocounter{LWR@tabletotalcols}{1}%
2235 \addtocounter{LWR@tabletotalcolsnext}{1}%
2236 \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcolsnext}{\relax}%

```

```

2237 \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcolsnext}{\relax}%
2238 \LWR@setexparray{LWR@colbefore-spec}{\theLWR@tabletotalcolsnext}{\relax}%
2239 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcolsnext}{\relax}%
2240 \booltrue{LWR@validtablecol}%
2241 }

```

## 49.8 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn`  $\{ \langle thiscolumn \rangle \}$  The width will be ignored.

```
2242 \newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type.

```
2243 \LWR@parsenormalcolumn{#1}%
```

skips the following width

```
2244 \addtocounter{LWR@tablecolspos}{1}%
2245 }
```

## 49.9 Parsing ‘D’ columns

From the dcolumn package.

`\LWR@parseDcolumn`  $\{ \langle thiscolumn \rangle \}$  The three parameters will be ignored.

```
2246 \newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
2247 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
2248 \addtocounter{LWR@tablecolspos}{3}%
2249 }
```

## 49.10 Parsing the column specifications



HTML CSS cannot exactly match the  $\text{\LaTeX}$  concept of a baseline for a table row. Table 6 shows the  $\text{\LaTeX}$  results for various vertical-alignment choices, with the

Table 6: Tabular baseline

l	p	m	b	r
			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

baseline of the first column drawn across all the columns for comparison. See the **p** column specification in table 7 for details.

Table 7 describes how each kind of column is converted to HTML.

Bool LWR@validtablecol True if found a valid table column type.

2250 \newbool{LWR@validtablecol}

\LWR@parsetablecols {\colspecs}

Scans the column specification left to right.

Builds \LWR@tablecolspec with the final specification, one column per entry. The number of final columns is stored in LWR@tabletotalcols.

2251 \newcommand\*{\LWR@parsetablecols}[1]{%

2252 \LWR@traceinfo{LWR@parsetablecols started}%

Remember the original supplied column spec:

2253 \renewcommand\*{\LWR@origcolspec}{#1}%

Clear the parsed resulting column spec:

2254 \renewcommand\*{\LWR@tablecolspec}{}%

Total number of columns found so far. Also pre-initialize the first several columns of specs:

2255 \setcounter{LWR@tabletotalcols}{0}%

2256 \setcounter{LWR@tabletotalcolsnxt}{1}%

2257 \LWR@setexparray{LWR@colatspec}{leftedge}{\relax}%

2258 \LWR@setexparray{LWR@colatspec}{1}{\relax}%

2259 \LWR@setexparray{LWR@colatspec}{2}{\relax}%

2260 \LWR@setexparray{LWR@colatspec}{3}{\relax}%

Table 7: Tabular HTML column conversions

---

<b>l, r, c:</b>	Converted to table cells without paragraph tags. Uses CSS <code>vertical-align:middle</code> so that top or bottom-aligned cells may go above or below this cell.
<b>p:</b>	Converted to table cells with paragraph tags. Ref: Table 6, $\text{\LaTeX}$ places the top line of a parbox aligned with the rest of the text line, so CSS <code>vertical-align:bottom</code> is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the CSS may be desired, changing <code>td.tdp</code> , <code>td.tdP</code> , <code>td.tdprule</code> , and <code>td.tdPrule</code> to <code>vertical-align: middle</code> . Another possibility is to change L,R,C, and P to <code>vertical-align: top</code> and not worry about the alignment of B and M cells or trying to approximate $\text{\LaTeX}$ baselines.
<b>m:</b>	With paragraph tags, CSS <code>vertical-align:middle</code> .
<b>b:</b>	With paragraph tags, CSS <code>vertical-align:top</code> so that the bottom of the text is closest to the middle of the text line.
<b>P, M, B:</b>	Horizontally-centered versions.
<b>S:</b>	Converted to 'r'. From the siunitx package.
<b>D:</b>	Converted to 'c'. From the dcolumn package.
<b>@, !, &gt;, &lt;:</b>	One each, in that order.
<b>Unknown:</b>	Converted to 'l'.
<b>\newcolumn:</b>	Currently treated as unknown.

---



```

2261 \LWR@setexparray{LWR@colbangspec}{\leftedge}{\relax}%
2262 \LWR@setexparray{LWR@colbangspec}{1}{\relax}%
2263 \LWR@setexparray{LWR@colbangspec}{2}{\relax}%
2264 \LWR@setexparray{LWR@colbangspec}{3}{\relax}%
2265 \LWR@setexparray{LWR@colbeforespec}{1}{\relax}%
2266 \LWR@setexparray{LWR@colbeforespec}{2}{\relax}%
2267 \LWR@setexparray{LWR@colbeforespec}{3}{\relax}%
2268 \LWR@setexparray{LWR@colafterspec}{1}{\relax}%
2269 \LWR@setexparray{LWR@colafterspec}{2}{\relax}%
2270 \LWR@setexparray{LWR@colafterspec}{3}{\relax}%

```

Starting at the first column specification:

```

2271 \setcounter{LWR@tablecolspos}{1}%

```

Place the colspecs string length into `\LWR@strresult`, and remember the number of characters in the column specification:

```

2272 \LWR@traceinfo{about to StrLen}%
2273 \noexpandarg%
2274 \StrLen{#1}[\LWR@strresult]%
2275 \fullexpandarg%
2276 \LWR@traceinfo{finished StrLen}%
2277 \setcounter{LWR@tablecolwidth}{\LWR@strresult}%

```

Scan through the column specifications:

```

2278 \whileof{\not\value{LWR@tablecolspos}>\value{LWR@tablecolwidth}}{%

```

Place the next single-character column type into `\LWR@strresult`:

```

2279 \noexpandarg%
2280 \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]%
2281 \fullexpandarg%

```

Not yet found a valid column type

```

2282 \boolfalse{LWR@validtablecol}%

```



Note that the parameter for a `p{spec}` column is a token list which will NOT match `l,c,r,p`.

```

2283 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{1}}{%
2284 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{%
2285 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{%
2286 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{%
2287 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
2288 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
2289 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%

```

```

2290 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
2291 \IfStrEq{\LWR@strresult}{|}{\LWR@parseskipcolumn}{}%
2292 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{%
2293 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{%
2294 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{%

```

From the dcolumn package:

```

2295 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{%

```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```

2296 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{%

```

---

Many people define centered versions “P”, “M”, and “B”:

```

\newcolumnntype{P}[1]{>{\centering\arraybackslash}p{#1}}

```

---

```

2297 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{%
2298 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{%
2299 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{%

```

If this column was an invalid column type, convert it to a p column:

```

2300 \ifbool{LWR@validtablecol}{}%
2301 \LWR@parsenormalcolumn{1}%
2302 }%
2303 \addtocounter{LWR@tablecolspos}{1}%
2304 }%
2305 }%

```

## 49.11 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.

```

2306 \newcommand*{\LWR@maybenewtablerow}
2307 {%
2308 \ifbool{LWR@startedrow}%
2309 {}% started the row
2310 {% not started the row

```

Remember that now have started the row:

```

2311 \global\booltrue{LWR@startedrow}%

```

Create the row tag, with a class if necessary.

```

2312 \global\booltrue{LWR@intabularmetadata}%
2313 \ifbool{LWR@doinghline}%
2314 {\LWR@htmltag{tr class="hline"{}}\LWR@orignewline}%
2315 {% not doing hline
2316 \ifbool{LWR@doingtbrule}%
2317 {\LWR@htmltag{tr class="tbrule"{}}\LWR@orignewline}%
2318 {\LWR@htmltag{tr}\LWR@orignewline}%
2319 }% end of not doing hline
2320 }% end of not started the row
2321 }
```

## 49.12 Data opening tag

`\LWR@tabledatasinglecolumntag` Print a table data opening tag with style for alignment

```

2322 \newcommand*{\LWR@tabledatasinglecolumntag}%
2323 {%
2324 \LWR@maybenewtablerow%
```

If have found the end of tabular command, do not create the next data cell:

```

2325 \ifbool{LWR@exitingtabular}{}%
2326 {% not exiting tabular
```

Fetch the current column's alignment character into `\LWR@strresult`:

```

2327 \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
```

print the start of a new table data cell:

```

2328 \LWR@htmltag{td class="td%
```

append this column's spec:

```

2329 \LWR@strresult%
```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag:

```

2330 \ifthenelse{\equal{\LWR@gettexpparray{LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
2331 "{}}%
```

If this is a p, m, b, or X column, allow paragraphs:

```

2332 \ifthenelse{%
```

```

2333 \equal{\LWR@strresult}{p}\OR%
2334 \equal{\LWR@strresult}{m}\OR%
2335 \equal{\LWR@strresult}{b}\OR%
2336 \equal{\LWR@strresult}{P}\OR%
2337 \equal{\LWR@strresult}{M}\OR%
2338 \equal{\LWR@strresult}{B}\OR%
2339 \equal{\LWR@strresult}{X}%
2340 }%
2341 {% allow pars
2342 \LWR@startpars%
2343 \global\booltrue{\LWR@tableparcell}%
2344 }% allow pars
2345 {}% no pars

```

Print the @ and ! contents before first column, and then the > contents:

```

2346 \ifthenelse{\cnttest{\value{\LWR@tablecolspos}}=1}%
2347 {%
2348 \LWR@getexparray{\LWR@colatspec}{\leftedge}%
2349 \LWR@getexparray{\LWR@colbangspec}{\leftedge}%
2350 }% left edge
2351 {}% not left edge
2352 \LWR@getexparray{\LWR@colbeforespec}{\the\LWR@tablecolspos}%
2353 \global\boolfalse{\LWR@intabularmetadata}%
2354 }% not exiting tabular
2355 }%

```

### 49.13 Midrules

**LWR@midrules** LWR@midrules is a data array (section 23) of columns containing Y if a midrule should be created for each column.

**Ctr LWR@midrulecounter** Indexes across the LWR@midrules data array.

```

2356 \newcounter{\LWR@midrulecounter}

```

**\LWR@clearmidrules** Start new midrules. Called at beginning of tabular and also at \.

Clears all LWR@midrules markers for this line.

```

2357 \newcommand*{\LWR@clearmidrules}
2358 {%
2359 \setcounter{\LWR@midrulecounter}{1}%
2360 \whiledo{%
2361 \cnttest{\value{\LWR@midrulecounter}}{<=}{\value{\LWR@tablecolwidth}}%
2362 }%
2363 {%

```

```

2364 \LWR@setexpparray{LWR@midrules}{\theLWR@midrulecounter}{\relax}%
2365 \addtocounter{LWR@midrulecounter}{1}%
2366 }%
2367 }

```

`\LWR@subcmidrule` [*width*] [*trim*] [*leftcolumn*] [*rightcolumn*]

Marks LWR@midrules data array elements to be “Y” from left to right columns.

```

2368 \newcommand*{\LWR@subcmidrule}[4]{%
2369 \setcounter{LWR@midrulecounter}{#3}%
2370 \whiledo{\cnttest{\value{LWR@midrulecounter}}{<=}{#4}}%
2371 {%
2372 \LWR@setexpparray{LWR@midrules}{\theLWR@midrulecounter}{Y}%
2373 \addtocounter{LWR@midrulecounter}{1}%
2374 }% end of the whiledo
2375 }

```

`\LWR@docmidrule` [*width*] [*trim*] [*leftcolumn-rightcolumn*]

Marks LWR@midrules array elements to be “Y” from left to right columns.

```

2376 \NewDocumentCommand{\LWR@docmidrule}{o d() >{\SplitArgument{1}{-}}m}%
2377 {\LWR@subcmidrule{#1}{#2}{#3}}

```

## 49.14 Multicolumns

### 49.14.1 Parsing multicolumns

```

2378 \newcounter{LWR@tablemulticolwidth}
2379 \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`.

```

2380 \newcommand*{\LWR@printmccoltype}[1]{%
2381 \LWR@traceinfo{lw@printmccoltype -#1-}%

```

Get one token of the column spec:

```

2382 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%

```

Add to the HTML tag depending on which column type is found:

```

2383 \IfStrEq{\LWR@strresult}{l}{l}{}%
2384 \IfStrEq{\LWR@strresult}{c}{c}{}%
2385 \IfStrEq{\LWR@strresult}{r}{r}{}%
2386 \IfStrEq{\LWR@strresult}{p}{p}{}%
2387 \IfStrEq{\LWR@strresult}{m}{m}{}%
2388 \IfStrEq{\LWR@strresult}{b}{b}{}%
2389 \IfStrEq{\LWR@strresult}{P}{P}{}%
2390 \IfStrEq{\LWR@strresult}{M}{M}{}%
2391 \IfStrEq{\LWR@strresult}{B}{B}{}%
2392 \IfStrEq{\LWR@strresult}{S}{r}{}%
2393 \IfStrEq{\LWR@strresult}{X}{p}{}%
2394 \LWR@traceinfo{lwarp@printmccoltype done}%
2395 }

```

`\LWR@multicolpartext` Print the data with paragraph tags:

```

2396 \newcommand*{\LWR@multicolpartext}{%
2397 \LWR@startpars%
2398 \LWR@multicoltext%
2399 \LWR@stoppars%
2400 }

```

`\LWR@multicolother`  $\{ \langle colspec \rangle \}$  For @, >, <, print the next token without paragraph tags:

```

2401 \newcommand*{\LWR@multicolother}[1]{%
2402 \addtocounter{LWR@tablemulticolspos}{1}%
2403 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
2404 \LWR@strresult%

```

A valid column data type was found:

```

2405 \booltrue{LWR@validtablecol}%
2406 }

```

`\LWR@multicolskip` Nothing to print for this column type.

```

2407 \newcommand*{\LWR@multicolskip}{%

```

A valid column data type was found:

```

2408 \booltrue{LWR@validtablecol}%
2409 }

```

`\LWR@printmccoldata`  $\{ \langle colspec \rangle \}$  Print the data for any valid column type found.

```

2410 \newcommand*{\LWR@printmccoldata}[1]{%
2411 \LWR@traceinfo{lwarp@printmccoldata -#1}%

```

Not yet found a valid column type:

```
2412 \boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
2413 \StrChar{#1}{\theLWR@tablemulticolspos}[LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
2414 \IfStrEq{LWR@strresult}{l}{LWR@multicoltext}{}%
2415 \IfStrEq{LWR@strresult}{c}{LWR@multicoltext}{}%
2416 \IfStrEq{LWR@strresult}{r}{LWR@multicoltext}{}%
2417 \IfStrEq{LWR@strresult}{D}{}%
2418 \addtocounter{LWR@tablemulticolspos}{3}% skip parameters
2419 LWR@multicoltext%
2420 }{}%
2421 \IfStrEq{LWR@strresult}{p}{LWR@multicolparttext}{}%
2422 \IfStrEq{LWR@strresult}{m}{LWR@multicolparttext}{}%
2423 \IfStrEq{LWR@strresult}{b}{LWR@multicolparttext}{}%
2424 \IfStrEq{LWR@strresult}{P}{LWR@multicolparttext}{}%
2425 \IfStrEq{LWR@strresult}{M}{LWR@multicolparttext}{}%
2426 \IfStrEq{LWR@strresult}{B}{LWR@multicolparttext}{}%
2427 \IfStrEq{LWR@strresult}{S}{LWR@multicolparttext}{}%
2428 \IfStrEq{LWR@strresult}{X}{LWR@multicolparttext}{}%
2429 \IfStrEq{LWR@strresult}{|}{LWR@multicolskip}{}%
2430 \IfStrEq{LWR@strresult}{\detokenize@}{LWR@multicolother{#1}}{}%
2431 \IfStrEq{LWR@strresult}{\detokenize!}{LWR@multicolother{#1}}{}%
2432 \IfStrEq{LWR@strresult}{\detokenize>}{LWR@multicolother{#1}}{}%
2433 \IfStrEq{LWR@strresult}{\detokenize<}{LWR@multicolother{#1}}{}
```

If an invalid column type:

```
2434 \ifbool{LWR@validtablecol}{LWR@multicoltext}%
```

Tracing:

```
2435 LWR@traceinfo{lwarp@printmccoldata done}%
2436 }
```

```
\parsemulticolumnalignment {<1: colspec>} {<2: printresults>}
```

Scan the multicolumn specification and execute the printfunction for each entry.

Note that the spec for a p{spec} column, or @, >, <, is a token list which will NOT match l, c, r, or p.

```

2437 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
2438 \setcounter{LWR@tablemulticolspos}{1}%
2439 \StrLen{#1}[\LWR@strresult]%
2440 \setcounter{LWR@tablemulticolwidth}{\LWR@strresult}%

```

Scan across the tokens in the column spec:

```

2441 \whiledo{%
2442 \not\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolwidth}%
2443 }%
2444 {%

```

Execute the assigned print function for each token in the column spec:

```

2445 #2{#1}%

```

Move to the next token in the column spec:

```

2446 \addtocounter{LWR@tablemulticolspos}{1}%
2447 }%
2448 }

```

#### 49.14.2 High-level multicolumn interface

```
\LWR@domulticolumn {<1: numcols>} {<2: colspec>} {<3: text>}
```

```

2449 \newcommand{\LWR@multicoltext}{}
2450
2451 \NewDocumentCommand{\LWR@domulticolumn}{m m +m}{%
2452 \LWR@traceinfo{lwrdomulticolumn -#1- -#2-}%

```

Remember the text to be inserted, and remember that a valid column type was found:

```

2453 \renewcommand{\LWR@multicoltext}{%
2454 #3%
2455 \booltrue{LWR@validtablecol}%
2456 }%

```

Row processing:

```

2457 \LWR@maybe newtable row%

```

Begin the opening table data tag:

```

2458 \LWR@htmltag{td colspan="#1"
2459 class="td%

```



Print the column type:

```
2460 \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”.

```
2461 \ifthenelse{\equal{\LWR@gettexpparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
```

Close the class tag’s opening quote:

```
2462 "%
2463 }% end of the opening table data tag
2464 \global\boolfalse{\LWR@intabularmetadata}%
2465 \LWR@parsemulticolumnalignment{#2}{\LWR@printmccoldata}%
2466 }
```

### 49.14.3 Longtable captions

Bool `LWR@starredlongtable` Per the caption package, step the counter if `longtable*`.

```
2467 \newbool{\LWR@starredlongtable}
2468 \boolfalse{\LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
2469 \providecommand*{\LTcapttype}{table}
```

`\LWR@longtabledatacaptiontag` \* [*toc entry*] {*caption*}

```
2470 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
2471 {%
```

Remember the latest name for `\nameref`:

```
2472 \IfValueTF{#2}{% optional given?
2473 \ifthenelse{\equal{#2}{}}{% optional empty?
2474 {\LWR@setlatestname{#3}}% empty
2475 {\LWR@setlatestname{#2}}% given and non-empty
2476 }% optional given
2477 {\LWR@setlatestname{#3}}% no optional
```

create a multicolumn across all the columns

```
2478 \LWR@domulticolumn{\theLWR@tabletotalcols}{P}{% \LWR@domulticolumn
```

```

2479 % \IfBooleanTF{#1}% star?
2480 % {\IfValueTF{#2}{\LWR@origcaption*{#2}{#3}}{\LWR@origcaption*{#3}}}
2481 % {\IfValueTF{#2}{\LWR@origcaption[2]{#3}}{\LWR@origcaption{#3}}}
2482 \IfBooleanTF{#1}% star?

```

Star version, show a caption but do not make a LOT entry:

```

2483 {% yes star
2484 \LWR@htmlblocktag{figcaption}%
2485 #3%
2486 \LWR@htmlblocktag{/figcaption}%
2487 }%
2488 {% No star:

```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```

2489 \ifbool{LWR@starredlongtable}%
2490 {%
2491 \ifthenelse{\equal{#2}{}}% TOC entry
2492 {}%
2493 {%
2494 \refstepcounter{\LTcapttype}%
2495 \protected@edef\@currentlabel{%
2496 \csuse{p@\LTcapttype}\csuse{the\LTcapttype}}%
2497 }%
2498 }{}%

```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```

2499 \LWR@htmlblocktag{figcaption}%
2500 \csuse{fnum@\LTcapttype}\CaptionSeparator#3%
2501 \LWR@htmlblocktag{/figcaption}%

```

See if an optional caption was given:

```

2502 \ifthenelse{\equal{#2}{}}% TOC entry empty

```

if the optional caption was given, but empty, do not form a TOC entry

```

2503 {}%

```

If the optional caption was given, but might only be []:

```

2504 {% TOC entry not empty
2505 \IfNoValueTF{#2}% No TOC entry?

```

The optional caption is []:

```

2506 {% No TOC entry
2507 \addcontentsline%
2508 {\csuse{ext@LTcapttype}}%
2509 {\LTcapttype}%
2510 {%
2511 \protect\numberline%
2512 {\csuse{p@LTcapttype}\csuse{theLTcapttype}}%
2513 {\ignorespaces #3\protect\relax}%
2514 }%
2515 }% end of No TOC entry

```

The optional caption has text enclosed:

```

2516 {% yes TOC entry
2517 \addcontentsline%
2518 {\csuse{ext@LTcapttype}}%
2519 {\LTcapttype}%
2520 {%
2521 \protect\numberline%
2522 {\csuse{p@LTcapttype}\csuse{theLTcapttype}}%
2523 {\ignorespaces #2\protect\relax}%
2524 }%
2525 }% end of yes TOC entry
2526 }% end of TOC entry not empty
2527 }% end of no star
2528 }% end of \LWR@domulticolumn
2529
2530 \addtocounter{LWR@tablecolspos}{\theLWR@tabletotalcols}
2531 \addtocounter{LWR@tablecolspos}{-1}
2532
2533 }

```

#### 49.14.4 \tabledatamulticolumntag

`\LWR@tabledatamulticolumntag`  $\{\langle numcols \rangle\}$   $\{\langle alignment \rangle\}$   $\{\langle text \rangle\}$

```

2534 \NewDocumentCommand{\LWR@tabledatamulticolumntag}{m m +m}%
2535 {%
2536 \LWR@domulticolumn{#1}{#2}{#3}%
2537 \addtocounter{LWR@tablecolspos}{#1}%
2538 \addtocounter{LWR@tablecolspos}{-1}%
2539 }

```

## 49.15 Multirow

Pkg multirow

```
\LWR@tabledatamultirowtag {<numrows>} [<bigstruts>] {<width>} [<fixup>] {<text>}
```

```
2540 \NewDocumentCommand{\LWR@tabledatamultirowtag}{m o m o m}%
```

```
2541 {%
```

```
2542 \LWR@maybenewtablerow%
```

Print the start of a new table data cell:

```
2543 \LWR@htmltag{td rowspan="#1" class="td%
```

Append this column's spec:

```
2544 \StrChar{\LWR@tablecolspec}{\the\LWR@tablecolspos}%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag:

```
2545 \ifthenelse{\equal{\LWR@gettexpparray{\LWR@midrules}{\the\LWR@tablecolspos}}{Y}}{rule}{}%
```

```
2546 "{}}
```

While printing the text, redefine \\\ to generate a new line

```
2547 \begingroup \let\\\LWR@endoffline #5 \endgroup
```

```
2548 \LWR@stoppars%
```

```
2549 \global\boolfalse{\LWR@intabularmetadata}%
```

```
2550 }%
```

## 49.16 Utility macros inside a table

```
2551 \newcommand*{\LWR@donothing}{}%
```

```
2552 \newcommand*{\LWR@domidrule}{\booltrue{\LWR@doinghline}}%
```

```
2553 \newcommand*{\LWR@dotbrule}{\booltrue{\LWR@doingtbrule}}%
```

## 49.17 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:

```
2554 \newbool{\LWR@exitingtabular}
```

```
2555 \newcommand*{\LWR@tabledatacolumnntag}%
```

```
2556 {%
```

`\show\LWR@mynexttoken` to see what tokens to look for

If not any of the below, start a new table cell:

```
2557 \let\mynext\LWR@tabledatasinglecolumnntag%
```

If exiting the tabular:

```
2568 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\end}}{%
2569 {\booltrue\LWR@exitingtabular}}{ }%
```

`longtable` can have a caption in a cell

```
2560 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\caption}}{%
2561 {\let\mynext\LWR@donothing}}{ }%
```

Look for other things which would not start a table cell:

```
2562 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multicolumn}}{%
2563 {\let\mynext\LWR@donothing}}{ }%
2564 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multirow}}{%
2565 {\let\mynext\LWR@donothing}}{ }%
```

if come to an `\mrowcell`, this is a cell to be skipped over

```
2566 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\mrowcell}}{%
2567 {\let\mynext\LWR@donothing}}{ }%
2568 %
2569 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\hline}}{%
2570 {\let\mynext\LWR@donothing}}{ }%
2571 %
2572 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\toprule}}{%
2573 { }%
2574 \let\mynext\LWR@donothing}}{ }%
2575 %
2576 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\midrule}}{%
2577 {\let\mynext\LWR@donothing}}{ }%
2578 %
2579 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cmidrule}}{%
2580 {\let\mynext\LWR@donothing}}{ }%
2581 %
2582 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cline}}{%
2583 {\let\mynext\LWR@donothing}}{ }%
2584 %
2585 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\bottomrule}}{%
2586 {\let\mynext\LWR@donothing}}{ }%
2587 %
2588 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpprintonly}}
```

```

2589 {\let\mynext\LWR@donothing}{}%
2590 %
2591 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpHTMLonly}}
2592 {\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:


```

2593 \mynext%
2594 }

2595 \end{warpHTML}

```

## 49.18 `\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:**

```

2596 \begin{warpall}
2597 \newcommand*{\mrowcell}{}
2598 \end{warpall}

```

## 49.19 New `\tabular` definition

**for HTML output:**

```

2599 \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.

```

2600 \newenvironment*{LWR@tabular}[2][]
2601 {%
2602 \LWR@traceinfo{tabular started}%
2603 \begingroup%
2604 \addtocounter{LWR@tabulardepth}{1}%

```

Not yet started a table row:

```

2605 \global\boolfalse{LWR@startedrow}%

```

Not yet doing an hline:

```
2606 \global\boolfalse{LWR@doinghline}%
```

Not yet doing a top/bottom rule:

```
2607 \global\boolfalse{LWR@doingtbrule}%
```

Have not yet found the end of tabular command:

```
2608 \boolfalse{LWR@exitingtabular}%
```

Create the `table` tag:

```
2609 \global\booltrue{LWR@intabularmetadata}%
```

```
2610 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
2611 \LWR@parsetablecols{#2}%
```

Table col spec is: `\LWR@tablecolspec` which is a string of `llccrr`, etc.

Do not place the table inside a paragraph:

```
2612 \LWR@stoppars%
```

Track column `#` for setting text-align:

```
2613 \setcounter{LWR@tablecolspos}{1}%
```

Start looking for midrules:

```
2614 \LWR@clearmidrules%
```

`\` becomes a macro to end the table row:

```
2615 \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:

```
2616 \renewcommand*{\hline}{\LWR@domidrule\LWR@getmynexttoken}%
```

```
2617 \newcommand*{\midrule}{\LWR@domidrule\LWR@getmynexttoken}%
```

```
2618 \NewDocumentCommand{\cmidrule}{o d() m}%
```

```
2619 {\LWR@docmidrule[##1](##2){##3}\LWR@getmynexttoken}%
```

```
2620 \RenewDocumentCommand{\cline}{m}%
```

```

2621 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%
2622 \newcommand*{\toprule}{\LWR@dotbrule\LWR@getmynexttoken}%
2623 \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`.

```

2624 \let\multicolumn\LWR@tabledatamulticolumntag%
2625 \let\multirow\LWR@tabledatamultirowtag%
2626 \renewcommand*{\mrowcell}{\global\booltrue{\LWR@skippingmrowcell}}%
2627 \let\caption\LWR@longtabledatacaptiontag%

```

Reset for new processing:

```

2628 \global\boolfalse{\LWR@tableparcell}%
2629 \global\boolfalse{\LWR@skippingmrowcell}%

```

Look ahead for a possible table data cell:

```

2630 \LWR@getmynexttoken%
2631 }%

```

Ending the environment:

```

2632 {%
2633 \LWR@closetabledatacell%
2634 \LWR@htmlblocktag{/tr}%
2635 \LWR@htmlblocktag{/table}%
2636 \global\boolfalse{\LWR@intabularmetadata}%

2637 \addtocounter{\LWR@tabulardepth}{-1}%
2638 \endgroup%
2639 }

2640 \end{warpHTML}

```

## 49.20 Array

Pkg `array`

`array` is also automatically loaded by `siunitx`.



## 50 Cross-references

Sectioning commands have been emulated from scratch, so the cross-referencing commands are custom-written for them. Emulating both avoids several layers of patches.

The `zref` package is used to remember section name, file, and lateximage depth and number for each label.

Table 8 shows the data structures related to cross-referencing.

for HTML output: 2641 `\begin{warpHTML}`

### 50.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

```
2642 \newcommand*{\@currentlabelname}{}
```

`\LWR@stripperperiod` `{\text}` `[{.}]`

Removes a trailing period.

```
2643 \def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%
```

`\LWR@setlatestname` `{\object name}`

Removes `\label`, strips any final period, and remembers the result.

```
2644 \newcommand*{\LWR@setlatestname}[1]{%
```

Remove `\label` and other commands from the name, the strip any final period.  
See `zref-titleref` and `getttitlestring`.

```
2645 \GetTitleStringExpand{#1}%
2646 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
2647 \edef\@currentlabelname{%
2648 \expandafter\LWR@stripperperiod\@currentlabelname%
2649 \ltx@empty.\ltx@empty\@nil%
2650 }%
2651 }
```

Table 8: 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 50.3), plus \slabel data (section 50.2): <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 50.4.	
<b>\ref and \nameref:</b> Adds HTML tags. See section 50.4.	
<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 35.4 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 59.	
<b>\label inside math:</b> See section 54.4.1.	
<b>Footnotes:</b> See \noteentry in section 35.4.	

---

## 50.2 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:

```
2652 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
2653 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
2654 \zref@newprop{zLWR@htmlfilenumber}{%
2655 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilenumber}}%
2656 }%
```

Additional properties for lateximages:

```
2657 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
2658 \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:

```
2659 \zref@addprop{special}{zLWR@name}
2660 \zref@addprop{special}{zLWR@htmlfilenumber}
2661 \zref@addprop{special}{zLWR@lateximagedepth}
2662 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
2663 \newcommand*{\LWR@spreff}[2]{%
2664 \zref@extractdefault{#1}{#2}{??}}
```

`\LWR@nameref`  $\{\langle label \rangle\}$  Returns the section name for this label:

```
2665 \newcommand*{\LWR@nameref}[1]{%
2666 \LWR@spreff{#1}{zLWR@name}%
2667 }
```

`\LWR@htmlfileref`  $\{\langle label \rangle\}$  Returns the file number for this label:

```

2668 \newcommand*\LWR@htmlfileref}[1]{%
2669 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
2670 \LWR@sprep{#1}{zLWR@htmlfilenumber}%
2671 }

```

`\LWR@lateximagedepthref`  $\{\langle label \rangle\}$  Returns the `lateximagedepth` for this label:

```

2672 \newcommand*\LWR@lateximagedepthref}[1]{%
2673 \LWR@sprep{#1}{zLWR@lateximagedepth}%
2674 }

```

`\LWR@lateximagenumberref`  $\{\langle label \rangle\}$  Returns the `lateximagenumber` for this label:

```

2675 \newcommand*\LWR@lateximagenumberref}[1]{%
2676 \LWR@sprep{#1}{zLWR@lateximagenumber}%
2677 }

```

`\LWR@splabel`  $\{\langle label \rangle\}$  Sanitize the name and then creates the label:

```

2678 \newcommand*\LWR@splabel}[1]{%
2679 \LWR@setlatestname{\@currentlabelname}%
2680 \zref@labelbylist{#1}{special}}

```

### 50.3 Labels

`\LWR@subsublabel`  $\{\langle label \rangle\}$  Creates an HTML id tag.

```

2681 \newcommand*\LWR@subsublabel}[1]{%

```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```

2682 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{%
2683 {}%
2684 {% not lateximage

```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```

2685 \ifbool{LWR@doingstartpars}%
2686 {% pars allowed
2687 \ifbool{LWR@doingapar}
2688 {% par started
2689 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
2690 }% par started
2691 {% par not started

```

```

2692 \LWR@stoppars%
2693 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
2694 \LWR@startpars%
2695 }% par not started
2696 }% pars allowed
2697 {% pars not allowed
2698 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
2699 }% pars not allowed
2700 }% not lateximage
2701 }

```

`\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.

```

2702 \NewDocumentCommand{\LWR@newlabel}{m o}{%
2703 \LWR@traceinfo{\LWR@newlabel: starting}%
2704 \LWR@traceinfo{\LWR@newlabel: !#1!}%
2705 % \@bsphack%

```

Create a traditional L<sup>A</sup>T<sub>E</sub>X label, as modified by `cleveref`:

```

2706 \LWR@origlabel{#1}%

```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```

2707 \LWR@traceinfo{\LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
2708 \LWR@traceinfo{\LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
2709 \LWR@traceinfo{\LWR@newlabel: LWR@htmlfilenumber is \theLWR@htmlfilenumber}%
2710 \LWR@splabel{#1}%
2711 \LWR@subsublabel{#1}%
2712 % \@esphack%
2713 \LWR@traceinfo{\LWR@newlabel: done}%
2714 }

```

## 50.4 References

`\LWR@startref`  $\{ \langle label \rangle \}$  (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, `#` character, and a label.

```

2715 \newcommand*{\LWR@startref}[1]
2716 {%
2717 \edef\LWR@lhref{\LWR@lateximagedepthref{#1}}
2718 \LWR@traceinfo{\LWR@startref A: !#1!}%

```

Create the filename part of the link:

```

2719 \LWR@htmltag{a href="%
2720 \LWR@traceinfo{\LWR@startref B}%
2721 \LWR@htmlrefsectionfilename{#1}%
2722 \LWR@traceinfo{\LWR@startref C}%
2723 \#%

```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```

2724 \LWR@traceinfo{\LWR@startref D: !#1!}%
2725 \ifthenelse{\equal{\LWR@lhref}{??}}%

```

“??” if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:

```

2726 {%
2727 \LWR@traceinfo{\LWR@startref D0: ??}
2728 ??}%

```

If `LWR@lateximagedepth` is known. Use a `lateximage` if the depth is greater than zero, or a regular link otherwise:

```

2729 {%
2730 \LWR@traceinfo{\LWR@startref D1: \LWR@lhref}%
2731 \ifthenelse{\cnttest{\LWR@lhref}{>}{0}}%
2732 {%
2733 \LWR@traceinfo{\LWR@startref D2: \LWR@lhref}%
2734 lateximage\LWR@lateximagenumberref{#1}%
2735 }%
2736 {%
2737 \LWR@traceinfo{\LWR@startref D3}%
2738 #1%
2739 }%
2740 }%
2741 \LWR@traceinfo{\LWR@startref E}%

```

Closing quote:

```

2742 "}}%
2743 \LWR@traceinfo{\LWR@startref F}%
2744 }

```

`\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)”.

```
2745 \NewDocumentCommand{\LWR@subnewref}{m m}{%
2746 \LWR@traceinfo{\LWR@subnewref #1 #2}%
2747 \LWR@startref{#1}%
2748 \LWR@origref{#2}%
2749 \LWR@htmltag{/a}%
2750 }
```

`\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.

```
2751 \NewDocumentCommand{\LWR@newref}{s m}{%
2752 \LWR@traceinfo{\LWR@newref #2}%
2753 \IfBooleanTF{#1}%
2754 {\LWR@origref{#2}}%
2755 {\LWR@subnewref{#2}{#2}}%
2756 }
```

`\pagerefPageFor` Text for starred page references.

```
2757 \newcommand*{\pagerefPageFor}{see }
```

`\pageref` \*  $\{\langle label \rangle\}$  Create an internal document reference, or just the unlinked number if starred, per hyperref.

```
2758 \NewDocumentCommand{\LWR@newpageref}{s m}{%
2759 \IfBooleanTF{#1}%
2760 {(\pagerefPageFor\LWR@origref{#2})}%
2761 {(\cpageref{#2})}%
2762 }
```

`\nameref`  $\{\langle label \rangle\}$

```
2763 \newcommand*{\nameref}[1]{%
2764 \LWR@traceinfo{\nameref A}%
2765 \LWR@startref{#1}%
2766 \LWR@traceinfo{\nameref B}%
2767 \LWR@nameref{#1}%
2768 \LWR@traceinfo{\nameref C}%
2769 \LWR@htmltag{/a}%

```

```
2770 \LWR@traceinfo{nameref D}%
2771 }
```

`\Nameref {<label>}` In print, adds the page number. In HTML, does not.

```
2772 \let\Nameref\nameref
```

## 50.5 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.

```
2773 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF:
2774 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Create a link with a text name:

```
\LWR@subhyperref {<URL>} {<text>}
```

```
2775 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
2776 \LWR@htmltag{a href="#1" target="_{blank}\LWR@orignewline}#2\LWR@htmltag{/a}%
2777 \LWR@ensuredoingapar%
2778 }
```

```
\LWR@subhyperrefclass {<URL>} {<text>} {<htmlclass>}
```

```
2779 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
2780 \LWR@htmltag{a href="#1"
2781 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
2782 \LWR@ensuredoingapar%
2783 }
```

```
\href [<options>] {<URL>} {<text>}
```



Create a link with accompanying text:

```

2784 \NewDocumentCommand{\LWR@hrefb}{0{} m +m}{%
2785 \LWR@subhyperref{#2}{#3}%
2786 \endgroup%
2787 \LWR@ensuredoingapar%
2788 }
2789
2790 \newcommand{\href}{%
2791 \LWR@ensuredoingapar%
2792 \begingroup%
2793 \catcode'\_ =12
2794 \LWR@hrefb%
2795 }

```

`\nolinkurl`  $\{\langle URL \rangle\}$

Print the name of the link without creating the link:

```

2796 \newcommand*{\LWR@nolinkurlb}[1]{#1\endgroup\LWR@ensuredoingapar}
2797
2798 \newcommand{\nolinkurl}{%
2799 \LWR@ensuredoingapar%
2800 \begingroup\catcode'\_ =12
2801 \LWR@nolinkurlb%
2802 }

```

`\url`  $\{\langle URL \rangle\}$

Create a link whose text name is the address of the link:

```

2803 \newcommand*{\LWR@urlb}[1]{%
2804 \href{#1}{#1}%
2805 \endgroup%
2806 \LWR@ensuredoingapar%
2807 }
2808
2809 \newcommand{\url}{%
2810 \LWR@ensuredoingapar%
2811 \begingroup\catcode'\_ =12
2812 \LWR@urlb%
2813 }

```

`\LWR@subinlineimage`  $[\langle alttag \rangle] \{\langle class \rangle\} \{\langle filename \rangle\} \{\langle extension \rangle\} \{\langle style \rangle\}$

```

2814 \newcommand*{\LWR@subinlineimage}[5][{}]{%
2815 \ifthenelse{\equal{#1}{}}{
2816 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"}}}{%

```

---

```
2817 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{}%  
2818 }  
  
2819 \end{warpHTML}
```

Table 9: Float data structures

---

For each `<type>` of float (figure, table, etc.) there exists the following:

---

**counter `<type>`:** A counter called `<type>`, such as `figure`, `table`.

**`\<type>name`:** Name. `\figurename` prints “Figure”, etc.

**`\ext@<type>`:** File extension. `\ext@figure` prints “lof”, etc.

**`\fps@<type>`:** Placement.

**`\the<type>`:** Number. `\thetable` prints the number of the table, etc.

**`\p@<type>`:** Parent’s number. Prints the number of the [within] figure, etc.

**`\fnum@<type>`:** Prints the figure number for the caption.

`\<type>name \the<type>`, “Figure 123”.

**`\<type>`:** Starts the float environment. `\figure` or `\begin{figure}`

**`\end<type>`:** Ends the float environment. `\endfigure` or `\end{figure}`

**`\tf@<ext>`:** The L<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.

---

## 51 Floats

Floats are supported, although partially through emulation.

Table 9 shows the data structure associated with each `<type>` of float.

`\@makecaption` is redefined to print the float number and caption text, separated by `\CaptionSeparator`, which works with the `babel` package to adjust the caption separator according to the language. French, for example, uses an en-dash instead of a colon: “Figure 123 – Caption text”.

## 51.1 Float captions

for HTML output: 2820 \begin{warpHTML}

\LWR@floatbegin {<type>} [<placement>]

Begins a \newfloat environment.

```
2821 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
2822 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
2823 \addtocounter{LWR@thisfloat}{1}%
2824 \booltrue{LWR@freezethisfloat}%
```

```
2825 \begingroup
```

Settings while inside the environment:

```
2826 \LWR@origraggedright
```

Open an HTML figure tag:

```
2827 \LWR@htmltag{figure id="autofloat-\arabic{LWR@thisfloat}" class="#1"}
```

```
2828 \renewcommand*{\@capttype}{#1}
2829 \caption@settype{#1}
2830 \LWR@startpars
2831 }
```

\@float Support packages which create floats directly.  
 \@dblfloat

```
2832 \let\@float\LWR@floatbegin
2833 \let\@dblfloat\LWR@floatbegin
```

\LWR@floatend Ends a \newfloat environment.

```
2834 \newcommand*{\LWR@floatend}{%
2835 \LWR@stoppars%
2836 \LWR@htmllementend{figure}%
2837 \endgroup%
2838 \boolfalse{LWR@freezethisfloat}%
2839 \LWR@startpars%
2840 }
```

`\end@float` Support packages which create floats directly.  
`\end@dblfloat`

2841 `\let\end@float\LWR@floatend`  
 2842 `\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.

2843 `\newcounter{LWR@thisfloat}`

**Bool** `LWR@freezethisfloat` Prevents multiple increments of `\LWR@thisfloat` inside a float.

2844 `\newbool{LWR@freezethisfloat}`  
 2845 `\boolfalse{LWR@freezethisfloat}`

`\LWR@maybeinthisfloat`

2846 `\newcommand*{\LWR@maybeinthisfloat}{%`  
 2847 `\ifbool{LWR@freezethisfloat}{\addtocounter{LWR@thisfloat}{1}}%`  
 2848 `}`

`\@capttype` Remembers which float type is in use.

2849 `\newcommand*{\@capttype}{}`

### 51.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

2850 `\AtBeginDocument{\providecommand*\CaptionSeparator}{:~}}`

`\@makecaption` `{\langle name and num \rangle}{\langle text \rangle}`

Prints the float type and number, the caption separator, and the caption text.

2851 `\AtBeginDocument{\renewcommand{\@makecaption}[2]{\#1\CaptionSeparator\#2}}`

### 51.1.2 Caption and LOF linking and tracking

When a new HTML file is marked in the  $\text{\LaTeX}$  PDF file, the  $\text{\LaTeX}$  page number at that point is stored in `LWR@latestautopage`, (and the associated filename is remembered by the special  $\text{\LaTeX}$  labels). This page number is used to generate an `autofloat` HTML ID in the HTML output at the start of the new HTML file.

Meanwhile, there is a float counter used to generate an HTML `autofloat` IDs 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.

```
2852 \newcounter{LWR@nextautofloat}
2853 \newcounter{LWR@nextautopage}
```

`\LWRsetnextfloat`  $\{\langle autopage \rangle\} \{\langle autofloat \rangle\}$

This is written to the `.lof` file just before each float's usual entry. The `autopage` and `autofloat` are remembered for `\l@figure` to use when creating the HTML links.

```
2854 \newcommand*{\LWRsetnextfloat}[2]{%
2855 \setcounter{LWR@nextautopage}{#1}%
2856 \setcounter{LWR@nextautofloat}{#2}%
2857 }
```

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.

```
2858 \newcounter{LWR@latestautopage}% updated each new HTML file
2859 \setcounter{LWR@latestautopage}{1}
```

```
2860 \let\LWR@origcaption@begin\caption@begin
2861 \let\LWR@origcaption@end\caption@end
2862 \let\LWR@orig@@par\@@par
```

`\LWR@caption@begin` Low-level patches to create HTML tags for captions.

```
2863 \newcommand{\LWR@caption@begin}
2864 {
2865 \LWR@traceinfo{LWR@caption@begin}%
```

Keep `par` and `minipage` changes local:

```
2866 \begingroup%
```

The `caption` code was not allowing the closing `par` tag:

```
2867 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a `minipage` or `\parbox` inside the caption:

```
2868 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
2869 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
2870 \LWR@htmlblocktag{figcaption}%
2871 \LWR@origcaption@begin%
2872 }
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
2873 \newcommand{\LWR@caption@end}
2874 {%
2875 \LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
2876 \vspace*{\baselineskip}%
```

Closing tag:

```
2877 \LWR@htmlblocktag{/figcaption}%
2878 \endgroup%
2879 % \leavevmode% avoid bad space factor (0) error
2880 \LWR@traceinfo{\LWR@caption@end: done}%
2881 }
```

`\caption@begin` Low-level patches to create HTML tags for captions.

```
\caption@end
2882 \AtBeginDocument{
2883 \let\caption@begin\LWR@caption@begin
2884 \let\caption@end\LWR@caption@end
2885 }
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```
2886 \let\LWR@origcaptionlistentry\captionlistentry
2887
2888 \renewcommand*{\captionlistentry}{%
2889 \LWR@maybeinthisfloat%
2890 \LWR@ensuredoingapar%
2891 \LWR@htmltag{a id="autofloat-\arabic{\LWR@thisfloat}"{}}\LWR@htmltag{/a}%
2892 \LWR@origcaptionlistentry%
2893 }
```

```

2894
2895 \def\LWR@LTcaptionlistentry{%
2896 \LWR@ensuredoingapar%
2897 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
2898 \bgroup
2899 \@ifstar{\egroup\LWR@LT@captionlistentry}% gobble *
2900 {\egroup\LWR@LT@captionlistentry}}%
2901 \def\LWR@LT@captionlistentry#1{%
2902 \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.

```

2903 \let\LWR@origaddcontentsline\addcontentsline
2904
2905 \renewcommand*{\addcontentsline}[3]{%
2906 \ifthenelse{\equal{#1}{toc}}{ }{%
2907 \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{ }
2908 \addtocontents{\@nameuse{ext@#2}}{ }%
2909 \protect\LWR@setnextfloat%
2910 {\arabic{LWR@latestautopage}}%
2911 {\arabic{LWR@thisfloat}}%
2912 }% addtocontents
2913 }% not toc
2914 \LWR@origaddcontentsline{#1}{#2}{#3}%
2915 }

```

`\captionof` Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

2916 \AtBeginDocument{
2917 \let\LWR@origcaptionof\captionof
2918
2919 \renewcommand*{\captionof}{%
2920 \LWR@maybeinthisfloat%
2921 \LWR@stoppars
2922 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
2923 \LWR@origcaptionof%
2924 }
2925 }

2926 \end{warpHTML}

```



## 52 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: 2927 `\begin{warpHTML}`

### 52.1 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.

```
2928 \newcommand*{\LWR@myshorttoc}[1]{
2929 \LWR@ensuredoingapar
```

Only if the file exists:

```
2930 \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.



disabled

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`.

```
2931 \begingroup
2932 % \ifxetexorluatex%
2933 % \else
2934 % \inputencoding{latin1}% currently disabled
2935 % \fi
2936 \makeatletter
```

Read in the TOC file:

```
2937 \@input{\jobname.#1}
2938 % \makeatother
2939 \endgroup
2940 }%
2941 {}%
2942 }
```

`\LWR@subtableofcontents`  $\{\langle toc/lof/lot \rangle\} \{\langle sectionstarname \rangle\}$

Places a TOC/LOF/LOT at the current position.

```
2943 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```
2944 \@ifundefined{chapter}
2945 {\LWR@closeprevious{\LWR@depthsection}}
2946 {\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
2947 \LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
2948 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}
```

Create a new HTML nav containing the TOC/LOF/LOT:

```
2949 \LWR@htmlelementclass{nav}{#1}
```

Create the actual list:

```
2950 \LWR@myshorttoc{#1}
```

Close the nav:

```
2951 \LWR@htmlelementclassend{nav}{#1}
2952 }
```

Patch `\@starttoc` to encapsulate the TOC inside HTML tags:

```
2953 \let\LWR@orig@starttoc\@starttoc
2954
2955 \renewcommand{\@starttoc}[1]{
```

```

2956 \LWR@html elementclass{nav}{#1}
2957 \LWR@orig@starttoc{#1}
2958 \LWR@html elementclassend{nav}{#1}
2959 }

```

Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.

```

2960 \let\LWR@origtableofcontents\tableofcontents
2961 \let\LWR@origlistoffigures\listoffigures
2962 \let\LWR@origlistoftables\listoftables
2963 \renewcommand*{\tableofcontents}{%

```

Copy the `.toc` file to `.sidetoc` for printing the `sidetoc`. The original `.toc` file is renewed when `\tableofcontents` is finished.

```

2964 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
2965 \LWR@printpendingfootnotes
2966 \LWR@origtableofcontents%
2967 }
2968 \renewcommand*{\listoffigures}{
2969 \LWR@printpendingfootnotes
2970 \LWR@origlistoffigures
2971 }
2972
2973 \renewcommand*{\listoftables}{
2974 \LWR@printpendingfootnotes
2975 \LWR@origlistoftables
2976 }

```

## 52.2 High-level TOC commands

`\listof`  $\{\langle type \rangle\} \{\langle title \rangle\}$

Emulate the `\listof` command from the `float` package (section 91). 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.

```

2977 \NewDocumentCommand{\listof}{m +m}{%
2978 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
2979 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
2980 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
2981 \jobname.\csuse{ext@#1}\relax
2982 }

```

### 52.3 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.

---

```
2983 \end{warpHTML}
```

**for HTML & PRINT:** 2984 `\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`.

```
2985 \newcounter{SideTOCDepth}
2986 \setcounter{SideTOCDepth}{1}
```

`\sidetocname` Holds the default name for the sidetoc.

```
2987 \newcommand{\sidetocname}{Contents}
```

```
2988 \end{warpall}
```

**for HTML output:** 2989 `\begin{warpHTML}`

`\LWR@sidetoc` Creates the actual side-TOC.

```
2990 \newcommand*{\LWR@sidetoc}{
2991 \LWR@stoppars
2992
```

The entire sideTOC is placed into a `nav` of class `sidetoc`.

```
2993 \LWR@htmlclass{nav}{sidetoc}
2994
2995 \setcounter{tocdepth}{\value{SideTOCDepth}}
2996
```

The title is placed into a div of class `sidetoctitle`, and may contain paragraphs.

```
2997 \begin{BlockClass}{sidetoctitle}
2998 \sidetocname
2999 \end{BlockClass}
```

The table of contents is placed into a div of class `sidetoccontents`.

```
3000 \begin{BlockClass}{sidetoccontents}
3001 \LinkHome
3002
3003 \LWR@myshorttoc{sidetoc}
3004 \end{BlockClass}
3005 \LWR@html@elementclassend{nav}{sidetoc}
3006 }
```

## 52.4 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:

```
3007 \renewcommand*{\numberline}[1]{%
3008 \LWR@sectionnumber{#1}%
3009 }
```

`\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

```
3010 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
3011 \ifthenelse{\cnttest{#1}{<=}{\value{tocdepth}}}{%
3012 \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```
3013 \LWR@subhyperrefclass{%
3014 \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
3015 \LWR@stoppars%
3016 }
3017 {}
3018 }
```

Ctrl **lofdepth** TOC depth for figures.

```
3019 \newcounter{lofdepth}
3020 \setcounter{lofdepth}{1}
```

Ctrl **lotdepth** TOC depth for tables.

```
3021 \newcounter{lotdepth}
3022 \setcounter{lotdepth}{1}
```

**\hypertocfloat** {<1: depth>} {<2: type>} {<3: ext of parent>} {<4: caption>} {<5: page>}

**#1** is depth

**#2** is figure, table, etc.

**#3** is lof, lot, of the parent.

**#4** the text of the caption

**#5** page number

```
3023 \newcommand{\hypertocfloat}[5]{%
3024 \LWR@startpars
```

If some float-creation package has not yet defined the float type's **lofdepth** counter, etc, define it here:

```
3025 \@ifundefined{c@#3depth}{%
3026 \newcounter{#3depth}%
3027 \setcounter{#3depth}{1}%
3028 }{}%
```

Respond to **lofdepth**, etc.:

```
3029 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
3030 \ifthenelse{\cnttest{#1}{<=}}{\arabic{#3depth}}{%
3031 \LWR@startpars%
```

Create an HTML link to filename#autofloat-(float number), with text of the caption, of the given HTML class.

```

3032 \LWR@subhyperrefclass{%
3033 \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
3034 \#autofloat-\arabic{LWR@nextautofloat}}%
3035 {#4}{toc#2}%
3036 \LWR@stoppars%
3037 }{}%
3038 }

```

Automatically called by \contentsline:

```

3039 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
3040 \DeclareDocumentCommand{\l@chapter}{m m}
3041 {\hypertoc{0}{chapter}{#1}{#2}}
3042 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
3043 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
3044 \renewcommand{\l@subsubsection}[2]
3045 {\hypertoc{3}{subsubsection}{#1}{#2}}
3046 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
3047 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
3048 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
3049 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

3050 \end{warpHTML}

```

## 53 Index

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)

for HTML output: 3051 \begin{warpHTML}

**\LWR@indexsection** Controls whether the index will be in a section or a chapter, depending on the documentclass.

```

3052 \@ifundefined{chapter}
3053 {\newcommand*\LWR@indexsection{\section{\indexname}}}
3054 {\newcommand*\LWR@indexsection{\chapter{\indexname}}}

```

\printindex

```

3055 \let\LWR@origprintindex\printindex
3056
3057 \renewcommand*{\printindex}
3058 {
3059 \LWR@indexsection
3060 \LWR@startpars
3061 \LWR@origprintindex
3062 }

```

Env **theindex**

```

3063 \renewenvironment*{theindex}{%
3064 \let\item\LWR@indexitem%
3065 \let\subitem\LWR@indexsubitem%
3066 \let\subsubitem\LWR@indexsubsubitem%
3067 }{}

```

**\LWR@indexitem**

```

3068 \newcommand{\LWR@indexitem}{
3069
3070 \InlineClass{indexitem}{}
3071 }

```

**\LWR@indexitem**

```

3072 \newcommand{\LWR@indexsubitem}{
3073
3074 \InlineClass{indexsubitem}{}
3075 }

```

**\LWR@indexitem**

```

3076 \newcommand{\LWR@indexsubsubitem}{
3077
3078 \InlineClass{indexsubsubitem}{}
3079 }

```

**\hyperindexref**  $\{\langle autosecnumber \rangle\}$

**\hyperindexref{web address}** is inserted into \*.ind by the xindy style file  
lwarp\_html.xdy

```

3080 \newcommand*{\hyperindexref}[1]{
3081 \LWR@htmltag{a href="%

```



Create an HTML reference to the autosection:

```
3082 \LWR@htmlrefsectionfilename{autopage-#1}\#autosec-#1"{}%}
```

Add the section number and section name, then close the tag:

```
3083 \LWR@origref{autopage-#1}\enskip\LWR@nameref{autopage-#1}%
3084 \LWR@htmltag{/a}%
3085 }
```

```
3086 \end{warpHTML}
```

**for PRINT output:** A null command for print mode, in case `hyperref` was not used:

```
3087 \begin{warpprint}
3088 \newcommand{\hyperindexref}[1]{#1}
3089 \end{warpprint}
```

## 54 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>7</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}`

**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.

<sup>7</sup>See section 159 regarding fonts and fractions.

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

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

**MathJax math option** The popular MathJax alternative ([mathjax.org](https://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, `\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**

**L<sup>A</sup>T<sub>E</sub>X** macros

- MathJax does not automatically support custom L<sup>A</sup>T<sub>E</sub>X macros, but they may be set up by the user.

## custom MathJax macros

As an example of using custom L<sup>A</sup>T<sub>E</sub>X macros with MathJax, place the following at the start of the document, after `\begin{document}`:

---

```
\begin{warpHTML} % Only for HTML output,
\ifbool{mathjax} % and only for MathJax output:
{
\(\ % New macros for MathJax are placed inside a math expression:
\newcommand{\expval}[1]{\langle#1\rangle}
\newcommand{\abs}[1]{\lvert#1\rvert}
\)
}{ }
\end{warpHTML}
```

---

for HTML output: 3090 `\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.

```
3091 \let\LWR@origtextdollar\$
3092
3093 \renewcommand*{\$}{%
3094 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{
3095 {\LWR@origtextdollar}%
3096 {\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
3097 }
```

Ctr LWR@externalfilecnt Counter for the external files which are generated and then referenced from the HTML:

```
3098 \newcounter{LWR@externalfilecnt}
```

## 54.1 Inline and display math

```
3099 \let\LWR@origdollar=$
3100 \let\secondorigdollar=$% balance for editor syntax highlighting

3101 \let\LWR@origopenparen\(  
3102 \let\LWR@origcloseparen\)
```

$\$$  Redefine the dollar sign to place math inside a `lateximage`, or use MathJax:  
 $\$ \$$

```
3103 \begingroup
3104 \catcode'\$=\active%
3105 \protected\gdef$\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%
```

`\LWR@doubledollar` Redefine the double dollar sign to place math inside a `lateximage`, or use MathJax:

```
3106 \gdef\LWR@doubledollar$#1$${
3107 \ifbool{mathjax}%
```

For MathJax, print the math between `\[` and `\]`:

```
3108 {\textbackslash[\LWR@HTMLsanitize{#1}\textbackslash]}
```

For SVG, print the math inside a `lateximage`, with an ALT tag of the  $\LaTeX$  code:

```
3109 {% not mathjax
3110
3111 \begin{lateximage}%
3112 [\textbackslash{[] \LWR@HTMLsanitize{#1} \textbackslash{[]}]%
3113 \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
3114 \end{lateximage}
3115
3116 }
3117 }%
```

`\LWR@singledollar` Redefine the single dollar sign to place math inside a `lateximage`, or use MathJax:

```
3118 \gdef\LWR@singledollar#1${%
3119 \ifbool{mathjax}%
```

For MathJax, print the math between `\(` and `\)`:

```
3120 {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}
```

For SVG, print the math inside a `lateximage`, with an ALT tag of the  $\LaTeX$  code:

```
3121 {% not mathjax
3122 \begin{lateximage}%
3123 [\textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)]%
3124 \LWR@origdollar#1\LWR@origdollar%
3125 \end{lateximage}%
3126 }%
3127 }%
```

`\(` Redefine to the above dollar macros.  
`\)`

```

3128 \gdef\(#1\){$#1$}
3129 \gdef\[#1\]{$$#1$$}
3130
3131 \endgroup

```

Remove the old `math` and `displaymath` environments:

```

3132 \let\math\relax
3133 \let\endmath\relax
3134 \let\displaymath\relax
3135 \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`.

```

3136 \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`.

```

3137 \NewEnviron{displaymath}{\expandafter[\BODY]\@ignoretrue}

```

When the document begins, the dollar sign must be made active to trigger the new math macros:

```

3138 \AtBeginDocument{\catcode'\$=\active}

```

## 54.2 MathJax support

Ctr `LWR@nextequation` Used to add one to compute the next equation number.

```

3139 \newcounter{LWR@nextequation}

```

`\LWR@syncmathjax` Sets the MathJax equation format and number for the following equations.

These MathJax commands are printed inside “\( $\)$ ” characters. They are printed to HTML output, not interpreted by  $\text{\LaTeX}$ .

```

3140 \newcommand*{\LWR@syncmathjax}{%

```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```

3141 \ifcsdef{thechapter}{

```

```

3142 \BlockClassSingle{hidden}{
3143 \textbackslash(
3144 \textbackslash{}seteqsection \{\thechapter\}
3145 \textbackslash)
3146 }
3147 }
3148 {}% not using chapters
3149

```

MathJax doesn't allow setting the equation number to 1:

```

3150 \ifthenelse{\cnttest{\value{equation}}>0}
3151 {

```

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.

```

3152 \setcounter{LWR@nextequation}{\value{equation}}
3153 \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.

```

3154 \BlockClassSingle{hidden}{
3155 \textbackslash(
3156 \textbackslash{}seteqnumber \{\arabic{LWR@nextequation}\}
3157 \textbackslash)
3158 }
3159 {}% not eq > 1
3160 }

```

`\LWR@restoremathlatexformatting` While producing math, use regular L<sup>A</sup>T<sub>E</sub>X formatting instead of HTML tags.

```

3161 \newcommand*{\LWR@restoremathlatexformatting}{%
3162 \let\hspace\LWR@orighspace%
3163 \let\rule\LWR@origrule%
3164 \let\,\LWR@origcomma% disable HTML short unbreakable space
3165 \let\textit\LWR@origtextit%
3166 \let\textbf\LWR@origtextbf%
3167 \let\texttt\LWR@origtexttt%
3168 \let\textsc\LWR@origtextsc%
3169 \let\textsf\LWR@origtextsf%
3170 \let\textrm\LWR@origtextrm%
3171 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
3172 \let\textsuperscript\LWR@origtextsuperscript%
3173 \let\textsubscript\LWR@origtextsubscript%
3174 \let~\LWR@origtilde%
3175 \let\enskip\LWR@origenskip%
3176 \let\quad\LWR@origquad%

```

```
3177 \let\qqquad\LWR@origqqquad%
3178 }
```

`\LWR@hidelatexequation`  $\{\langle environment \rangle\} \{\langle contents \rangle\}$

Creates the  $\text{\LaTeX}$  version of the equation inside an HTML comment.

```
3179 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```
3180 \LWR@stoppars
3181 \LWR@htmlopencomment
3182
```

Start the  $\text{\LaTeX}$  math environment inside the HTML comment:

```
3183 \begingroup
3184 \csuse{\LWR@orig#1}
```

While in the math environment, restore various commands to their  $\text{\LaTeX}$  meanings.

```
3185 \LWR@restoremathlatexformatting
```

See `\LWR@htmlmathlabel` in section [54.4.1](#).

Print the contents of the equation:

```
3186 #2
```

End the  $\text{\LaTeX}$  math environment inside the HTML comment:

```
3187 \csuse{\LWR@origend#1}
3188 \endgroup
3189
```

Close the HTML comment and resume HTML paragraph handling:

```
3190 \LWR@htmlclosecomment
3191 \LWR@startpars
3192 }
```

`\LWR@addmathjax`  $\{\langle environment \rangle\} \{\langle contents \rangle\}$

Given the name of a math environment and its contents, create a MathJax instance. The contents are printed to HTML output, not interpreted by  $\text{\LaTeX}$ .

```
3193 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
3194
```

Enclose the MathJax environment inside printed “\(" and “\)” characters.

```
3195 \textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
3196 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
3197 \textbackslash{}end\{#1\}
3198
3199 }
```

### 54.3 Equation environment

Remember existing `equation` environment:

```
3200 \let\LWR@origequation\equation
3201 \let\LWR@origendequation\endequation
```

Remove existing `equation` environment:

```
3202 \let\equation\relax
3203 \let\endequation\relax
```

**Env**   **equation**   The new `equation` environment is created with `\NewEnviron` (from the `environ` package), which stores the contents of its environment in a macro called `\BODY`.

For SVG math output, the contents are typeset using the original `equation` inside a `lateximage`, along with an ALT tag containing a detokenized copy of the L<sup>A</sup>T<sub>E</sub>X 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.

```
3204 \NewEnviron{equation}{%
3205
3206 \ifbool{mathjax}
```

MathJax output:

```
3207 {
```



Print commands to synchronize MathJax's equation number and format to the current L<sup>A</sup>T<sub>E</sub>X chapter/section and equation number:

```
3208 \LWR@syncmathjax
```

Print the L<sup>A</sup>T<sub>E</sub>X math inside an HTML comment:

```
3209 \LWR@hidelatexequation{equation}{\BODY}
3210 }
```

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`.

```
3211 {% not mathjax
```

Begin the `lateximage` with an ALT tag containing the math source:

```
3212 \begin{lateximage}[(\theequation) \textbackslash{begin\{equation\}}]%
3213 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}}}%
3214 \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.

```
3215 \LWR@origequation
3216 \BODY% contents collected by NewEnviron
3217 \LWR@origendequation
3218 \end{lateximage}%
3219 }
3220
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
3221 }[\ifbool{mathjax}{\LWR@addmathjax{equation}{\BODY}}{}]
```

## 54.4 AMS Math environments

### 54.4.1 Support macros

Bool `LWR@amsmultline` True if processing a multiline environment.

To compensate for `multline`-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 `multline` environment and not an `equation`.

```
3222 \newbool{LWR@amsmultline}
3223 \boolfalse{LWR@amsmultline}
```

```
\LWR@htmlmathlabel {(\label)}
```

`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`.

```
3224 \newcommand*{\LWR@htmlmathlabel}[1]{%
3225 \LWR@traceinfo{\LWR@htmlmathlabel #1}%
3226 \ifbool{mathjax}{%
```

The combined  $\text{\LaTeX}$  & HTML label is printed in a `\text` field:

```
3227 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
3228 \ifbool{LWR@amsmultline}{\hspace*{\totwidth0}}
```

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.)

```
3229 \LWR@htmlclosecomment%
3230 \LWR@origltx@label{#1}%
3231 \LWR@htmlopencomment%
3232 }% text
3233 }% mathjax
3234 {%
3235 \LWR@origltx@label{#1}%
3236 }%
3237 }
```

`\LWR@beginhideamsmath` Starts hiding  $\text{\LaTeX}$  math inside an HTML comment.

```
3238 \newcommand*{\LWR@beginhideamsmath}{
3239 \LWR@stoppars
3240
3241 \LWR@htmlopencomment
```

```

3242
3243 \begingroup
3244 \LWR@restoremathlatexformatting
3245 }

```

`\LWR@endhideamsmath` Ends hiding  $\text{\LaTeX}$  math inside an HTML comment.

```

3246 \newcommand*{\LWR@endhideamsmath}{
3247 \endgroup
3248
3249 \LWR@htmlclosecomment
3250
3251 \LWR@startpars
3252 }

```

#### 54.4.2 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  $\mathcal{A}\mathcal{M}\mathcal{S}$  environment.

For MathJax: Each environment is syched with  $\text{\LaTeX}$ 's equation numbers, typeset with  $\text{\LaTeX}$  inside an HTML comment, then printed to HTML output for MathJax to process.

Env `multline`

```

3253 \BeforeBeginEnvironment{multline}{
3254 \ifbool{mathjax}
3255 {
3256 \LWR@syncmathjax
3257 \booltrue{LWR@amsmultline}
3258 \LWR@beginhideamsmath
3259 }
3260 {
3261 \lateximage[LWRAMSMATHBODY]
3262 }
3263 }
3264
3265 \AfterEndEnvironment{multline}{
3266
3267 \ifbool{mathjax}
3268 {

```

```

3269 \LWR@endhideamsmath
3270 \boolfalse{LWR@amsmultline}
3271 \LWR@addmathjax{multline*}{\the\@envbody}
3272 }
3273 {\endlateximage}
3274
3275 }

```

Env **multline\***

```

3276 \BeforeBeginEnvironment{multline*}{
3277 \ifbool{mathjax}
3278 {
3279 \LWR@syncmathjax
3280 \booltrue{LWR@amsmultline}
3281 \LWR@beginhideamsmath
3282 }
3283 {
3284 \lateximage[LWRAMSMATHBODY]
3285 }
3286 }
3287
3288 \AfterEndEnvironment{multline*}{
3289
3290 \ifbool{mathjax}
3291 {
3292 \LWR@endhideamsmath
3293 \boolfalse{LWR@amsmultline}
3294 \LWR@addmathjax{multline*}{\the\@envbody}
3295 }
3296 {\endlateximage}
3297
3298 }
3299

```

Env **gather**

```

3300 \BeforeBeginEnvironment{gather}{
3301 \ifbool{mathjax}
3302 {
3303 \LWR@syncmathjax
3304 \boolfalse{LWR@amsmultline}
3305 \LWR@beginhideamsmath
3306 }
3307 {
3308 \lateximage[LWRAMSMATHBODY]
3309 }
3310 }
3311

```

```

3312 \AfterEndEnvironment{gather}{
3313
3314 \ifbool{mathjax}
3315 {
3316 \LWR@endhideamsmath
3317 \LWR@addmathjax{gather}{\the\@envbody}
3318 }
3319 {\endlateximage}
3320
3321 }

```

Env **gather\***

```

3322 \BeforeBeginEnvironment{gather*}{
3323 \ifbool{mathjax}
3324 {
3325 \LWR@syncmathjax
3326 \boolfalse{LWR@amsmultline}
3327 \LWR@beginhideamsmath
3328 }
3329 {
3330 \lateximage[LWRAMSMATHBODY]
3331 }
3332 }
3333
3334 \AfterEndEnvironment{gather*}{
3335
3336 \ifbool{mathjax}
3337 {
3338 \LWR@endhideamsmath
3339 \LWR@addmathjax{gather*}{\the\@envbody}
3340 }
3341 {\endlateximage}
3342
3343 }

```

Env **align**

```

3344 \BeforeBeginEnvironment{align}{
3345 \ifbool{mathjax}
3346 {
3347 \LWR@syncmathjax
3348 \boolfalse{LWR@amsmultline}
3349 \LWR@beginhideamsmath
3350 }
3351 {
3352 \lateximage[LWRAMSMATHBODY]
3353 }
3354 }

```

```

3355
3356 \AfterEndEnvironment{align}{
3357
3358 \ifbool{mathjax}
3359 {
3360 \LWR@endhideamsmath
3361 \LWR@addmathjax{align}{\the\@envbody}
3362 }
3363 {\endlateximage}
3364
3365 }

```

Env **align\***

```

3366 \BeforeBeginEnvironment{align*}{
3367 \ifbool{mathjax}
3368 {
3369 \LWR@syncmathjax
3370 \boolfalse{LWR@amsmultiline}
3371 \LWR@beginhideamsmath
3372 }
3373 {
3374 \lateximage[LWRAMSMATHBODY]
3375 }
3376 }
3377
3378 \AfterEndEnvironment{align*}{
3379
3380 \ifbool{mathjax}
3381 {
3382 \LWR@endhideamsmath
3383 \LWR@addmathjax{align*}{\the\@envbody}
3384 }
3385 {\endlateximage}
3386
3387 }

```

Env **flalign**

```

3388 \BeforeBeginEnvironment{flalign}{
3389 \ifbool{mathjax}
3390 {
3391 \LWR@syncmathjax
3392 \boolfalse{LWR@amsmultiline}
3393 \LWR@beginhideamsmath
3394 }
3395 {
3396 \lateximage[LWRAMSMATHBODY]
3397 }

```

```

3398 }
3399
3400 \AfterEndEnvironment{flalign}{
3401
3402 \ifbool{mathjax}
3403 {
3404 \LWR@endhideamsmath
3405 \LWR@addmathjax{flalign}{\the\@envbody}
3406 }
3407 {\endlateximage}
3408
3409 }

```

Env **flalign\***

```

3410 \BeforeBeginEnvironment{flalign*}{
3411 \ifbool{mathjax}
3412 {
3413 \LWR@syncmathjax
3414 \boolfalse{LWR@amsmultline}
3415 \LWR@beginhideamsmath
3416 }
3417 {
3418 \lateximage[LWR@SMATHBODY]
3419 }
3420 }
3421
3422 \AfterEndEnvironment{flalign*}{
3423
3424 \ifbool{mathjax}
3425 {
3426 \LWR@endhideamsmath
3427 \LWR@addmathjax{flalign*}{\the\@envbody}
3428 }
3429 {\endlateximage}
3430
3431 }

3432 \end{warpHTML}

```

## 55 Lateximages

A `\lateximage` is typeset on its own PDF page inside an HTML comment which starts on the preceding page and ends on following page, and instructions are written to `lateximage.txt` for `lwarpnk` 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:** 3433 `\begin{warpHTML}`

Ctr `LWR@lateximagenumber` Sequence the images.

```
3434 \newcounter{LWR@lateximagenumber}
3435 \setcounter{LWR@lateximagenumber}{0}
```

Ctr `LWR@lateximagedepth` Do not create `\lateximage` inside of `\lateximage`.

```
3436 \newcounter{LWR@lateximagedepth}
3437 \setcounter{LWR@lateximagedepth}{0}
```

Declare the `\LWR@file` for writing to generate file `lateximages.txt`:

```
3438 \ifcsdef{\LWR@file}{-}{\newwrite{\LWR@file}}
```

A few utility macros to write special characters:

```
3439 \edef\LWR@hashmark{\string#} % for use in \write
3440 \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`.

```
3441 \newcounter{LWR@Lipage}
```

```
3442 \end{warpHTML}
```

**for HTML & PRINT:** 3443 `\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.*

```
3444 \newcommand*{\LateximageFontSizeName}{large}

3445 \end{warpall}
```

**for HTML output:** 3446 \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...

```
3447 \newcommand{\LWR@HTMLsanitize}[1]{%
3448 \protect\StrSubstitute{\detokenize{#1}}{%
3449 {\detokenize{&}}{%
3450 {\detokenize{&}}[\LWR@strresult]}%
3451 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3452 {\detokenize{<}}{%
3453 {\detokenize{<}}[\LWR@strresult]}%
3454 [\LWR@strresult]}%
3455 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3456 {\detokenize{>}}{%
3457 {\detokenize{>}}[\LWR@strresult]}%
3458 [\LWR@strresult]}%
3459 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3460 {\detokenize{##}}{%
3461 {\detokenize{##}}[\LWR@strresult]}%
3462 [\LWR@strresult]}%
3463 \LWR@strresult%
3464 }
```

`\LWR@HTMLsanitizeexpand`  $\{\langle text \rangle\}$

This version expands the argument before sanitizing it.

```
3465 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
3466 \protect\StrSubstitute{\detokenize\expandafter{#1}}{%
3467 {\detokenize{&}}{%
3468 {\detokenize{&}}[\LWR@strresult]}%
3469 [\LWR@strresult]}%
3470 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3471 {\detokenize{<}}{%
3472 {\detokenize{<}}[\LWR@strresult]}%
3473 [\LWR@strresult]}%
```

```

3474 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3475 {\detokenize{>}}}%
3476 {\detokenize{&gt;}}}%
3477 [\LWR@strresult]%
3478 \LWR@strresult%
3479 }

```

Env `lateximage` [*alttag*]

```

3480 \NewDocumentEnvironment{lateximage}{0{image}}{%
3481 \LWR@traceinfo{lateximage: starting on page \arabic{page}}}%
3482 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%

```

If nesting inside an already-existing `lateximage`, simply record one more level:

```

3483 {%
3484 \addtocounter{LWR@lateximagedepth}{1}%
3485 }%

```

Otherwise, this is the outer-most `lateximage`:

```

3486 {% start of outer-most lateximage

```

Starting a new `lateximage`:

```

3487 \addtocounter{LWR@lateximagenumber}{1}%
3488 \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%

```

While inside a `lateximage`, do not use `mathjax`:

```

3489 \boolfalse{mathjax}

```

Be sure that are doing a paragraph:

```

3490 \LWR@ensuredoingapar%

```

Next file:

```

3491 \addtocounter{LWR@externalfilecnt}{1}%
3492 \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%

```

Figure out what the next page number will be:

```

3493 \setcounterpageref{LWR@Lipage}{LWR@lateximage\theLWR@lateximagenumber}%
3494 \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:

```

3495 \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
3496 class="lateximagesource"{}} \LWR@orignewline

```

Write instructions to the lateximages.txt file:

```

3497 \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.

```

3498 \LWR@htmlopencomment%
3499 \addtocounter{LWR@lateximagedepth}{1}%

```

Start the new PDF page:

```

3500 \LWR@orignewpage%

```

Typeset the image in a “standard” width page and font size:

```

3501 \LWR@origminipage{6in}%
3502 \csuse{LWR@orig\LateximageFontSizeName}%

```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for \hspace, etc. inside a lateximage.

```

3503 \let\hspace\LWR@orighspace%
3504 \let\rule\LWR@origrule%
3505 \let\,\LWR@origcomma% disable HTML short unbreakable space
3506 \let\textit\LWR@origtextit%
3507 \let\textbf\LWR@origtextbf%
3508 \let\texttt\LWR@origtexttt%
3509 \let\textsc\LWR@origtextsc%
3510 \let\textsf\LWR@origtextsf%
3511 \let\textrm\LWR@origtextrm%
3512 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
3513 \let\textsuperscript\LWR@origtextsuperscript%
3514 \let\textsubscript\LWR@origtextsubscript%
3515 \let~\LWR@origtilde%
3516 \let\enskip\LWR@origenskip%
3517 \let\quad\LWR@origquad%
3518 \let\qquad\LWR@origqquad%
3519 \let\tabular\LWR@origtabular%
3520 \let\endtabular\LWR@origendtabular%
3521 \let\newline\LWR@orignewline%
3522 \LWR@origlabel{LWR@lateximage\arabic{LWR@lateximagenumber}}%
3523 }% end of outer-most lateximage
3524 }% end of \begin{lateximage}
3525 {% start of \end{lateximage}

```

```
3526 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
3527 {%
3528 \addtocounter{LWR@lateximagedepth}{-1}%
3529 }%
```

if this is the outer-most lateximage:

```
3530 {% end of outer-most lateximage
```

Finish the lateximage minipage and start a new PDF page:

```
3531 \LWR@origendminipage%
3532 \LWR@orignewpage%
3533 \LWR@origscriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by pdftotext:

```
3534 \LWR@htmlclosecomment{}\LWR@orignewline%
3535 \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}%
```

Create a link to the lateximage, allowing its natural height:

If the alt tag is given as “LWRAMSMATHBODY”, then use the text collected by the amsmath multiline, gather, or align environments.

```
3536 \ifthenelse{\equal{#1}{LWRAMSMATHBODY}}{%
3537 {%
3538 \LWR@subinlineimage[%
3539 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}}%
3540 ]%
3541 {lateximage}%
3542 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}%
3543 {svg}%
3544 {}%
3545 }%
3546 {%
3547 \LWR@subinlineimage[#1]{lateximage}%
3548 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}{svg}{}%
3549 }%
3550 % \LWR@orignewline% Removed to prevent extra space.
```

Be sure that are doing a paragraph:

```
3551 \LWR@ensuredoingapar%
```

Close the HTML span which has the `pdftotext` comment and also the link to the `.svg` image:

```
3552 \LWR@htmltag{/span}%
3553 \LWR@htmlcomment{End of lateximage}%
3554 % \LWR@orignewline% Removed to prevent extra space.
```

Undo one `lateximage` level:

```
3555 \addtocounter{LWR@lateximagedepth}{-1}%
3556 }% end of outer-most lateximage
3557 \LWR@traceinfo{lateximage: done}
3558 }%
3559 \end{warpHTML}
```

```
for PRINT output: 3560 \begin{warpprint}
3561 \newenvironment{lateximage}[1][\minipage{\linewidth}]{\endminipage}
3562 \end{warpprint}
```

## 56 center, flushleft, flushright

```
for HTML output: 3563 \begin{warpHTML}
```

Env **center** Replace `center` functionality with CSS tags:

```
3564 \renewenvironment*{center}
3565 {\BlockClass{center}}
3566 {\endBlockClass}
```

Env **flushright**

```
3567 \renewenvironment*{flushright}
3568 {\BlockClass{flushright}}
3569 {\endBlockClass}
```

Env **flushleft**

```
3570 \renewenvironment*{flushleft}
3571 {\BlockClass{flushleft}}
3572 {\endBlockClass}
```

```
3573 \end{warpHTML}
```

## 57 Siunitx

Pkg **siunitx**

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

for HTML output: 3574 `\begin{warpHTML}`

Options for siunitx:

```
3575 \PassOptionsToPackage{
3576 detect-mode=true,
3577 per-mode=symbol,% fraction is not seen by pdftotext
3578 text-celsius = {\HTMLentity{deg}C},
3579 text-degree = {\HTMLentity{deg}},
3580 }{siunitx}

3581 \end{warpHTML}
```

## 58 Graphics

Pkg **graphics**

Pkg **graphicx**

**\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:** 3582 `\begin{warpHTML}`

## 58.1 `\graphicspath`

`\graphicspath` `{\langle path \rangle}`

3583 `\newcommand*\thisgraphicspath{\{\}}`

3584 `\renewcommand*\graphicspath[1]{\renewcommand*\thisgraphicspath{\#1}}`


`\DeclareGraphicsExtensions` `{\langle list \rangle}`

`\DeclareGraphicsRule` `{\langle \rangle} {\langle \rangle} {\langle \rangle} {\langle \rangle}`

3585 `\renewcommand*\DeclareGraphicsExtensions[1]{}`

3586 `\renewcommand*\DeclareGraphicsRule[4]{}`

## 58.2 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.

3587 `\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.

3588 `\newlength{\LWR@igwidth}`

3589 `\newlength{\LWR@igheight}`

3590 `\newcommand*\LWR@igwidthstyle{\}`

3591 `\newcommand*\LWR@igheightstyle{\}`

3592 `\newcommand*\LWR@igorigin{\}`

```

3593 \newcommand*{\LWR@igangle}{%
3594 \newcommand*{\LWR@igxscale}{1}
3595 \newcommand*{\LWR@igyscale}{1}
3596 \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:

```

3597 \define@key{igraph}{width}{%
3598 \setlength{\LWR@igwidth}{#1}%
3599 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}{%
3600 {%

```

Default to use the converted fixed length given:

```

3601 \uselengthunit{PT}%
3602 \renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%

```

If ex or em dimensions were given, use those instead:

```

3603 \IfEndWith{#1}{ex}%
3604 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes ex
3605 {}% not ex
3606 \IfEndWith{#1}{em}%
3607 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes em
3608 {}% not em
3609 \IfEndWith{#1}{\}%
3610 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes percent
3611 {}% not percent
3612 \IfEndWith{#1}{px}%
3613 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
3614 {}% not px
3615 }{}% end of length > 0pt
3616 }

```

If an optional height was given, set an HTML style:

```

3617 \define@key{igraph}{height}{%
3618 \setlength{\LWR@igheight}{#1}%
3619 \ifthenelse{\lengthtest{\LWR@igheight > 0pt}}{%
3620 {%

```

Default to use the converted fixed length given:

```

3621 \uselengthunit{PT}%
3622 \renewcommand*{\LWR@igheightstyle}{%
3623 height:\rndprintlength{\LWR@igheight} %

```



3624 }%

If ex or em dimensions were given, use those instead:

```

3625 \IfEndWith{#1}{ex}%
3626 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
3627 {}% not ex
3628 \IfEndWith{#1}{em}%
3629 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
3630 {}% not em
3631 \IfEndWith{#1}{\}%
3632 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
3633 {}% not percent
3634 \IfEndWith{#1}{px}%
3635 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
3636 {}% not px
3637 }{}% end of length > Opt
3638 }

```

Handle origin key:

```

3639 \define@key{igraph}{origin}{%
3640 \renewcommand*{\LWR@igorigin}{#1}%
3641 }

```

Handle angle key:

```

3642 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}

```

Handle class key:

```

3643 \define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}
3644

```

It appears that `graphicx` does not have separate keys for `xscale` and `yscale`. `scale` adjusts both at the same time.

```

3645 \define@key{igraph}{scale}{%
3646 \renewcommand*{\LWR@igxscale}{#1}%
3647 \renewcommand*{\LWR@igyscale}{#1}}

```

Numerous ignored keys:

```

3648 \define@key{igraph}{bb}{}
3649 \define@key{igraph}{bblx}{}
3650 \define@key{igraph}{bbly}{}
3651 \define@key{igraph}{bburx}{}
3652 \define@key{igraph}{bbury}{}
3653 \define@key{igraph}{natwidth}{}

```

```

3654 \define@key{igraph}{natheight}{}
3655 \define@key{igraph}{hiresbb}{}
3656 \define@key{igraph}{viewport}{}
3657 \define@key{igraph}{trim}{}
3658 \define@key{igraph}{totalheight}{}
3659 \define@key{igraph}{keepaspectratio}{}
3660 \define@key{igraph}{clip}{}
3661 \define@key{igraph}{draft}{}
3662 \define@key{igraph}{type}{}
3663 \define@key{igraph}{ext}{}
3664 \define@key{igraph}{read}{}
3665 \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.

```

3666 \newcommand*{\LWR@rotstyle}[2]{%
3667   #1transform:rotate(-#2deg);
3668 }

```

`\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.

```

3669 \newcommand*{\LWR@scalestyle}[3]{%
3670   #1transform:scale(#2,#3);
3671 }

```

### 58.3 `\includegraphics`

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```

3672 \newbool{LWR@infloatrow}
3673 \boolfalse{LWR@infloatrow}

3674 \newcommand*{\LWR@imageextension}{}
3675 \newcommand*{\LWR@expgraphicsfilename}{}

```

`\LWR@includegraphicsb` [*options*] {*filename*}

```
3676
3677 \newcommand*{\LWR@includegraphicsb}[2] []
3678 {%
```

Start the image tag on a new line, allow PDF output word wrap:

```
3679 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
3680 \ifthenelse{\cnttest{\value{\LWR@minipagedepth}}{=}{0}}{%
3681 \ifbool{\LWR@infloatrow}%
3682 {}
3683 {% not in a minipage or a floatrow:
3684 \setlength{\linewidth}{6in}%
3685 \setlength{\textwidth}{6in}%
3686 \setlength{\textheight}{9in}%
3687 }%
3688 }{}%
```

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.

```
3689 \edef\LWR@expgraphicsfilename{#2}
3690 \renewcommand*{\LWR@imageextension}{}%
3691 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.jpg}%
3692 {\renewcommand*{\LWR@imageextension}{.jpg}}{}%
3693 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.JPG}%
3694 {\renewcommand*{\LWR@imageextension}{.JPG}}{}%
3695 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.png}%
3696 {\renewcommand*{\LWR@imageextension}{.png}}{}%
3697 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.PNG}%
3698 {\renewcommand*{\LWR@imageextension}{.PNG}}{}%
3699 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.gif}%
3700 {\renewcommand*{\LWR@imageextension}{.gif}}{}%
3701 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.GIF}%
3702 {\renewcommand*{\LWR@imageextension}{.GIF}}{}%
3703 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svg}%
3704 {\renewcommand*{\LWR@imageextension}{.svg}}{}%
3705 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVG}%
3706 {\renewcommand*{\LWR@imageextension}{.SVG}}{}%
3707 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svgz}%
3708 {\renewcommand*{\LWR@imageextension}{.svgz}}{}%
```

```

3709 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVGZ}%
3710 {\renewcommand*{\LWR@imageextension}{.SVGZ}}{}%

```

Convert a PDF extension to SVG, leave the result in \LWR@strresult:

Must also \detokenize .pdf and .svg comparison strings.

```

3711 \StrSubstitute{\detokenize\expandafter{\LWR@expgraphicsfilename}}%
3712 {\detokenize{.pdf}}{\detokenize{.svg}}[\LWR@strresult]%
3713 %
3714 \StrSubstitute{\LWR@strresult}%
3715 {\detokenize{.PDF}}{\detokenize{.SVG}}[\LWR@strresult]%

```

For correct em sizing during the width and height conversions:

```

3716 \large%

```

Reset some defaults, possibly will be changed below if options were given:

```

3717 \setlength{\LWR@igwidth}{0pt}%
3718 \setlength{\LWR@igheight}{0pt}%
3719 \renewcommand*{\LWR@igwidthstyle}{}%
3720 \renewcommand*{\LWR@igheightstyle}{}%
3721 \renewcommand*{\LWR@igorigin}{}%
3722 \renewcommand*{\LWR@igangle}{}%
3723 \renewcommand*{\LWR@igxscale}{1}%
3724 \renewcommand*{\LWR@igyscale}{1}%
3725 \renewcommand*{\LWR@igclass}{inlineimage}%

```

Options are in #1

```

3726 \setkeys{igraph}{#1}%

```

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.

```

3727 \href{\thisgraphicspath\LWR@strresult\LWR@imageextension}%
3728 {% start of href
3729 \LWR@htmltag{% start of image tags
3730 img src="\thisgraphicspath\LWR@strresult\LWR@imageextension" \LWR@orignewline
3731 \LWR@origtilde{} alt="\LWR@strresult" \LWR@orignewline

```

Only include a style tag if a width, height, angle, or scale was given:

```

3732 \ifthenelse{

```

```

3733 \NOT\equal{\LWR@igwidthstyle}{ } \OR
3734 \NOT\equal{\LWR@igheightstyle}{ } \OR
3735 \NOT\equal{\LWR@igorigin}{ } \OR
3736 \NOT\equal{\LWR@igangle}{ } \OR
3737 \NOT\equal{\LWR@igxscale}{1} \OR
3738 \NOT\equal{\LWR@igyscale}{1}
3739 }%
3740 {\LWR@origtilde{ } style="%
3741 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{ }}%
3742 {\LWR@igwidthstyle; }}%
3743 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{ }}%
3744 {\LWR@igheightstyle; }}%
3745 \ifthenelse{\NOT\equal{\LWR@igorigin}{ }}%
3746 {\LWR@origtilde{ } transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@originnewline}{ }%
3747 \ifthenelse{\NOT\equal{\LWR@igangle}{ }}%
3748 {%
3749 \LWR@rotstyle{-ms-}{\LWR@igangle}%
3750 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
3751 \LWR@rotstyle}{\LWR@igangle%
3752 } }%
3753 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%
3754 \NOT\equal{\LWR@igyscale}{1}}%
3755 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}%
3756 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
3757 \LWR@scalestyle}{\LWR@igxscale}{\LWR@igyscale}}}%
3758 " \LWR@originnewline}{ }%

```

Set the class:

```

3759 \LWR@origtilde{ } class="\LWR@igclass" \LWR@originnewline%
3760 }% end of image tags
3761 }% end of href
3762 \endgroup

```

Return to small-sized output:

```

3763 \LWR@origscriptsize
3764 }

```

`\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.

```

3765 \renewcommand*{\includegraphics}
3766 {%

```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```
3767 \LWR@ensuredoingapar%
3768 \begingroup%
3769 \LWR@includegraphicsb%
3770 }
```

```
3771 \end{warpHTML}
```

**for PRINT output:** For print output, accept and then discard the new `class` key:

```
3772 \begin{warpprint}
3773 \define@key{Gin}{class}{}
3774 \end{warpprint}
```

## 58.4 \rotatebox, \scalebox, \reflectbox

**for HTML output:** 3775 \begin{warpHTML}

`\LWR@rotboxorigin` Holds the origin key letters.

```
3776 \newcommand*{\LWR@rotboxorigin}{}%
```

`\LWR@originname`  $\{ \langle letter \rangle \}$

Given one L<sup>A</sup>T<sub>E</sub>X origin key value, translate into an HTML origin word:

```
3777 \newcommand*{\LWR@originname}[1]{%
3778 \ifthenelse{\equal{#1}{t}}{top}{}%
3779 \ifthenelse{\equal{#1}{b}}{bottom}{}%
3780 \ifthenelse{\equal{#1}{c}}{center}{}%
3781 \ifthenelse{\equal{#1}{l}}{left}{}%
3782 \ifthenelse{\equal{#1}{r}}{right}{}%
3783 }
```

`\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:

```
3784 \newcommand*{\LWR@originnames}[1]{%
3785 \StrChar{#1}{1}[\LWR@strresult]%
3786 \LWR@originname{\LWR@strresult}%
3787 \StrChar{#1}{2}[\LWR@strresult]%
3788 \LWR@originname{\LWR@strresult}%
3789 }
```

Handle the origin key for `\rotatebox`:

```
3790 \define@key{krotbox}{origin}{%
3791 \renewcommand*{\LWR@rotboxorigin}{#1}%
3792 }
```

These keys are ignored:

```
3793 \define@key{krotbox}{x}{}
3794 \define@key{krotbox}{y}{}
3795 \define@key{krotbox}{units}{}

```

`\rotatebox` [*keyval list*] {*angle*} {*text*}

Will `\let\rotatebox\LWR@rotatebox` at `\LWR@LwarpStart`, in case `\rotatebox` was over-written by a later package load.

```
3796 \NewDocumentCommand{\LWR@rotatebox}{0}{ m +m}{%
```

Reset the origin to “none-given”:

```
3797 \renewcommand*{\LWR@rotboxorigin}{}

```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
3798 \setkeys{krotbox}{#1}%

```

Select `inline-block` so that HTML will transform this span:

```
3799 \LWR@htmltagc{span style="display: inline-block; %

```

If an origin was given, translate and print the origin information:

```
3800 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{} }{%
3801 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%

```

Print the rotation information:

```
3802 \LWR@rotstyle{-ms-}{#2} %
3803 \LWR@rotstyle{-webkit-}{#2} %
3804 \LWR@rotstyle{}{#2} %
3805 "{}\LWR@orignewline%

```

Print the text to be rotated:

```
3806 \begin{\LWR@nestspan}%
3807 #3%

```

Close the span:

```
3808 \LWR@htmltagc{/span}%
3809 \end{LWR@nestspan}%
3810 }
```

`\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.

```
3811 \NewDocumentCommand{\LWR@scalebox}{m o m}{%
```

Select `inline-block` so that HTML will transform this span:

```
3812 \LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
3813 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3814 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3815 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3816 "}}%
```

Print the text to be scaled:

```
3817 \begin{LWR@nestspan}%
3818 #3%
```

Close the span:

```
3819 \LWR@htmltagc{/span}%
3820 \end{LWR@nestspan}%
3821 }
```

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

Will `\let\reflectbox\LWR@reflectbox` at `\LWR@LwarpStart`, in case `\reflectbox` was over-written by a later package load.

```
3822 \newcommand{\LWR@reflectbox}[1]{\LWR@scalebox{-1}[1]{#1}}
3823 \end{warpHTML}
```

## 58.5 Null functions

These functions are not supported by `lwarp`'s HTML conversion.



for HTML output: 3824 \begin{warpHTML}

```
\resizebox  {\langle h-length \rangle} {\langle v-length \rangle} {\langle text \rangle}
```

Simply prints its text argument.

```
3825 \renewcommand{\resizebox}[3]{#3}
```

```
3826 \end{warpHTML}
```

## 59 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 8 on 194 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: 3827 \begin{warpHTML}

`\AtEndPreamble` forces `cleveref` to be loaded last:

The following patches are applied after `cleveref` has loaded, and after `\AtBeginDocument`:

```
3828 \AfterEndPreamble{
```

```
\@@setcref  {\langle kindofref \rangle} {\langle label \rangle}
```

```
3829 \renewcommand*\@@setcref[2]{#1{\ref{#2}}-{}{}}
```

```
\@@setcrefrange  {\langle text \rangle} {\langle label \rangle} {\langle label \rangle}
```

```
3830 \renewcommand{\@@setcrefrange}[3]{%
```

```
3831 #1{\ref{#2}}{\ref{#3}}-{}{}}}
```

`\cpagerefFor` Redefinable word between “page(s)” and the page numbers.

```
3832 \newcommand*{\cpagerefFor}{for}
```

`\@@setcpageref`  $\langle typeofref \rangle$   $\langle label \rangle$ , where *typeofref* is “page” or “pages”

```
3833 \renewcommand*{\@@setcpageref}[2]{%
3834 #1{\cpagerefFor\ \cref{#2}}{}}%
3835 }
```

```
3836 \renewcommand{\@@setpagerefrange}[3]{%
3837 #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}}
3838 }% AfterEndPreamble
```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

```
3839 \let\LWR@origlabel\label
3840 \let\label\LWR@newlabel
3841 \let\LWR@origref\ref
3842 \let\ref\LWR@newref% \end{ syntax highlighting
3843 \let\LWR@origpageref\pageref
3844 \let\pageref\LWR@newpageref
3845
3846
3847
3848 \end{warpHTML}
```

## 60 Picture

Env `picture` The `picture` environment is enclosed inside a `\lateximage`.

for HTML output: 3849 `\begin{warpHTML}`

Env `picture`


```
3850 \BeforeBeginEnvironment{picture}{%
3851 \lateximage%
3852 \let\makebox\LWR@origmakebox%
3853 }
3854
3855 \AfterEndEnvironment{picture}{\endlateximage}

3856 \end{warpHTML}
```

## 61 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:** 3857 `\begin{warpHTML}`

### 61.1 Counters and lengths

**Ctrl** `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
3858 \newcounter{LWR@minipagedepth}
3859 \setcounter{LWR@minipagedepth}{0}
```

**Len** `\WR@minipagewidth` Used to convert the width into printable units.

```
3860 \newlength{\WR@minipagewidth}
```

Len `\WR@minipageheight` Used to convert the height into printable units.

```
3861 \newlength{\LWR@minipageheight}
```

Remember the original definitions:

```
3862 \let\LWR@origminipage\minipage
```

```
3863 \let\LWR@origendminipage\endminipage
```

## 61.2 Footnote handling

Also see section 35 for other forms of footnotes.

## 61.3 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:

```
3864 \def\LWR@endminipage{%
3865     \par
3866     \unskip
3867     \ifvoid\@mpfootins\else
3868         \vskip\skip\@mpfootins
3869         \normalcolor
3870 \LWR@htmldivclass{mpfootnotes}
3871 \LWR@origmedskip
3872     \unvbox\@mpfootins
3873 \LWR@htmldivclassend{mpfootnotes}
3874     \fi
3875     \@minipagefalse
3876     \color@endgroup
3877 \egroup
3878 \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.

```
3879 \newcommand*{\LWR@subminipage}{%
3880 \LWR@stoppars
3881 \LWR@origminipage{6in}
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
3882 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
3883 \LWR@startpars%
3884 }
```

`\LWR@endsubminipage` Closes the subminipage.

```
3885 \newcommand*{\LWR@endsubminipage}{%
3886 \LWR@stoppars%
3887 \LWR@endminipage% The following empty line is required:
3888
3889 }
```

Bool `LWR@minipagefullwidth` Should the next minipage have no HTML width?

```
3890 \newbool{LWR@minipagefullwidth}
3891 \boolfalse{LWR@minipagefullwidth}
```

`\minipagefullwidth` Requests that the next minipage have no width tag in HTML:

**for HTML output:** 3892 \newcommand\*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}  
3893 \end{warppHTML}

**for PRINT output:** 3894 \begin{warpprint}  
3895 \newcommand\*{\minipagefullwidth}{}  
3896 \end{warpprint}

**for HTML output:** 3897 \begin{warppHTML}

Bool `LWR@minipagethispar` Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
3898 \newbool{LWR@minipagethispar}
3899 \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.

```
3900 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}
3901 {%
```

Pre-compute the given width and height:

Reset the text area if are starting the outer-most minipage:

```

3902 \LWR@traceinfo{starting minipage of width #4}%
3903 \uselengthunit{in}%
3904 \setlength{\LWR@minipagewidth}{#4}%
3905 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
3906 \addtolength{\LWR@minipagewidth}{3em}% room for frames
3907 \setlength{\linewidth}{6in}%
3908 \setlength{\textwidth}{6in}%
3909 \setlength{\textheight}{9in}%
3910 }{}%
3911 \LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}
3912 \addtocounter{LWR@minipagedepth}{1}%
3913 \setlength{\LWR@minipageheight}{\textheight}% default unless specified
3914 \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.

```

3915 \LWR@stoppars%
```

Create the <div> tag with optional alignment style:

```

3916 \LWR@traceinfo{minipage: creating div class}%
3917 \LWR@htmltag{div class="minipage" style="%
3918 \ifthenelse{\equal{#1}{t}}{vertical-align: bottom ; }{}%
3919 \ifthenelse{\equal{#1}{c}}{vertical-align: middle ; }{}%
3920 \ifthenelse{\equal{#1}{b}}{vertical-align: top ; }{}%
3921 \ifthenelse{\equal{#3}{t}}{justify-content: flex-start ; }{}%
3922 \ifthenelse{\equal{#3}{c}}{justify-content: center ; }{}%
3923 \ifthenelse{\equal{#3}{b}}{justify-content: flex-end ; }{}%
3924 \ifthenelse{\equal{#3}{s}}{justify-content: space-between ; }{}%
```

Print the width and optional height styles:

```

3925 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
3926 \uselengthunit{PT}%
3927 \ifbool{LWR@minipagefullwidth}%
3928 {\boolfalse{LWR@minipagefullwidth}}%
3929 {%
3930 \ifthenelse{\lengthtest{#4}=\linewidth}%
3931 {}%
3932 {width:\rndprintlength{\LWR@minipagewidth} ; }%
3933 }%
3934 \LWR@traceinfo{minipage: about to print the height}%

```

```
3935 \IfValueTF{#2}{height:\rndprintlength{\LWR@minipageheight} ; }{}%
3936 "{}}%
```

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.

```
3937
3938 \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.

```
3939 \setlength{\linewidth}{#4}% the original width
3940 \setlength{\textwidth}{6in}%
3941 \setlength{\textheight}{9in}%
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
3942 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
3943 \LWR@startpars%
3944 \LWR@traceinfo{minipage: finished starting the minipage}%
3945 }
```

End the environment with L<sup>A</sup>T<sub>E</sub>X processing and closing tag:

```
3946 {%
3947 \LWR@stoppars%
3948 \LWR@endminipage% The following empty line is required:
3949
3950 \LWR@htmldivclassend{minipage}%
3951 \vspace{1\baselineskip}% required for subcaption
3952 \addtocounter{\LWR@minipagedepth}{-1}%
3953 \LWR@startpars%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
3954 \global\booltrue{\LWR@minipagethispar}%
3955 }
```

## 61.4 Parbox, makebox, framebox, fbox, raisebox

`\parbox` [*<pos>*] [*<height>*] [*<inner-pos>*] {*<width>*} {*<text>*}

A parbox uses the minipage code:

```
3956 \RenewDocumentCommand{\parbox}{0{t} o 0{t} m +m}
3957 {
3958 \LWR@traceinfo{parbox of width #4}%
3959 \begin{minipage}[#1][#2][#3]{#4}
3960 #5
3961 \end{minipage}
3962 }
```

`\makebox` [*<width>*] [*<pos>*] {*<text>*}

Width and position are ignored.

```
3963 \let\LWR@origmakebox\makebox
3964
3965 \RenewDocumentCommand{\makebox}{o o m}{%
3966 \mbox{#3}
3967 }
```

`\framebox` [*<width>*] [*<pos>*] {*<text>*}

Width and position are ignored.

```
3968 \RenewDocumentCommand{\framebox}{o o m}{%
3969 \fbox{#3}
3970 }
```

`\fbox` {*<text>*}

```
3971 \let\LWR@origfbox\fbox
3972 %
3973 \renewcommand*\fbox[1]{%
3974 \InlineClass{framebox}{#1}%
3975 }
```

`\raisebox` {*<raiselen>*} [*<height>*] [*<depth>*] {*<text>*}

```
3976 \RenewDocumentCommand{\raisebox}{m o o m}{%
3977 #4%
3978 }

3979 \end{warpHTML}
```



## 62 Direct formatting

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

For high-level block and inline custom CSS classes, see section 29.7.

for HTML output: 3980 `\begin{warpHTML}`

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

3981 `\renewcommand{\emph}[1]{\LWR@htmlspan{em}{#1}}`

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

3982 `\renewcommand{\textmd}[1]{\LWR@htmlspan{textmd}{#1}}`

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

3983 `\renewcommand{\textbf}[1]{\LWR@htmlspan{b}{#1}}`

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

3984 `\renewcommand{\textrm}[1]{\InlineClass{textrm}{#1}}`

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

3985 `\renewcommand{\textsf}[1]{\InlineClass{textsf}{#1}}`

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

3986 `\renewcommand{\texttt}[1]{\LWR@htmlspan{kbd}{#1}}`

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

3987 `\renewcommand{\textup}[1]{\LWR@htmlspan{textup}{#1}}`

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

3988 `\renewcommand{\textit}[1]{\LWR@htmlspan{i}{#1}}`

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

3989 `\renewcommand{\textsc}[1]{\InlineClass{textsc}{#1}}`

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

3990 `\renewcommand{\textnormal}[1]{\textmd{\textrm{\textup{#1}}}}`

`\mdseries`

3991 `\renewcommand*{\mdseries}{}`

`\bfseries`

3992 `\renewcommand*{\bfseries}{}`

`\rmfamily`

3993 `\renewcommand*{\rmfamily}{}`

`\sffamily`

3994 `\renewcommand*{\sffamily}{}`

`\ttfamily`

3995 `\renewcommand*{\ttfamily}{}`

`\upshape`

3996 `\renewcommand*{\upshape}{}`

`\itshape`

3997 `\renewcommand*{\itshape}{}`

`\scshape`

3998 `\renewcommand*{\scshape}{}`

`\scshape`

3999 `\renewcommand*{\normalfont}{}`

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

For siunitx. Must work in math mode.

4000 `\renewcommand{\sp}[1]{\text{<sup>#1</sup>}}`

`\sb`  $\langle text \rangle$

For siunitx. Must work in math mode.

```
4001 \renewcommand{\sb}[1]{\text{<sub>#1</sub>}{}}
```

`\textsuperscript`  $\langle text \rangle$

```
4002 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

`\textsubscript`  $\langle text \rangle$

```
4003 \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.

```
4004 \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.

```
4005 \AtBeginDocument{\let\fup\textsuperscript}
```

`\hfill`

```
4006 \renewcommand*{\hfill}{\quad}
```

`\hrulefill`

```
4007 \renewcommand*{\hrulefill}{\rule{1in}{1pt}}
```

`\dotfill`

```
4008 \renewcommand*{\dotfill}{\dots}
```

```
4009 \end{warpHTML}
```

## 63 Skips, spaces, font sizes

for HTML output: 4010 \begin{warpHTML}

\, must be redefined after \RequirePackage{printlen}

```
4011 \let\LWR@origcomma\,
4012 \let\LWR@origtilde~
4013 \let\LWR@origenskip\enskip
4014 \let\LWR@origquad\quad
4015 \let\LWR@origqqquad\qqquad
4016 \let\LWR@orighspace\hspace
4017 \let\LWR@origrule\rule
4018 \let\LWR@origmedskip\medskip
```

Direct-formatting space commands become HTML entities:

```
4019 \renewcommand*{\,}{\HTMLUnicode{202f}} % HTML thin non-breakable space
4020 \renewcommand*{~}{\HTMLentity{nbsp}}
4021 \renewcommand*{\textellipsis}{\HTMLUnicode{2026}}
```

Direct-formatting font sizes are ignored:

```
4022 \let\LWR@orignormalsize\normalsize
4023 \let\LWR@origsmall\small
4024 \let\LWR@origfootnotesize\footnotesize
4025 \let\LWR@origscriptsize\scriptsize
4026 \let\LWR@origtiny\tiny
4027 \let\LWR@origlarge\large
4028 \let\LWR@origLarge\Large
4029 \let\LWR@origLARGE\LARGE
4030 \let\LWR@orighuge\huge
4031 \let\LWR@origHuge\Huge
4032 \renewcommand*{\normalsize}{}
4033 \renewcommand*{\small}{}
4034 \renewcommand*{\footnotesize}{}
4035 \renewcommand*{\scriptsize}{}
4036 \renewcommand*{\tiny}{}
4037 \renewcommand*{\large}{}
4038 \renewcommand*{\Large}{}
4039 \renewcommand*{\LARGE}{}
4040 \renewcommand*{\huge}{}
4041 \renewcommand*{\Huge}{}
4042
4043 \renewcommand*{\onecolumn}{}
4044
4045 \renewcommand{\twocolumn}[1][{}]{
```

```

4046
4047 #1
4048
4049 }

```

`\newline` Uses HTML `<br />` tag

```

4050 \newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
4051 \let\n 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`.

```

4052 \NewDocumentCommand{\LWR@endoffline}{s o}
4053 {%
4054 \newline%
4055 }

```

`\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.

`\hspace`  
`\enskip`  
`\quad`  
`\qquad`

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.

```

4056 \newcommand*{\LWR@minipagestartpars}{%
4057 \ifbool{LWR@minipagethispar}%
4058 {%
4059 \LWR@startpars%

```

```
4060 }{}%
4061 }
```

`\LWR@minipagestoppars` Placed just after `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
4062 \newcommand*{\LWR@minipagestoppars}{%
4063 \ifbool{LWR@minipagethispar}%
4064 {%
4065 \LWR@stoppars%
4066 }{}%
4067 }
```

`\quad` Handles special minipage & horizontal space interactions.

```
4068 \renewcommand*{\quad}{%
4069 \LWR@minipagestoppars%
4070 \HTMLUnicode{2001}%
4071 \LWR@minipagestartpars%
4072 }
```

`\qquad` Handles special minipage & horizontal space interactions.

```
4073 \renewcommand*{\qquad}{\quad\quad}
```

`\enskip` Handles special minipage & horizontal space interactions.

```
4074 \renewcommand*{\enskip}{%
4075 \LWR@minipagestoppars%
4076 \HTMLUnicode{2000}%
4077 \LWR@minipagestartpars%
4078 }
```

Len `\WR@tempwidth` Used to compute span width, height, raise for `\hspace` and `\rule`:

```
Len \WR@tempheight 4079 \newlength{\LWR@tempwidth}
Len \WR@tempraise 4080 \newlength{\LWR@tempheight}
4081 \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`.

```
4082 \NewDocumentCommand{\LWR@hspace}{s m}{%
4083 \setlength{\LWR@tempwidth}{#2}%
```

If \fill, change to \qqquad:

```
4084 \ifnum\gluestretchorder\LWR@tempwidth>0%
4085 \setlength{\LWR@tempwidth}{2em}%
4086 \fi%
```

Only if the width is not zero:

```
4087 \ifthenelse{\dimtest{\LWR@tempwidth}={}\{0pt\}}{ }{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4088 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
4089 \ifthenelse{\dimtest{\LWR@tempwidth}={}\{.16667em\}}
4090 {%
4091 \HTMLUnicode{2009}% thin breakable space
4092 }%
```

Print the span with the converted width. Not rounded.

```
4093 {%
4094 \uselengthunit{PT}%
4095 \LWR@htmltagc{%
4096 span style="width:\printlength{\LWR@tempwidth}; display:inline-block;"%
4097 }%
4098 \LWR@htmltagc{/span}%
4099 }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4100 \LWR@minipagestartpars%
4101 }%
4102 }
```

`\hspace` \*  $\langle length \rangle$

Handles special minipage & horizontal space interactions.

```
4103 \let\hspace\LWR@hspace
```

`\linebreak` [ $\langle num \rangle$ ] Inserts an HTML `br` tag.

```
4104 \renewcommand*{\linebreak}[1] [] {\newline}
```

`\nolinebreak` [ $\langle num \rangle$ ]

```
4105 \renewcommand*{\nolinebreak}[1] [] {}
```

`\pagebreak` [ $\langle num \rangle$ ] Starts a new paragraph.

```
4106 \renewcommand*{\pagebreak}[1] [] {
4107
4108 }
```

`\nopagebreak` [ $\langle num \rangle$ ]

```
4109 \renewcommand*{\nopagebreak}[1] [] {}
```

`\enlargethispage` \*  $\{\langle len \rangle\}$

```
4110 \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.

```
4111 \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:

```
4112 \setlength{\LWR@tempwidth}{#2}%%
```

If it's zero-width then skip the entire rule:

```
4113 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}{
4114 {}}% zero- width
4115 {}% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:



```

4116 \ifthenelse{\lengthtest{\LWR@tempwidth>0pt}\AND%
4117 \lengthtest{\LWR@tempwidth<1pt}}%
4118 {\setlength{\LWR@tempwidth}{1pt}}{}%

```

Likewise with height:

```

4119 \setlength{\LWR@tempheight}{#3}%
4120 \ifthenelse{\lengthtest{\LWR@tempheight>0pt}\AND%
4121 \lengthtest{\LWR@tempheight<1pt}}%
4122 {\setlength{\LWR@tempheight}{1pt}}{}%

```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```

4123 \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.

```

4124 \uselengthunit{PT}%
4125 \LWR@htmltagc{%
4126 span
4127 style=" %

```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```

4128 background:\LWR@currenttextcolor; %

```

The width and height are printed, converted to PT:

```

4129 width:\printlength{\LWR@tempwidth}; %
4130 height:\printlength{\LWR@tempheight}; %

```

The raise height is converted to a CSS transform. The `*2` raise multiplier is to approximately match HTML output's X height. Conversion to a L<sup>A</sup>T<sub>E</sub>X length allows a typical L<sup>A</sup>T<sub>E</sub>X expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a L<sup>A</sup>T<sub>E</sub>X length limits the allowable syntax. To do: A superior method would compute a ratio of L<sup>A</sup>T<sub>E</sub>X ex height, then print that to HTML with an ex unit.

```

4131 \IfValueTF{#1}%
4132 {%
4133 \setlength{\LWR@tempraise}{Opt-#1}%
4134 \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
4135 \LWR@orignewline%
4136 -ms-transform: translate(0pt,\printlength{\LWR@tempraise}); %
4137 \LWR@orignewline%

```

```

4138 -webkit-transform: translate(0pt,\printlength{\LWR@tempraise}); %
4139 \LWR@orignewline%
4140 transform: translate(0pt,\printlength{\LWR@tempraise}); %
4141 \LWR@orignewline%
4142 }{}%

```

Display inline-block to place the span inline with the text:

```

4143 display:inline-block;"%
4144 }%
4145 \LWR@htmltagc{/span}%

```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

4146 \LWR@minipagestartpars%
4147 }% non-zero width
4148 }

```

`\rule` [*raise*] {*width*} {*height*}

Handles special minipage & horizontal space interactions.

```

4149 \let\rule\LWR@rule
4150 \end{warpHTML}

```

## 64 \phantomsection

**for HTML output:** 4151 \begin{warpHTML}

`\phantomsection` Emulate the hyperref `\phantomsection` command, often used to insert the bibliography into table of contents:

```

4152 \newcommand*\phantomsection{\section*{}}
4153 \end{warpHTML}

```

## 65 \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>

## 65.1 HTML logos

for HTML output: 4154 `\begin{warpHTML}`

`\TeX`  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$

`latexlogo` is a CSS class used to properly typeset the E and A in  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  and friends.

`latexlogofont` is a CSS class used to select the font for the rest of the logo in  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{LuaT}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{ConT}_{\mathrm{E}}\mathrm{X}$ t, etc.

```
4155 \newcommand*{\LWR@TeX}
4156 {\InlineClass{latexlogofont}%
4157 {\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
4158 \newcommand*{\LWR@LaTeX}
4159 {\InlineClass{latexlogofont}%
4160 {\InlineClass{latexlogo}%
4161 {L\textsuperscript{a}T\textsubscript{e}X}}}}
4162
4163 \renewcommand*{\LaTeXe}
4164 {\LaTeX\InlineClass{latexlogofont}%
4165 {\,2\textsubscript{\textit{\HTMLUnicode{3B5}}}}}
```

`\LuaTeX`  $\mathrm{LuaT}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{LuaL}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$

```
\LuaLaTeX
4166 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
4167 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

`\XeTeX`  $\mathrm{X}_{\mathrm{E}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{X}_{\mathrm{E}}\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$

`\XeLaTeX` `xetexlogo` is a CSS class which aligns the backwards E in  $\mathrm{X}_{\mathrm{E}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$  and spaces  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  appropriately.

`xelatexlogo` is a CSS class which aligns the backwards E in  $\text{Xe}\text{L}\text{A}\text{T}\text{E}\text{X}$  and spaces  $\text{L}\text{A}\text{T}\text{E}\text{X}$  appropriately.

```
4168 \newcommand*{\Xe}
4169   {X\textsubscript{\HTMLUnicode{18e}}}%
4170 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
4171 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}
```

`\ConTeXt`    $\text{ConT}\text{E}\text{Xt}$

```
4172 \newcommand*{\LWR@ConTeXt}
4173 {\InlineClass{latexlogofont}{Con}\TeX{}}%
4174 \InlineClass{latexlogofont}{t}}
```

`\BibTeX`    $\text{BIBT}\text{E}\text{X}$ , *MakeIndex*  
`\MakeIndex`

```
4175 \providecommand*{\BibTeX}
4176 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}
4177
4178 \newcommand*{\MakeIndex}
4179 {\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.

```
4180 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}}
4181 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}
```

`\MiKTeX`    $\text{MiK}\text{T}\text{E}\text{X}$

```
4182 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}
```

`\LyX`    $\text{Ly}\text{X}$

`lyxlogo` is a CSS class used for the  $\text{Ly}\text{X}$ logo.

```
4183 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
```

```
4184 \end{warpHTML}
```

## 65.2 Print logos

for PRINT output: 4185 \begin{warpprint}

```

4186 \newcommand*\XeTeXrevE{
4187   {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
4188 \providecommand*\XeTeX{\mbox{X\XeTeXrevE\TeX}}
4189 \providecommand*\XeLaTeX{\mbox{X\XeTeXrevE\LaTeX}}
4190 \providecommand*\AMS{%
4191 \leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex%
4192 \hbox{$\mathcal M$}\kern-.2em\mathcal S$}}
4193 \newcommand*\LyX{\textsf{LyX}}
4194 \providecommand*\LuaTeX{\mbox{Lua\TeX}}
4195 \providecommand*\LuaLaTeX{\mbox{Lua\LaTeX}}
4196 \providecommand*\BibTeX{\mbox{B\textsc{ib}\TeX}}
4197 \providecommand*\MakeIndex{\mbox{\textit{MakeIndex}}}
4198 \providecommand*\ConTeXt{\mbox{Con\TeXt}}
4199 \providecommand*\MiKTeX{\mbox{MiK\TeX}}
4200 \end{warpprint}

```

## 66 \AtBeginDocument, \AtEndDocument

for HTML output: 4201 \begin{warppHTML}

\LWR@LwarpStart Automatically sets up the HTML-related actions for the start and end of the  
 \LWR@LwarpEnd document.

```

4202 \AfterEndPreamble{\LWR@LwarpStart}
4203 \AtEndDocument{\LWR@LwarpEnd}

4204 \end{warppHTML}

```

## 67 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
```

---

Defaults unless already over-ridden by the user:

```
4205 \providecommand*\HomeHTMLFileName{\BaseJobname}
4206 \providecommand*\HTMLFileName{}
4207 \providecommand*\UseLatexmk{false}
```

for PRINT output:

```
4208 \begin{warpprint}
4209 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
4210 \immediate\openout{LWR@file}=lwarpmk.conf
4211 \ifwindows
4212 \immediate\write{LWR@file}{opsystem = "Windows"}
4213 \else
4214 \immediate\write{LWR@file}{opsystem = "Unix"}
4215 \fi
4216 \ifPDFTeX
4217 \immediate\write{LWR@file}{latexname = "pdflatex"}
4218 \fi
4219 \ifXeTeX
4220 \immediate\write{LWR@file}{latexname = "xelatex"}
4221 \fi
4222 \ifLuaTeX
4223 \immediate\write{LWR@file}{latexname = "lualatex"}
4224 \fi
4225 \immediate\write{LWR@file}{sourcename = "\jobname"}
4226 \immediate\write{LWR@file}{%
4227 homehtmlfilename = "\HomeHTMLFileName"%
4228 }
4229 \immediate\write{LWR@file}{htmlfilename = "\HTMLFileName"}
4230 \immediate\write{LWR@file}{uselatexmk = "\UseLatexmk"}
4231 \immediate\closeout{LWR@file}
4232 \end{warpprint}
```

## 68 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-newproject.sty

## 69 lwarp-newproject

Pkg	<b>lwarp-newproject</b>	lwarp-newproject is used to create lwarp-specific system files in a new project.
Opt lwarp-newproject	<b>warpprint</b>	Configuration files are only created if the <b>warpprint</b> option was used. When this is the case, lwarp-newproject knows that <code>\jobname</code> is the source code's filename (as opposed to the <code>&lt;project&gt;_html.tex</code> filename).
Opt lwarp-newproject	<b>warpHTML</b>	Does not create configuration files. <b>warpprint</b> or <b>warpHTML</b> are passed to both lwarp and lwarp-newproject by lwarpmk, depending on whether lwarpmk <b>print</b> or <b>html</b> was used.
Opt lwarp-newproject	<b>BaseJobName</b>	<code>BaseJobname=somename</code> sets <code>\BaseJobname</code> to <code>somename</code> , which is used to set <code>\HomeHTMLFileName</code> , which is then written to the <code>lwarpmk.conf</code> and <code>&lt;project&gt;.lwarpmkconf</code> configuration files.  <code>\BaseJobname</code> is also written to <code>&lt;project&gt;_html.tex</code> as an option given for the lwarp and lwarp-newproject packages during the creation of HTML output.  <code>\BaseJobname</code> is the <code>\jobname</code> of the printed version, even while compiling the HTML version, where <code>\jobname</code> has <code>_html</code> appended.
Opt lwarp-newproject	<b>lwarpmk</b>	Tells lwarp-newproject to generate a local copy of lwarpmk called <code>lwarpmk.lua</code> . Useful for archiving for future use. This file may be made executable and acts just like <code>lwarpmk</code> .

In the document source:

---

```

\documentclass{article} % or book, report
...
(font selection, input encoding)
...
\newcommand*{\HomeHTMLFileName}{index}, or {projectname}
\newcommand*{\HTMLFileName}{}
\usepackage{lwarp-newproject}
\usepackage{lwarp}
...
(load other packages, the rest of the preamble)
...
\NewCSS{project.css}
\NewHTMLdescription{Summarize this webpage.}

```



---

```

...
\begin{document}
...
\end{document}

```

---

Place `\usepackage{lwarp-newproject}` just before `\usepackage{lwarp}` in the document source. The operating system and  $\TeX$  engine will be auto-detected, and a customized `lwarpmk.conf` file will be generated, along with the other support files (`*.css`, `lwarp_html.xdy`, `lwarp_mathjax.txt`). Rename `sample_project.css` to your own custom `projectname.css` and edit it if desired.

After the first compile, `\usepackage{lwarp-newproject}` may be commented out unless the configuration changes. For example, it may be re-enabled to switch between `pdflatex`, `xelatex`, and `lualatex`, in which case a new manual compile (without using `lwarpmk`) will auto-configure and regenerate the files. It may also be left enabled at all times, in which case the configuration files will be regenerated each time. Note that your own `projectname.css` file will not be over-written by the newly regenerated `sample_project.css`.

Discard all options for `lwarp-newproject`:

```

1 \ProvidesPackage{lwarp-newproject}
2
3 \RequirePackage{etoolbox}
4 \RequirePackage{comment}
5 \RequirePackage{fancyvrb}
6 \RequirePackage{ifplatform}% sense op-system platform
7 \RequirePackage{iftex}% sense pdflatex/lualatex/xelatex
8 \RequirePackage{kvoptions}
9 \SetupKeyvalOptions{family=LWRNP,prefix=LWRNP@}

```

Generate config files?

```

10 \newbool{LWRNP@genconf}
11 \boolfalse{LWRNP@genconf}

```

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

```

12 \excludecomment{LWR@createlwarpmk}

```

Process options:

```

13 \DeclareVoidOption{warpprint}{\booltrue{LWRNP@genconf}}
14 \DeclareVoidOption{warpHTML}{\boolfalse{LWRNP@genconf}}
15 \DeclareVoidOption{lwarpmk}{\includecomment{LWR@createlwarpmk}}
16 \DeclareStringOption[\jobname]{BaseJobname}
17

```

```

18 \booltrue{LWRNP@genconf}% warpprint
19
20 \ProcessKeyvalOptions*\relax

```

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

```

21 \providecommand*{\BaseJobname}{\LWRNP@BaseJobname}

```

Defaults unless already over-ridden by the user:

```

22 % \ifbool{LWRNP@genconf}{
23 \providecommand*{\HomeHTMLFileName}{\BaseJobname}
24 % }{}
25 \providecommand*{\HTMLFileName}{%
26 \providecommand*{\UseLatexmk}{false}

```

## 69.1 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.

```

27 \ifbool{LWRNP@genconf}{
28 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
29 \immediate\openout{LWR@file}=\jobname_html.tex
30 \immediate\write{LWR@file}{%
31 \detokenize{\PassOptionsToPackage}%
32 {warpHTML,BaseJobname=\jobname}{lwarp}%
33 }
34 \immediate\write{LWR@file}{%
35 \detokenize{\PassOptionsToPackage}%
36 {warpHTML,BaseJobname=\jobname}{lwarp-newproject}%
37 }
38 \immediate\write{LWR@file}{%
39 \detokenize{\input}\string{\jobname.tex}\string }%
40 }
41 \immediate\closeout{LWR@file}
42 }{}

```

## 69.2 project.lwarpmkconf

File `project.lwarpmkconf` The configuration file for `lwarpmk`.

```

43 \ifbool{LWRNP@genconf}{
44 \ifcsdef{LWR@file}{\newwrite{LWR@file}}

```

```

45 \immediate\openout\LWR@file=\jobname.lwarpmkconf
46 \ifwindows
47 \immediate\write\LWR@file{opsystem = "Windows"}
48 \else
49 \immediate\write\LWR@file{opsystem = "Unix"}
50 \fi
51 \ifPDFTeX
52 \immediate\write\LWR@file{latexname = "pdflatex"}
53 \fi
54 \ifXeTeX
55 \immediate\write\LWR@file{latexname = "xelatex"}
56 \fi
57 \ifLuaTeX
58 \immediate\write\LWR@file{latexname = "lualatex"}
59 \fi
60 \immediate\write\LWR@file{sourcename = "\jobname"}
61 \immediate\write\LWR@file{%
62 homehtmlfilename = "\HomeHTMLFileName"%
63 }
64 \immediate\write\LWR@file{htmlfilename = "\HTMLFileName"}
65 \immediate\write\LWR@file{uselatemk = "\UseLatexmk"}
66 \immediate\closeout\LWR@file
67 }{}

```

### 69.3 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.

```

68 \begin{VerbatimOut}{lwarp.css}
69 /*
70  CSS stylesheet for the LaTeX lwarp package
71  Copyright 2016-2017 Brian Dunn -- BD Tech Concepts LLC
72 */
73
74
75 /* a fix for older browsers: */
76 header, section, footer, aside, nav, main,
77  article, figure { display: block; }
78
79
80 A:link {color:#000080 ; text-decoration: none ; }
81 A:visited {color:#800000 ; }
82 A:hover {color:#000080 ; text-decoration: underline ;}
83 A:active {color:#800000 ; }

```

```
84
85 a.tocpart {display: inline-block ; margin-left: 0em ;
86     font-weight: bold ;}
87 a.tocchapter {display: inline-block ; margin-left: 0em ;
88     font-weight: bold ;}
89 a.tocsection {display: inline-block ; margin-left: 1em ;
90     text-indent: -.5em ; font-weight: bold ; }
91 a.tocsubsection {display: inline-block ; margin-left: 2em ;
92     text-indent: -.5em ; }
93 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
94     text-indent: -.5em ; }
95 a.tocparagraph {display: inline-block ; margin-left: 4em ;
96     text-indent: -.5em ; }
97 a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
98     text-indent: -.5em ; }
99 a.tocfigure {margin-left: 0em}
100 a.tocsubfigure {margin-left: 2em}
101 a.toctable {margin-left: 0em}
102 a.tocsubtable {margin-left: 2em}
103 a.toctheorem {margin-left: 0em}
104 a.toclstlisting {margin-left: 0em}
105
106
107 body {
108     font-family: "DejaVu Serif", "Bitstream Vera Serif",
109         "Lucida Bright", Georgia, serif;
110     background: #FAF7F4 ;
111     color: black ;
112     margin: 0em ;
113     padding: 0em ;
114     font-size: 100% ;
115     line-height: 1.2 ;
116 }
117
118 p {margin: 1.5ex 0em 1.5ex 0em ;}
119
120 /* Holds a section number to add space between it and the name */
121 span.sectionnumber { margin-right: .6em }
122
123 /* Inserted in front of index lines */
124 span.indexitem {margin-left: 0em}
125 span.indexsubitem {margin-left: 2em}
126 span.indexsubsubitem {margin-left: 4em}
127
128 div.hidden { display: none ; }
129
130 kbd {
131     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
132         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
133         "Courier New", monospace;
```

```
134     font-size: 100% ;
135 }
136
137 span.strong { font-weight: bold; }
138
139 span.textmd { font-weight: normal; }
140
141 span.textsc { font-variant: small-caps; }
142
143 span.textup { font-variant: normal; }
144
145 span.textrm {
146     font-family: "DejaVu Serif", "Bitstream Vera Serif",
147     "Lucida Bright", Georgia, serif;
148 }
149
150 span.textsf {
151     font-family: "DejaVu Sans", "Bitstream Vera Sans",
152     Geneva, Verdana, sans-serif ;
153 }
154
155 span.attribution {
156     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
157 }
158
159 span.citetitle {
160     margin-left: 1em ; font-size: 80% ; font-style: oblique;
161 }
162
163 span.poemtitle {
164     font-size: 120% ; font-weight: bold;
165 }
166
167 blockquote {
168     margin-left: 0px ;
169     margin-right: 0px ;
170 }
171
172 blockquote p {
173     line-height: 1.5;
174     text-align: left ;
175     font-size: .85em ;
176     margin-left: 3em ;
177     margin-right: 3em ;
178 }
179
180 blockquotation {
181     margin-left: 0px ;
182     margin-right: 0px ;
183 }
```

```
184
185 blockquotation p {
186   line-height: 1.5;
187   text-align: left ;
188   font-size: .85em ;
189   margin-left: 3em ;
190 margin-right: 3em ;
191 }
192
193 div.epigraph {
194   line-height: 1.2;
195   text-align: left ;
196   padding: 3ex 1em 0ex 1em ;
197 /*   margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
198   margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
199 /*   margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
200   font-size: .85em ;
201   max-width: 27em ;
202 }
203
204
205
206 div.epigraphsource{
207   text-align:right ;
208   margin-left:auto ;
209 /*   max-width: 50% ; */
210   border-top: 1px solid #A0A0A0 ;
211   padding-bottom: 3ex ;
212   line-height: 1.2;
213 }
214
215 div.epigraph p { padding: .5ex ; margin: 0ex ;}
216 div.epigraphsource p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
217
218
219 /* lettrine package: */
220 span.lettrine { font-size: 3ex ; float: left ; }
221 span.lettrinetext { font-variant: small-caps ; }
222
223 /* ulem and soul packages: */
224 span.uline {
225   text-decoration: underline ;
226   text-decoration-skip ;
227 }
228
229 span.uuline {
230   text-decoration: underline ;
231   text-decoration-skip ;
232   text-decoration-style: double ;
233 }
```

```
234
235 span.uwave {
236     text-decoration: underline ;
237     text-decoration-skip ;
238     text-decoration-style: wavy ;
239 }
240
241 span.sout {
242     text-decoration: line-through ;
243 }
244
245 span.xout {
246     text-decoration: line-through ;
247 }
248
249 span.dashuline {
250     text-decoration: underline ;
251     text-decoration-skip ;
252     text-decoration-style: dashed ;
253 }
254
255 span.dotuline {
256     text-decoration: underline ;
257     text-decoration-skip ;
258     text-decoration-style: dotted ;
259 }
260
261 span.letterspacing { letter-spacing: .2ex ; }
262
263 span.capsspacing {
264     font-variant: small-caps ;
265     letter-spacing: .1ex ;
266 }
267
268 span.highlight { background: #F8E800 ; }
269
270
271
272
273 html body {
274     margin: 0 ;
275     line-height: 1.2;
276 }
277
278
279 body div {
280     margin: 0ex;
281 }
282
283
```

```
284 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
285 {
286     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
287         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
288         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
289         "Times New Roman", serif;
290     font-style: normal ;
291     font-weight: bold ;
292     text-align: left ;
293 }
294
295 h1 { /* title of the entire website, used on each page */
296     text-align: center ;
297     font-size: 2.5em ;
298     padding: .4ex 0em 0ex 0em ;
299 }
300 h2 { font-size: 2.25em }
301 h3 { font-size: 2em }
302 h4 { font-size: 1.75em }
303 h5 { font-size: 1.5em }
304 h6 { font-size: 1.25em }
305 span.paragraph {font-size: 1em ; font-variant: normal ;
306     margin-right: 1em ; }
307 span.subparagraph {font-size: 1em ; font-variant: normal ;
308     margin-right: 1em ; }
309
310
311
312 /* Title of the file */
313 h1 {
314     margin: 0ex 0em 0ex 0em ;
315     line-height: 1.3;
316     text-align: center ;
317 }
318
319 /* Part */
320 h2 {
321     margin: 1ex 0em 1ex 0em ;
322     line-height: 1.3;
323     text-align: center ;
324 }
325
326 /* Chapter */
327 h3 {
328     margin: 3ex 0em 1ex 0em ;
329     line-height: 1.3;
330 }
331
332 /* Section */
333 h4 {
```



```
334 margin: 3ex 0em 1ex 0em ;
335 line-height: 1.3;
336 }
337
338 /* Sub-Section */
339 h5 {
340 margin: 3ex 0em 1ex 0em ;
341 line-height: 1.3;
342 }
343
344 /* Sub-Sub-Section */
345 h6 {
346 margin: 3ex 0em 1ex 0em ;
347 line-height: 1.3;
348 }
349
350
351 div.titlepage {
352 text-align: center ;
353 }
354
355 .footnotes {
356 font-size: .85em ;
357 margin: 3ex 1em 0ex 1em ;
358 padding-bottom: 1ex ;
359 border-top: 1px solid silver ;
360 }
361
362 .marginpar {
363 max-width:50%;
364 float:right;
365 text-align:left;
366 margin: 1ex 0.5em 1ex 1em ;
367 padding: 1ex 0.5em 1ex 0.5em ;
368 font-size: 85% ;
369 border-top: 1px solid silver ;
370 border-bottom: 1px solid silver ;
371 overflow-x: auto;
372 }
373
374 .marginpar br { margin-bottom: 2ex ; }
375
376 div.marginblock {
377 max-width:50%;
378 float:right;
379 text-align:left;
380 margin: 1ex 0.5em 1ex 1em ;
381 padding: 1ex 0.5em 1ex 0.5em ;
382 overflow-x: auto;
383 }
```

```
384
385 div.marginblock div.minipage {
386     display: block ;
387     margin: 0pt auto 0pt auto ;
388 }
389
390 div.marginblock div.minipage p { font-size: 85%}
391
392 div.marginblock br { margin-bottom: 2ex ; }
393
394
395 section.textbody div.footnotes{
396     margin: 3ex 0em 0ex 0em ;
397     border-bottom: 2px solid silver ;
398 }
399
400 .footnoteheader {
401     border-top: 2px solid silver ;
402     margin-top: 3ex ;
403     padding-top: 1ex ;
404     font-weight: bold ;
405 }
406
407 .mpfootnotes {
408     text-align: left ;
409     font-size: .85em ;
410     margin-left: 1em ;
411     border-top: 1px solid silver ;
412 }
413
414 /* Remove footnote top border in the title page. */
415 div.titlepage div.mpfootnotes {
416     border-top: none ;
417 }
418
419
420
421 ol {
422     margin: 1ex 1em 1ex 0em;
423     line-height: 1.2;
424 }
425
426 ul, body dir, body menu {
427     margin: 1ex 1em 1ex 0em;
428     line-height: 1.2;
429 }
430
431 li { margin: 0ex 0em 1ex 0em; }
432
433 html {
```

```
434 margin: 0;
435 padding: 0;
436 }
437
438 .programlisting {
439 font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
440             "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
441             "Courier New", monospace;
442 margin: 1ex 0ex 1ex 0ex ;
443 padding: .5ex 0pt .5ex 0pt ;
444 overflow-x: auto;
445 }
446
447 section.textbody>pre.programlisting {
448 border-top: 1px solid silver ;
449 border-bottom: 1px solid silver ;
450 }
451
452
453 .inlineprogramlisting {
454 font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
455             "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
456             "Courier New", monospace;
457 overflow-x: auto;
458 }
459
460
461 div.abstract {
462 margin: 2em 5% 2em 5% ;
463 padding: 1ex 1em 1ex 1em ;
464 /* font-weight: bold ; */
465 font-size: 90% ;
466 }
467
468 div.abstract dl {line-height:1.5;}
469 div.abstract dt {color:#304070;}
470
471 div.abstracttitle{
472 font-family: "URW Classico", Optima, "Linux Biolinum O",
473             "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
474             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
475 font-weight:bold;
476 font-size:1.25em;
477 text-align: center ;
478 }
479
480 span.abstracrunintitle{
481 font-family: "URW Classico", Optima, "Linux Biolinum O",
482             "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
483             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
```

```
484     font-weight:bold;
485 }
486
487
488 .verbatim {
489     overflow-x: auto ;
490 }
491
492 .alltt {
493     overflow-x: auto ;
494 }
495
496
497 .bverbatim {
498     margin: 1ex Opt 1ex Opt ;
499     padding: .5ex Opt .5ex Opt ;
500     overflow-x: auto ;
501 }
502
503 .lverbatim {
504     margin: 1ex Opt 1ex Opt ;
505     padding: .5ex Opt .5ex Opt ;
506     overflow-x: auto ;
507 }
508
509 .fancyvrb {
510     font-size:.85em ;
511     margin: 3ex Opt 3ex Opt
512 }
513
514 .fancyvrblabel {
515     font-weight:bold;
516     text-align: center ;
517 }
518
519
520 .verse {
521     font-family: "Linux Libertine Mono O", "Lucida Console",
522                 "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
523                 "Liberation Mono", "FreeMono", "Andale Mono",
524                 "Nimbus Mono L", "Courier New", monospace;
525     margin-left: 1em ;
526 }
527
528
529 div.singlespace { line-height: 1.2 ; }
530 div.onehalfspace { line-height: 1.5 ; }
531 div.doublespace { line-height: 2 ; }
532
533
```

```
534
535
536
537 /* Minipage environments, vertically aligned to top, center, bottom: */
538 .minipage {
539     /* display: inline-block ; */
540     /* Mini pages which follow each other will be tiled. */
541     margin: .25em .25em .25em .25em;
542     padding: .25em .25em .25em .25em;
543     display: inline-flex;
544     flex-direction: column ;
545     overflow: auto;
546 }
547
548 /* Paragraphs in the flexbox did not collapse their margins. */
549 /* Have not yet researched this. */
550 .minipage p {margin: .75ex 0em .75ex 0em ;}
551
552
553
554 .framebox {
555     margin: 0ex ;
556     padding: 0ex ;
557     border: 1px solid black;
558     border-radius: 0px ;
559     padding: .3ex .2em 0ex .2em ;
560     margin: .1ex ;
561     display: inline-block ;
562 }
563
564
565 .mdframed {
566 /*     padding: 0ex ; */
567 /*     border: 1px solid black; */
568 /*     border-radius: 0px ; */
569     padding: 0ex ;
570     margin: 3ex 5% 3ex 5% ;
571 /*     display: inline-block ; */
572 }
573
574 .mdframed p { padding: 0ex .5em 0ex .5em ; }
575
576 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
577
578 .mdframedtitle {
579     padding: .5em ;
580     display: block ;
581     font-size: 130%
582 }
583
```

```
584 .mdframedsubtitle {
585     padding: 0ex .5em 0ex .5em ;
586     display: block ;
587     font-size: 115% ;
588 }
589
590 .mdframedsubsubtitle {
591     padding: 0ex .5em 0ex .5em ;
592     display: block ;
593 }
594
595 .mdtheorem {
596     padding: 0ex .5em 0ex .5em ;
597     margin: 3ex 5% 3ex 5% ;
598 /*     display: inline-block ; */
599 }
600
601
602 /* framed package */
603 .framed {
604     margin: 3ex 0em 3ex 0em ;
605     border: 1px solid black;
606     border-radius: 0px ;
607     padding: .3ex 1em 0ex 1em ;
608     display: block ;
609 }
610
611 .snugframed {
612     margin: 3ex 0em 3ex 0em ;
613     border: 1px solid black;
614     border-radius: 0px ;
615     display: block ;
616 }
617
618 .framedleftbar {
619     margin: 3ex 0em 3ex 0em ;
620     border-left: 3pt solid black;
621     border-radius: 0px ;
622     padding: .3ex .2em .3ex 1em ;
623     display: block ;
624 }
625
626 .framedtitle {
627 margin: 0em ;
628 padding: 0em ;
629     font-size: 130%
630 }
631
632 .framedtitle p { padding: .3em }
633
```

```
634
635
636 dl {
637     margin: 1ex 2em 1ex 0em;
638     line-height: 1.3;
639 }
640
641 dl dt {
642     margin-top: 1ex;
643     font-weight: bold;
644 }
645
646 dl dd p { margin-top: 0em; }
647
648
649 nav.toc, nav.lof, nav.lot, nav.lol, nav.lothm {
650     font-family: "URW Classico", Optima, "Linux Biolinum O",
651         "DejaVu Sans", "Bitstream Vera Sans",
652         Geneva, Verdana, sans-serif ;
653     margin-bottom: 4ex ;
654 }
655
656 nav.toc p, nav.lof p, nav.lot p, nav.lol p, nav.lothm p {
657     line-height: 1.2 ;
658     margin-top:.5ex ;
659     margin-bottom:.5ex;
660     font-size: .9em ;
661 }
662
663
664
665 img, img.hyperimage, img.borderimage {
666     max-width: 600px;
667     border: 1px solid silver;
668     box-shadow: 3px 3px 3px #808080 ;
669     padding: .5% ;
670     margin: .5% ;
671     background: none ;
672 }
673
674 img.inlineimage{
675     padding: 0px ;
676     box-shadow: none ;
677     border: none ;
678     background: none ;
679     margin: 0px ;
680     display: inline-block ;
681     border-radius: 0px ;
682 }
683
```

```
684 img.logoimage{
685     max-width: 300px ;
686     box-shadow: 3px 3px 3px #808080 ;
687     border: 1px solid black ;
688     background:none ;
689     padding:0 ;
690     margin:.5ex ;
691     border-radius: 10px ;
692 }
693
694
695 .section {
696 /*
697     To have each section float relative to each other:
698 */
699 /*
700     display: block ;
701     float: left ;
702     position: relative ;
703     background: white ;
704     border: 1px solid silver ;
705     padding: .5em ;
706 */
707     margin: 0ex .5em 0ex .5em ;
708     padding: 0 ;
709 }
710
711
712 figure {
713     margin: 3ex auto 3ex auto ;
714     padding: 1ex 1em 1ex 1em ;
715     overflow-x: auto ;
716 }
717
718
719 /* To automatically center images in figures: */
720 /*
721 figure img.inlineimage {
722     margin: 0ex auto 0ex auto ;
723     display: block ;
724 }
725 */
726
727 /* To automatically center minipages in figures: */
728 /*
729 figure div.minipage, figure div.minipage div.minipage {
730     margin: 1ex auto 1ex auto ;
731     display: block ;
732 }
733 */
```



```
734
735 figure div.minipage p { font-size: 85% ; }
736
737 figure.subfigure, figure.subtable {
738     display: inline-block ; margin: 3ex 1em 3ex 1em ;
739 }
740
741 figcaption .minipage { margin:0 ; padding: 0 }
742
743 div.floatrow { text-align: center; }
744
745 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
746
747 div.floatfoot { font-size: .85em ;
748     border-top: 1px solid silver ; line-height: 1.2 ; }
749
750 figcaption , .lstlistingtitle {
751     font-size: .85em ;
752     text-align: center ;
753     font-weight: bold ;
754     margin-top: 1ex ;
755     margin-bottom: 1ex ;
756 }
757
758 figure.subfigure figcaption, figure.subtable figcaption {
759     border-bottom: none ; background: none ;
760 }
761
762 div.nonfloatcaption {
763     margin: 1ex auto 1ex auto ;
764     font-size: .85em ;
765     text-align: center ;
766     font-weight: bold ;
767 }
768
769 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
770 figure div.floatrow div.minipage figcaption {
771     border: none ;
772     background: none ;
773 }
774
775
776 table {
777     margin: 1ex auto 1ex auto ;
778     border-collapse: collapse ;
779     border-spacing: 0px ;
780     line-height: 1.3 ;
781 }
782
783 tr.hline {border-top: 1px solid silver ; margin-top: 0ex ;
```

```

784     margin-bottom: 0ex ; } /* for \hline */
785
786 tr.tbrule {border-top: 1px solid black ; margin-top: 0ex ;
787     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
788
789 td {padding: 1ex .5em 1ex .5em ;}
790
791 table td.tdl { text-align: left ; vertical-align: middle ; }
792 table td.tdc { text-align: center ; vertical-align: middle ; }
793 table td.tdr { text-align: right ; vertical-align: middle ; }
794 table td.tdp { text-align: left ; vertical-align: bottom ; }
795 table td.tdm { text-align: left ; vertical-align: middle ; }
796 table td.tdb { text-align: left ; vertical-align: top ; }
797 table td.tdP { text-align: center ; vertical-align: bottom ; }
798 table td.tdM { text-align: center ; vertical-align: middle ; }
799 table td.tdB { text-align: center ; vertical-align: top ; }
800 table td.tdlrule { text-align: left ; border-top: 1px solid silver ;
801     vertical-align: middle ; } /* for cmidrule */
802 table td.tdcrule { text-align: center ; border-top: 1px solid silver ;
803     vertical-align: middle ; }
804 table td.tdrrule { text-align: right ; border-top: 1px solid silver ;
805     vertical-align: middle ; }
806 table td.tdprule { text-align: left ; border-top: 1px solid silver ;
807     vertical-align: bottom ; }
808 table td.tdmrule { text-align: left ; border-top: 1px solid silver ;
809     vertical-align: middle ; }
810 table td.tdbrule { text-align: left ; border-top: 1px solid silver ;
811     vertical-align: top ; }
812 table td.tdPrule { text-align: center ; border-top: 1px solid silver ;
813     vertical-align: bottom ; }
814 table td.tdMrule { text-align: center ; border-top: 1px solid silver ;
815     vertical-align: middle ; }
816 table td.tdBrule { text-align: center ; border-top: 1px solid silver ;
817     vertical-align: top ; }
818
819 /* Margins of paragraphs inside table cells: */
820 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
821     padding-bottom: 1ex ; margin: 0ex ; }
822 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
823     padding-bottom: 1ex ; margin: 0ex ; }
824 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
825     padding-bottom: 1ex ; margin: 0ex ; }
826
827 td.tdp , td.tdprule , td.tdP , td.tdPrule
828     { padding: 0ex .5em 0ex .5em ; }
829 td.tdm , td.tdmrule , td.tdM , td.tdMrule
830     { padding: 0ex .5em 0ex .5em ; }
831 td.tdb , td.tdbrule , td.tdB , td.tdBrule
832     { padding: 0ex .5em 0ex .5em ; }
833

```

```
834
835 /* table notes: */
836 .tnotes {
837     margin: 0ex 5% 1ex 5% ;
838     padding: 0.5ex 1em 0.5ex 1em;
839     font-size:.85em;
840     text-align: left ;
841 }
842
843 .tnotes dl dt p {margin-bottom:0px;}
844
845 .tnoteitemheader {margin-right: 1em;}
846
847
848
849 /* center, flushleft, flushright environments */
850 div.center{text-align:center;}
851 div.center table {margin-left:auto;margin-right:auto;}
852 div.flushleft{text-align:left;}
853 div.flushleft table {margin-left:0em ; margin-right:auto;}
854 div.flushright{text-align:right;}
855 div.flushright table {margin-left:auto ; margin-right: 0em ;}
856
857
858
859
860 /* program listing callouts: */
861 span.callout {
862     font-family: "DejaVu Sans", "Bitstream Vera Sans",
863         Geneva, Verdana, sans-serif ;
864     border-radius: .5em;
865     background-color:black;
866     color:white;
867     padding:0px .25em 0px .25em;
868     margin: 0 ;
869     font-weight: bold;
870     font-size:.72em ;
871 }
872
873 div.programlisting pre.verbatim span.callout{
874 font-size: .85em ;
875 }
876
877
878
879
880
881 div.published
882 {
883     text-align: center ;
```

```
884     font-variant: normal ;
885     font-style: italic ;
886     font-size: 1em ;
887     margin: 3ex 0em 3ex 0em ;
888 }
889
890 div.subtitle
891 {
892     text-align: center ;
893     font-variant: normal ;
894     font-style: italic ;
895     font-size: 1.25em ;
896     margin: 3ex 0em 3ex 0em ;
897 }
898
899 div.subtitle p { margin: 1ex ; }
900
901 div.author
902 {
903     font-variant: normal ;
904     font-style: normal ;
905     font-size: 1em ;
906     margin: 3ex 0em 3ex 0em ;
907 }
908
909 div.author table {
910     margin: 3ex auto 0ex auto ;
911     background: none ;
912 }
913
914 div.author table tbody tr td { padding: .25ex ; }
915
916 span.affiliation {font-size: .85em ; font-variant: small-caps; }
917
918 div.titledate {
919     text-align: center ;
920     font-size: .85em ;
921     font-style: italic;
922     margin: 6ex 0em 6ex 0em ;
923 }
924
925
926 nav.topnavigation{
927     text-align: left ;
928     padding: 0.5ex 1em 0.5ex 1em ;
929 /*     margin: 2ex 0em 3ex 0em ; */
930     margin: 0 ;
931     border-bottom: 1px solid silver ;
932     border-top: 1px solid silver ;
933     clear:right ;
```

```
934 }
935
936 nav.botnavigation{
937     text-align: left ;
938     padding: 0.5ex 1em 0.5ex 1em ;
939 /*     margin: 3ex 0em 2ex 0em ; */
940     margin: 0 ;
941     border-top: 1px solid silver ;
942     border-bottom: 1px solid silver ;
943     clear:right ;
944 }
945
946
947 header{
948     line-height: 1.2 ;
949     font-size: 1em ;
950 /*     border-bottom: 2px solid silver ; */
951     margin: 0px ;
952     padding: 0ex 1em 0ex 1em ;
953     text-align:center ;
954 }
955
956 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
957
958
959 footer{
960     font-size: .85em ;
961     line-height: 1.2 ;
962     margin-top: 1ex ;
963     border-top: 2px solid silver ;
964     padding: 2ex 1em 2ex 1em ;
965     clear:right ;
966     text-align:left ;
967 }
968
969
970 a.linkhome { font-weight:bold ; font-size: 1em ;}
971
972
973 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
974
975 img.lateximage{
976     padding: 0px 0px 0px 0px ;
977     box-shadow: none ;
978     border: none ;
979     background: none ;
980     margin: 0px 0px -.15ex 0px ;
981     /* pdfcrop leaves a slight margin, adjust to baseline */
982     max-width: 100% ;
983     border-radius: 0ex ;
```

```
984     border: none ;
985 }
986
987
988
989 nav.sidetoc {
990     font-family: "DejaVu Serif", "Bitstream Vera Serif",
991         "Lucida Bright", Georgia, serif;
992     float:right ;
993     width: 20%;
994     border-left: 1px solid silver;
995     border-top: 1px solid silver;
996     border-bottom: 1px solid silver;
997 /*     border-top: 2px solid #808080 ; */
998     background: #FAF7F4 ;
999     padding: 2ex 0em 2ex 1em ;
1000     margin: 0ex 0em 2ex 1em ;
1001     font-size:.9em ;
1002     border-radius: 20px 0px 0px 20px ;
1003 }
1004
1005 div.sidetoccontents {
1006 /*     border-top: 1px solid silver ; */
1007     overflow-y: auto ;
1008     width: 100% ;
1009     text-align: left ;
1010 }
1011
1012 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
1013     text-indent: 0 ; }
1014 nav.sidetoc p a {color:black ; font-size: .7em ;}
1015 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
1016     border-bottom: 1px solid silver ; }
1017 nav.sidetoc a:hover {text-decoration: underline ; }
1018
1019
1020
1021 section.textbody { margin: 0ex 1em 0ex 1em ;}
1022
1023
1024 div.multicolsheading { -webkit-column-span: all;
1025     -moz-column-span: all; column-span: all; }
1026 div.multicols { -webkit-columns: 3 380px ;
1027     -moz-columns: 3 380px ; columns: 3 380px ; }
1028 div.multicols p {margin-top: 0ex}
1029
1030
1031
1032 /* Used to support algorithmicx: */
1033 span.floatright { float: right ; }
```

```
1034
1035
1036
1037
1038 /* Native LaTeX theorems: */
1039
1040 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
1041 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
1042
1043
1044 /* theorem, amsthm, and ntheorem packages */
1045
1046 span.theoremheader,
1047 span.theoremheaderplain,
1048 span.theoremheaderdefinition,
1049 span.theoremheaderbreak,
1050 span.theoremheadermarginbreak,
1051 span.theoremheaderchangebreak,
1052 span.theoremheaderchange,
1053 span.theoremheadermargin
1054 {
1055 font-style:normal ; font-weight: bold ; margin-right: 1em ;
1056 }
1057
1058 span.amsthmnameplain,
1059 span.amsthmnamedefinition,
1060 span.amsthmnumberplain,
1061 span.amsthmnumberdefinition
1062 {
1063 font-style:normal ; font-weight: bold ;
1064 }
1065
1066
1067 span.amsthmnameremark,
1068 span.amsthmnumberremark
1069 {font-style:italic ; font-weight: normal ; }
1070
1071
1072 span.amsthmnoteplain,
1073 span.amsthmnotedefinition
1074 {font-style:normal ;}
1075
1076
1077 span.theoremheaderremark,
1078 span.theoremheaderproof,
1079 span.amsthmproofname
1080 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
1081
1082 span.theoremheadersc
1083 {
```

```
1084 font-style:normal ;
1085 font-variant: small-caps ;
1086 font-weight: normal ;
1087 margin-right: 1em ;
1088 }
1089
1090 .theoremdemark {float:right}
1091
1092 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
1093 div.theorembodybreak, div.theorembodynonumberbreak,
1094 div.theorembodymarginbreak,
1095 div.theorembodychangebreak,
1096 div.theorembodychange,
1097 div.theorembodymargin
1098 {
1099 font-style:italic;
1100 margin-top: 3ex ; margin-bottom: 3ex ;
1101 }
1102
1103 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
1104 div.theorembodyplainupright, nonumberplainuprightsc,
1105 div.amsthmbodydefinition, div.amsthmbodyremark,
1106 div.amsthmproof
1107 {
1108 font-style: normal ;
1109 margin-top: 3ex ; margin-bottom: 3ex ;
1110 }
1111
1112 span.amsthmnoteremark {}
1113
1114
1115
1116 /*
1117 For CSS LaTeX and related logos:
1118 Based on:
1119 http://edward.oconnor.cx/2007/08/tex-poshlet
1120 http://nitens.org/taraborelli/texlogo
1121 */
1122
1123 .latexlogofont {
1124     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1125                 "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1126     font-variant: normal ;
1127 }
1128
1129 .latexlogo {
1130     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1131                 "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1132     letter-spacing: .03em ;
1133     font-size: 1.1em;
```



```
1134 }
1135
1136 .latexlogo sup {
1137   text-transform: uppercase;
1138   letter-spacing: .03em ;
1139   font-size: 0.85em;
1140   vertical-align: 0.15em;
1141   margin-left: -0.36em;
1142   margin-right: -0.15em;
1143 }
1144
1145 .latexlogo sub {
1146   text-transform: uppercase;
1147   vertical-align: -0.5ex;
1148   margin-left: -0.1667em;
1149   margin-right: -0.125em;
1150   font-size: 1em;
1151 }
1152
1153 .xetexlogo {
1154   font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1155               "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1156   letter-spacing: .03em ;
1157   font-size: 1.1em;
1158 }
1159
1160 /* A smaller gap between Xe and Tex v.s. LaTeX: */
1161 .xetexlogo sub {
1162   text-transform: uppercase;
1163   vertical-align: -0.5ex;
1164   margin-left: -0.0667em;
1165   margin-right: -0.2em;
1166   font-size: 1em;
1167   letter-spacing: .03em ;
1168 }
1169
1170 /* A large gap between Xe and LaTeX v.s. TeX: */
1171 .xelatexlogo sub {
1172   text-transform: uppercase;
1173   vertical-align: -0.5ex;
1174   margin-left: -0.0667em;
1175   margin-right: -.05em;
1176   font-size: 1em;
1177   letter-spacing: .03em ;
1178 }
1179
1180 .amslogo {
1181   font-family: "TeXGyreChorus", "URW Chancery L",
1182               "Apple Chancery", "ITC Zapf Chancery", "Monotype Corsiva",
1183               "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
```

```
1184         "Hoefler Text", Times, "Times New Roman", serif;
1185     font-style: italic;
1186 }
1187
1188 .lyxlogo {
1189     font-family: "URW Classico", Optima, "Linux Biolinum O",
1190         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
1191     Verdana, sans-serif ;
1192 }
1193
1194
1195
1196
1197 /* Only display top and bottom navigation if a small screen: */
1198 /* Hide the sidetoc if a small screen: */
1199 nav.topnavigation { display:none; }
1200 nav.botnavigation { display:none; }
1201
1202 @media screen and (max-width: 45em) {
1203     /*     nav.sidetoc {display:none;} */
1204     nav.sidetoc {
1205         float: none ;
1206         width: 100% ;
1207         margin: 5ex 0px 5ex 0px ;
1208         padding: 0 ;
1209         border-radius: 0 ;
1210         border-bottom: 1px solid black ;
1211         border-top: 1px solid black ;
1212         box-shadow: none ;
1213     }
1214     /*     nav.topnavigation { display:block } */
1215     nav.botnavigation { display:block }
1216     .marginpar {
1217         max-width: 100%;
1218         float: none;
1219         display:block ;
1220         margin: 1ex 1em 1ex 1em ;
1221     }
1222 }
1223
1224 @media print {
1225     body {
1226         font-family: "Linux Libertine O",
1227             "DejaVu Serif", "Bitstream Vera Serif",
1228             "Liberation Serif", "Nimbus Roman No 9 L",
1229             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1230     }
1231     nav.sidetoc { display:none; }
1232     nav.topnavigation { display: none; }
1233     nav.botnavigation { display: none; }
```

```

1234 }
1235
1236 @media handheld {
1237     nav.sidetoc { display:none; }
1238     nav.topnavigation { display:block }
1239     nav.botnavigation { display:block }
1240 }
1241
1242 @media projection {
1243     nav.sidetoc { display:none; }
1244     nav.topnavigation { display:block }
1245     nav.botnavigation { display:block }
1246 }
1247 \end{VerbatimOut}
1248 % \end{Verbatim}% for syntax highlighting

```

## 69.4 lwarp\_sagebrush.css

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

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

```

1249 \begin{VerbatimOut}{lwarp_sagebrush.css}
1250 @import url("lwarp.css") ;
1251
1252
1253 A:link {color:#105030 ; text-decoration: none ; }
1254 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
1255 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
1256 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
1257
1258
1259
1260 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1261 {
1262     font-family: "URW Classico", Optima, "Linux Biolinum O",
1263         "Linux Libertine O", "Liberation Serif",
1264         "Nimbus Roman No 9 L", "FreeSerif",
1265         "Hoefler Text", Times, "Times New Roman", serif;
1266     font-variant: small-caps ;
1267     font-weight: normal ;
1268     color: #304070 ;
1269     text-shadow: 2px 2px 3px #808080;
1270 }
1271
1272 h1 { /* title of the entire website, used on each page */

```

```
1273     font-variant: small-caps ;
1274     color: #304070 ;
1275     text-shadow: 2px 2px 3px #808080;
1276     background-color: #F7F7F0 ;
1277     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
1278 }
1279
1280 h1 {
1281     border-bottom: 1px solid #304070;
1282     border-top: 2px solid #304070;
1283 }
1284
1285 h2 {
1286     border-bottom: 1px solid #304070;
1287     border-top: 2px solid #304070;
1288     background-color: #F7F7F0 ;
1289     background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
1290 }
1291
1292
1293
1294 div.abstract {
1295     background: #f5f5eb ;
1296     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1297
1298     border: 1px solid silver;
1299     border-radius: 1em ;
1300 }
1301
1302 div.abstract dl {line-height:1.5;}
1303 div.abstract dt {color:#304070;}
1304
1305 div.abstracttitle{
1306     font-family: "URW Classico", Optima, "Linux Biolinum O",
1307         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1308         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1309     font-weight:bold;
1310     font-variant: small-caps ;
1311     font-size:1.5em;
1312     border-bottom: 1px solid silver ;
1313     color: #304070 ;
1314     text-align: center ;
1315     text-shadow: 1px 1px 2px #808080;
1316 }
1317
1318 span.abstracrunintitle{
1319     font-family: "URW Classico", Optima, "Linux Biolinum O",
1320         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1321         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1322     font-weight:bold;
```

```
1323 }
1324
1325
1326 div.epigraph {
1327     background: #f5f5eb ;
1328     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1329
1330     border: 1px solid silver ;
1331     border-radius: 1ex ;
1332     box-shadow: 3px 3px 3px #808080 ;
1333 }
1334
1335
1336 .example {
1337     background-color: #f5f5eb ;
1338     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1339
1340 }
1341
1342 div.exampletitle{
1343     font-family: "URW Classico", Optima, "Linux Biolinum O",
1344         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1345         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1346     font-weight:bold;
1347     font-variant: small-caps ;
1348     border-bottom: 1px solid silver ;
1349     color: #304070 ;
1350     text-align: center ;
1351     text-shadow: 1px 1px 2px #808080;
1352 }
1353
1354
1355 .sidebar {
1356     background-color: #f5f5eb ;
1357     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1358
1359 }
1360
1361 div.sidebar{
1362     font-family: "URW Classico", Optima, "Linux Biolinum O",
1363         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1364         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1365     font-weight:bold;
1366     font-variant: small-caps ;
1367     border-bottom: 1px solid silver ;
1368     color: #304070 ;
1369     text-align: center ;
1370     text-shadow: 1px 1px 2px #808080;
1371 }
1372
```

```
1373
1374 .fancyvrblabel {
1375     font-family: "URW Classico", Optima, "Linux Biolinum O",
1376         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1377         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1378     font-weight:bold;
1379     font-variant: small-caps ;
1380 font-size: 1.5em ;
1381     color: #304070 ;
1382     text-align: center ;
1383     text-shadow: 1px 1px 2px #808080;
1384 }
1385
1386
1387
1388 .minipage {
1389     background-color: #eeeeee7 ;
1390     border: 1px solid silver ;
1391 border-radius: 1ex ;
1392 }
1393
1394 .framed .minipage , .framedleftbar .minipage {
1395 border: none ;
1396 background: none ;
1397 padding: 0ex ;
1398 margin: 0ex ;
1399 }
1400
1401 figure.figure .minipage, figcaption .minipage { border: none; }
1402
1403 div.marginblock div.minipage { border: none; }
1404
1405 figure , div.marginblock {
1406     background-color: #eeeeee7 ;
1407     border: 1px solid silver ;
1408     border-radius: 1ex ;
1409     box-shadow: 3px 3px 3px #808080 ;
1410 }
1411
1412 figure figure {
1413     border: 1px solid silver ;
1414     margin: 0em ;
1415 box-shadow: none ;
1416 }
1417
1418 /*
1419 figcaption {
1420     border-top: 1px solid silver ;
1421     border-bottom: 1px solid silver ;
1422     background-color: #e8e8e8 ;
```

```
1423 }
1424 */
1425
1426
1427 div.table {
1428     box-shadow: 3px 3px 3px #808080 ;
1429 }
1430
1431 /*
1432 .tnotes {
1433     background: #e8e8e8;
1434     border: 1px solid silver;
1435 }
1436 */
1437
1438
1439 nav.topnavigation{
1440     background-color: #b0b8b0 ;
1441     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
1442 }
1443
1444 nav.botnavigation{
1445     background-color: #b0b8b0 ;
1446     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
1447 }
1448
1449
1450
1451 header{
1452     background-color: #F7F7F0 ;
1453     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
1454 }
1455
1456 footer{
1457     background-color: #F7F7F0 ;
1458     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
1459 }
1460
1461
1462
1463 nav.sidetoc {
1464     background-color: #F7F7F0 ;
1465     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
1466     box-shadow: 3px 3px 3px #808080 ;
1467     border-radius: 0px 0px 0px 20px ;
1468 }
1469
1470 div.sidetocitle {color: #304070 ; }
1471 nav.sidetoc a:hover {color:#006000 ; text-decoration: none ; text-shadow:0px 0px 2px #a0a0a0;}
1472
```

```

1473
1474 @media screen and (max-width: 45em) {
1475     nav.sidetoc { border-radius: 0 ; }
1476 }
1477
1478
1479 \end{VerbatimOut}
1480 % \end{Verbatim}% for syntax highlighting

```

## 69.5 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.

```

1481 \begin{VerbatimOut}{lwarp_formal.css}
1482 @import url("lwarp.css") ;
1483
1484
1485
1486 A:link {color:#802020 ; text-decoration:none; }
1487 A:visited {color:#802020 ; text-shadow:none ;}
1488 A:hover {color:#400000 ; text-shadow:none ;}
1489 A:active {color:#C00000 ; text-shadow:none ;}
1490
1491
1492 body {
1493     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1494         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1495         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1496         "Times New Roman", serif;
1497     background: #fffcf5;
1498 }
1499
1500 span.textrm {
1501     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1502         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1503         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1504         "Times New Roman", serif;
1505 }
1506
1507 span.textsf {
1508     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1509         Geneva, Verdana, sans-serif ;
1510 }
1511

```



```
1512
1513
1514 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1515 {
1516     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1517         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1518         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1519         "Times New Roman", serif;
1520     color: #800000 ;
1521     text-shadow: none ;
1522 }
1523
1524 h1, h2 {
1525     background-color: #fffcf5 ;
1526     background-image: none ;
1527     border-bottom: 1px solid #808080;
1528     border-top: 2px solid #808080;
1529 }
1530
1531 div.abstracttitle {
1532     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1533         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1534         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1535         "Times New Roman", serif;
1536     color: black ;
1537     text-shadow: none ;
1538 }
1539
1540 span.abstracrunintitle {
1541     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1542         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1543         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1544         "Times New Roman", serif;
1545     color: black ;
1546     text-shadow: none ;
1547 }
1548
1549 div.abstract { font-size: 100% }
1550
1551 .sidebar {
1552     background: #fffcf5;
1553     background-image: none ;
1554     margin: 2em 5% 2em 5%;
1555     padding: 0.5em 1em;
1556     border: none ;
1557     border-top : 1px solid silver;
1558     border-bottom : 1px solid silver;
1559     font-size: 90% ;
1560 }
1561
```

```
1562 div.sidebartitle{
1563     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1564         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1565         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1566         "Times New Roman", serif;
1567     color: #800000 ;
1568     text-shadow: none ;
1569     border: none ;
1570 }
1571
1572 .example {
1573     background: #fffcf5;
1574     background-image: none ;
1575     margin: 2em 5% 2em 5%;
1576     padding: 0.5em 1em;
1577     border: none ;
1578     border-top : 1px solid silver;
1579     border-bottom : 1px solid silver;
1580 }
1581
1582 div.exampletitle{
1583     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1584         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1585         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1586         "Times New Roman", serif;
1587     color: #800000 ;
1588     text-shadow: none ;
1589     border: none ;
1590 }
1591
1592 div.fancyvrblabel{
1593     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1594         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1595         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1596         "Times New Roman", serif;
1597     color: #800000 ;
1598     text-shadow: none ;
1599     border: none ;
1600 }
1601
1602
1603
1604 .verse {
1605     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1606         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1607         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1608         "Times New Roman", serif;
1609 }
1610
1611
```

```
1612 figure {
1613     margin: 3ex 5% 3ex 5% ;
1614     padding: 1ex 1em 1ex 1em ;
1615     background-color: #fffcf5 ;
1616     overflow-x: auto ;
1617     border: none ;
1618 /*     border-top: 1px solid silver; */
1619 /*     border-bottom: 1px solid silver; */
1620 }
1621
1622
1623 figcaption , .lstlisting {
1624     border: none ;
1625 /*     border-top: 1px solid silver ; */
1626 /*     border-bottom: 1px solid silver ; */
1627     background-color: #fffcf5 ;
1628 }
1629
1630 .tnotes {
1631     background: #fffcf5 ;
1632 }
1633
1634 .theorem {
1635     background: none ;
1636 }
1637
1638 .minipage {
1639     background-color: #fffcf5 ;
1640     border: none ;
1641 }
1642
1643 div.floatrow figure { border: none ; }
1644
1645 figure figure { border: none ; }
1646
1647
1648 nav.toc, nav.lof, nav.lot, nav.lol {
1649     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1650         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1651         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1652         "Times New Roman", serif;
1653 }
1654
1655 nav.sidetoc {
1656     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1657         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1658         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1659         "Times New Roman", serif;
1660     background-image: linear-gradient(to bottom, #fffcf5, #C0C0C0);
1661     border-radius: 0px 0px 0px 20px ;
```

```

1662 }
1663
1664 div.sidetocitle{
1665     color: #800000 ;
1666 }
1667
1668 header{
1669     background-color: #e0e0e0 ;
1670     background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
1671     text-align:center ;
1672 }
1673
1674 footer{
1675     background-color: #e0e0e0 ;
1676     background-image: linear-gradient(to bottom, #fffcf5, #b0b0b0);
1677     padding: 2ex 1em 2ex 1em ;
1678     clear:right ;
1679     text-align:left ;
1680 }
1681
1682 nav.botnavigation {
1683     background: #dedcd5 ;
1684     border-top: 1px solid black ;
1685 }
1686 \end{VerbatimOut}
1687 % \end{Verbatim}% for syntax highlighting

```

## 69.6 sample\_project.css

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

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

```

1688 \begin{VerbatimOut}{sample_project.css}
1689 /* ( --- Start of project.css --- ) */
1690 /* A sample project-specific CSS file for lwarp --- ) */
1691
1692 /* Load default lwarp settings: */
1693 @import url("lwarp.css") ;
1694 /* or lwarp_formal.css, lwarp_sagebrush.css */
1695
1696 /* Project-specific CSS setting follow here. */
1697 /* . . . */
1698
1699 /* ( --- End of project.css --- ) */
1700 \end{VerbatimOut}
1701 % \end{Verbatim}% for syntax highlighting

```

## 69.7 lwarp\_html.xdy

File `lwarp_html.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.

```

1702 \begin{VerbatimOut}{lwarp_html.xdy}
1703 (require "tex/inputenc/latin.xdy")
1704 (merge-rule "\\PS *" "Postscript")
1705 (require "texindy.xdy")
1706 (require "page-ranges.xdy")
1707 (require "book-order.xdy")
1708 (markup-locref :open "\hyperindexref{" :close "{")
1709 \end{VerbatimOut}
1710 % \end{Verbatim}% for syntax highlighting

```

## 69.8 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.

```

1711 \begin{VerbatimOut}{lwarp_mathjax.txt}
1712 <!-- https://groups.google.com/forum/#!topic/
1713                               mathjax-users/jUteWUcE2bY -->
1714 <script type="text/x-mathjax-config">
1715 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
1716     var seteqsectionDefault = {name: "", num: 0};
1717     var seteqsections = {}, seteqsection = seteqsectionDefault;
1718     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
1719     var AMS = MathJax.Extension["TeX/AMSmath"];
1720     TEX.Definitions.Add({
1721       macros: {
1722         seteqsection: "mySection",
1723         seteqnumber: "mySetEqNumber"
1724       }
1725     });
1726
1727     PARSE.Augment({
1728       mySection: function (name) {
1729         seteqsection.num = AMS.number;
1730         var n = this.GetArgument(name);
1731         if (n === "") {
1732           seteqsection = seteqsectionDefault;
1733         } else {

```

```

1734         if (!seteqsections["_"+n])
1735             seteqsections["_"+n] = {name:n, num:0};
1736         seteqsection = seteqsections["_"+n];
1737     }
1738     AMS.number = seteqsection.num;
1739 },
1740 mySetEqNumber: function (name) {
1741     var n = this.GetArgument(name);
1742     if (!n || !n.match(/^ *[0-9]+ *$/))
1743         n = ""; else n = parseInt(n)-1;
1744     <!-- $ syntax highlighting -->
1745     if (n === "" || n < 1)
1746         TEX.Error
1747             ("Argument to "+name+" should be a positive integer");
1748     AMS.number = n;
1749 }
1750 });
1751 MathJax.Hub.Config({
1752   TeX: {
1753     equationNumbers: {
1754       formatTag: function (n)
1755         {return "("+(seteqsection.name+"."+n).replace(/\./,"")+")"},
1756       formatID: function (n) {
1757         n = (seteqsection.name+'.'+n).replace
1758           (/[:'"<>&]/g,"").replace(/\./,"");
1759         return 'mjx-eqn-' + n;
1760       }
1761     }
1762   }
1763 });
1764 });
1765 </script>
1766
1767 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
1768 <script type="text/x-mathjax-config">
1769   MathJax.Ajax.config.path["Contrib"] =
1770     "https://cdn.mathjax.org/mathjax/contrib";
1771 </script>
1772
1773 <!-- https://github.com/mathjax/MathJax-third-party-extensions/
1774                                     tree/master/siunitx -->
1775 <script type="text/x-mathjax-config">
1776   MathJax.Hub.Config({
1777     extensions: ["tex2jax.js", "[Contrib]/siunitx/siunitx.js"],
1778     jax: ["input/TeX", "output/HTML-CSS"],
1779     tex2jax: {inlineMath: [["$","$"],["\\(","\\)"]]},
1780     TeX: {extensions: ["AMSmath.js", "AMSsymbols.js", "sinuitx.js"]}
1781   });
1782 </script>
1783

```

```

1784 <script type="text/x-mathjax-config">
1785 MathJax.Hub.Config({
1786     TeX: {
1787         equationNumbers: {
1788             autoNumber: "AMS"
1789         }
1790     }
1791 });
1792 </script>
1793
1794 <script
1795     src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
1796 </script>
1797 \end{VerbatimOut}
1798 % \end{Verbatim}% for syntax highlighting

```

## 69.9 Lwarpmk option

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

```

1799 \begin{LWR@createlwarpmk}

```

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

```

1800 \begin{VerbatimOut}{lwarpmk.lua}
1801 #!/usr/bin/env texlua
1802
1803 -- Copyright 2016-2017 Brian Dunn
1804
1805 -- Print the usage of the lwarpmk command:
1806
1807 printversion = "v0.27"
1808
1809 function printhelp ()
1810 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
1811 end
1812
1813 function printusage ()
1814 print ( [[
1815
1816 lwarpmk print [project]: Compile a print version.
1817 lwarpmk printindex [project]: Process the index for the print version.
1818 lwarpmk html [project]: Compile an HTML version.
1819 lwarpmk htmlindex [project]: Process the index for the html version.
1820 lwarpmk again [project]: Touch the source code to trigger recompiles.
1821 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
1822 lwarpmk pdftohtml [project]:

```

```

1823     For use with latexmk or a Makefile:
1824     Convert project_html.pdf to project_html.html and
1825     individual HTML files.
1826 lwarpmk clean [project]: Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log
1827 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
1828 lwarpmk -h: Print this help message.
1829 lwarpmk --help: Print this help message.
1830
1831 ]] )
1832 printconf ()
1833 end
1834
1835 -- Print the format of the configuration file lwarpmk.conf:
1836
1837 function printconf ()
1838 print ( [[
1839 An example lwarpmk.conf or <project>.lwarpmkconf project file:
1840 --
1841 opsystem = "Unix"      (or "Windows")
1842 latexname = "pdflatex" (or "lualatex", or "xelatex")
1843 sourcename = "projectname" (the source-code filename w/o .tex)
1844 homehtmlfilename = "index" (or perhaps the project name)
1845 htmlfilename = ""      (or "projectname" - filename prefix)
1846 uselatexmk = "false"   (or "true" to use latexmk to build PDFs)
1847 --
1848 Filenames must contain only letters, numbers, underscore, or dash.
1849 Values must be in "quotes".
1850
1851 ]] ) ;
1852 end
1853
1854
1855 -- Split one large sourcefile into a number of files,
1856 -- starting with destfile.
1857 -- The file is split at each occurrence of <!--|Start file|newfilename|*
1858
1859 function splitfile (destfile,sourcefile)
1860 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
1861 io.input(sourcefile)
1862 io.output(destfile)
1863 for line in io.lines() do
1864 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
1865 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
1866 io.output(newfilename) ;
1867 else -- not a splitpoint
1868 io.write (line .. "\n") ;
1869 end
1870 end -- do
1871 end -- function
1872

```



```
1873 -- Incorrect value, so print an error and exit.
1874
1875 function cvalueerror ( line, linenum , cvalue )
1876     print ( linenum .. " : " .. line ) ;
1877     print ("lwarpmk: incorrect variable value \"" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
1878     printconf ( ) ;
1879     os.exit(1) ;
1880 end
1881
1882 -- Load settings from the project's "lwarpmk.conf" file:
1883
1884 function loadconf ( )
1885 -- Default configuration filename:
1886 local conffile = "lwarpmk.conf"
1887 -- Optional configuration filename:
1888 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
1889 -- Verify the file exists:
1890 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
1891 print("lwarpmk: " .. conffile .. " does not exist.")
1892 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
1893 printhelp ( ) ;
1894 os.exit(1) -- exit the entire lwarpmk script
1895 else -- file exists
1896 -- Read the file:
1897 print ("lwarpmk: Reading " .. conffile .. ".")
1898 io.input(conffile) ;
1899 -- Scan each line:
1900 local linenum = 0
1901 for line in io.lines() do -- scan lines
1902 linenum = linenum + 1
1903 i,j,cvarname,cvalue = string.find (line,"([%w-_]*)%s*=%s*\"([%w-_]*)\"") ;
1904 -- Error if incorrect enclosing characters:
1905 if ( i == nil ) then
1906 print ( linenum .. " : " .. line ) ;
1907 print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
1908 printconf ( ) ;
1909 os.exit(1) ;
1910 end
1911 if ( cvarname == "opsystem" ) then
1912     -- Verify choice of opsystem:
1913     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
1914         opsystem = cvalue
1915     else
1916         cvalueerror ( line, linenum , cvalue )
1917     end
1918 elseif ( cvarname == "latexname" ) then
1919     -- Verify choice of LaTeX compiler:
1920     if (
1921         (cvalue == "pdflatex") or
1922         (cvalue == "xelatex") or
```

```

1923         (cvalue == "lualatex")
1924     ) then
1925         latexname = cvalue
1926     else
1927         cvalueerror ( line, linenum , cvalue )
1928     end
1929 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
1930 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
1931 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
1932 elseif ( cvarname == "uselatexmk" ) then uselatexmk = cvalue
1933 else
1934 print ( linenum .. " : " .. line ) ;
1935 print ("lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. conffile .. ".\n" ) ;
1936 printconf ( ) ;
1937 os.exit(1) ;
1938 end
1939 end -- do scan lines
1940 end -- file exists
1941 -- Select some operating-system commands:
1942 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
1943 rmname = "rm"
1944 mvname = "mv"
1945 touchnamepre = "touch"
1946 touchnamepost = ""
1947 dirslash = "/"
1948 opquote= "\""
1949 elseif opsystem=="Windows" then -- For Windows
1950 rmname = "DEL"
1951 mvname = "MOVE"
1952 touchnamepre = "COPY /b"
1953 touchnamepost = "+,,"
1954 dirslash = "\\"
1955 opquote= "\""
1956 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
1957 end --- for Windows
1958 end -- loadconf
1959
1960
1961 function refreshdate ( )
1962 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
1963 end
1964
1965
1966 -- Scan the LaTeX log file for the phrase "Rerun to get",
1967 -- indicating that the file should be compiled again.
1968 -- Return true if found.
1969
1970 function reruntoget (filesourse)
1971 io.input(filesourse)
1972 for line in io.lines() do

```

```
1973 if ( string.find(line,"Rerun to get") ~= nil ) then return true end
1974 end
1975 return false
1976 end
1977
1978 -- Compile one time, return true if should compile again.
1979 -- fsuffix is "" for print, "_html" for HTML output.
1980
1981 function onetime (fsuffix)
1982 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
1983 err = os.execute(
1984 --      "echo " ..
1985      latexname .. " " .. sourcename..fsuffix )
1986 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
1987 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
1988 end
1989
1990 -- Compile up to five times.
1991 -- fsuffix is "" for print, "_html" for HTML output
1992
1993 function manytimes (fsuffix)
1994 if onetime(fsuffix) == true then
1995 if onetime(fsuffix) == true then
1996 if onetime(fsuffix) == true then
1997 if onetime(fsuffix) == true then
1998 if onetime(fsuffix) == true then
1999 end end end end end
2000 end
2001
2002 -- Exit if the given file does not exist.
2003
2004 function verifyfileexists (filename)
2005 if (lfs.attributes ( filename , "modification" ) == nil ) then
2006 print ( "lwarpmk: " .. filename .. " not found." ) ;
2007 os.exit (1) ;
2008 end
2009 end
2010
2011 -- Convert <project>_html.pdf into HTML files:
2012
2013 function pdftohtml ()
2014     -- Convert to text:
2015     print ("lwarpmk: Converting " .. sourcename .. "_html.pdf to " .. sourcename .. "_html.html"
2016     os.execute("pdftotext -enc UTF-8 -nopgbrk -layout " .. sourcename .. "_html.pdf " .. sou
2017     -- Split the result into individual HTML files:
2018     splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
2019 end
2020
2021 -- Remove auxiliary files:
2022
```

```

2023 function removeaux ()
2024     os.execute ( rmname .. " " ..
2025         sourcename .. ".aux " .. sourcename .. "_html.aux " ..
2026         sourcename .. ".toc " .. sourcename .. "_html.toc " ..
2027         sourcename .. ".lof " .. sourcename .. "_html.lof " ..
2028         sourcename .. ".lot " .. sourcename .. "_html.lot " ..
2029         sourcename .. ".idx " .. sourcename .. "_html.idx " ..
2030         sourcename .. ".ind " .. sourcename .. "_html.ind " ..
2031         sourcename .. ".log " .. sourcename .. "_html.log "
2032     )
2033 end
2034
2035
2036
2037 -- Create lateximages based on lateximages.txt:
2038 function createlateximages ()
2039 print ("lwarpmk: Creating lateximages.")
2040 io.input("lateximages.txt")
2041 -- Create the lateximages directory, ignore error if already exists
2042 err = os.execute("mkdir lateximages")
2043 -- Scan lateximages.txt
2044 for line in io.lines() do
2045 -- lwimgpage is the page number in the PDF which has the image
2046 -- lwimgnum is the sequential lateximage number to assign for the image
2047 i,j,lwimgpage,lwimgnum = string.find (line,"|(.*)|(.*)|")
2048 -- For each entry:
2049 if ( i~=nil ) then
2050 -- Separate out the image into its own single-page pdf:
2051 err = os.execute(
2052 "pdfseparate -f " .. lwimgpage .. " -l " ..
2053 lwimgpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
2054 -- Crop the image:
2055 err = os.execute(
2056 "pdfcrop -- hires lateximagetemp-" .. lwimgpage .. ".pdf lateximage-" .. lwimgnum .. ".pdf")
2057 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2058 -- Convert the image to svg:
2059 err = os.execute(
2060 "pdftocairo -svg lateximage-" .. lwimgnum .. ".pdf lateximage-" .. lwimgnum .. ".svg")
2061 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2062 -- Move the result into lateximages/:
2063 err = os.execute(
2064 mvname .. " lateximage-" .. lwimgnum .. ".svg lateximages" .. dirslash )
2065 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2066 -- Remove the temporary files:
2067 err = os.execute(
2068 rmname .. " lateximage-" .. lwimgnum .. ".pdf lateximagetemp-" .. lwimgpage .. ".pdf")
2069 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2070 end
2071 end -- do
2072 end -- function

```

```
2073
2074
2075
2076
2077 -- lwarpmk --version :
2078
2079 if (arg[1] == "--version") then
2080   print ( "lwarpmk: " .. printversion )
2081
2082 else -- not -- version
2083
2084 -- print intro:
2085
2086 print ("lwarpmk: " .. printversion .. "   Automated make for the LaTeX lwarp package.")
2087
2088 -- lwarpmk print:
2089
2090 if arg[1] == "print" then
2091   loadconf ()
2092   if ( uselatexmk == "true" ) then
2093     os.execute ( "latexmk -pdf -dvi- -ps- -pdflatex=\"\" .. latexname .." %0 %S\" " .. sourcenam
2094     print ("lwarpmk: Done.")
2095   else -- not latexmk
2096     verifyfileexists (sourcename .. ".tex") ;
2097     -- See if up to date:
2098     if (
2099       ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
2100       (
2101         lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2102         lfs.attributes ( sourcename .. ".pdf" , "modification" )
2103       )
2104     ) then
2105       -- Recompile if not yet up to date:
2106       manytimes("")
2107       print ("lwarpmk: Done.") ;
2108     else
2109       print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
2110     end
2111   end -- not latexmk
2112
2113 -- lwarp printindex:
2114 -- Compile the index then touch the source
2115 -- to trigger a recompile of the document:
2116
2117 elseif arg[1] == "printindex" then
2118   loadconf ()
2119   print ("lwarpmk: Processing the index.")
2120   os.execute("texindy -M lwarp_html.xdy " .. sourcename .. ".idx")
2121   print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2122   refreshdate ()
```

```
2123 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2124 print ("lwarpmk: Done.")
2125
2126 -- lwarpmk html:
2127
2128 elseif arg[1] == "html" then
2129 loadconf ()
2130 if ( uselatexmk == "true" ) then
2131     -- The recorder option is required to detect changes in <project>.tex
2132     -- while we are loading <project>_html.tex.
2133     err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
2134         .. "-e "
2135         .. opquote .. "$makeindex = q/texindy -M lwarp_html.xdy/" .. opquote
2136         .. " -pdflatex=\"\" .. latexname .. " %0 %S\" "
2137         .. sourcename .. "_html.tex" ) ;
2138     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2139     pdftohtml ()
2140     print ("lwarpmk: Done.")
2141 else -- not latexmk
2142     verifyfileexists ( sourcename .. ".tex" ) ;
2143     -- See if exists and is up to date:
2144     if (
2145         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
2146         (
2147             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2148             lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
2149         )
2150     ) then
2151         -- Recompile if not yet up to date:
2152         manytimes("_html")
2153         pdftohtml ()
2154         print ("lwarpmk: Done.")
2155     else
2156         print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
2157     end
2158 end -- not latexmk
2159
2160 elseif arg[1] == "pdftohtml" then
2161     loadconf ()
2162     pdftohtml ()
2163
2164 -- lwarpmk htmlindex:
2165 -- Compile the index then touch the source
2166 -- to trigger a recompile of the document:
2167
2168 elseif arg[1] == "htmlindex" then
2169 loadconf ()
2170 print ("lwarpmk: Processing the index.")
2171 os.execute("texindy -M lwarp_html.xdy " .. sourcename .. "_html.idx")
2172 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
```

```
2173 refreshdate ()
2174 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2175 print ("lwarpmk: Done.")
2176
2177 -- lwarpmk limages:
2178 -- Scan the lateximages.txt file to create lateximages,
2179 -- then touch the source to trigger a recompile.
2180
2181 elseif arg[1] == "limages" then
2182 loadconf ()
2183 print ("lwarpmk: Processing images.")
2184 createlateximages ()
2185 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2186 refreshdate ()
2187 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2188 print ("lwarpmk: Done.")
2189
2190 -- lwarpmk again:
2191 -- Touch the source to trigger a recompile.
2192
2193 elseif arg[1] == "again" then
2194 loadconf ()
2195 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2196 refreshdate ()
2197 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2198 print ("lwarpmk: Done.")
2199
2200 -- lwarpmk clean:
2201 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log
2202
2203 elseif arg[1] == "clean" then
2204 loadconf ()
2205 removeaux ()
2206 print ("lwarpmk: Done.")
2207
2208 -- lwarpmk cleanall
2209 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log
2210 -- and also project.pdf, *.html
2211
2212 elseif arg[1] == "cleanall" then
2213 loadconf ()
2214 removeaux ()
2215 os.execute ( rmname .. " " ..
2216     sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
2217     "*.html"
2218 )
2219 print ("lwarpmk: Done.")
2220
2221 -- lwarpmk with no argument :
2222
```

```
2223 elseif (arg[1] == nil) then
2224 printhelp ()
2225
2226 -- lwarpmk -h or lwarpmk --help :
2227
2228 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
2229 printusage ()
2230
2231 else
2232 print ("lwarpmk: Unknown command \""..arg[1].."\".\n")
2233 printhelp ()
2234 end
2235
2236 end -- not --version
2237 \end{VerbatimOut}
2238 % \end{Verbatim}% for syntax highlighting
2239 \end{LWR@createlwarpmk}
```



## Package 3

# lwarp-abstract.sty

## 70 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}{\BlockClass{abstract}}
4 \AfterEndEnvironment{abstract}{\endBlockClass}
5 }
6
7 \renewcommand{\@bsrunintitle}{%
8 \hspace*{\abstitlekip}%
9 {\abstractnamefont%
10 \InlineClass{abstractrunintitle}{\abstractname}%
11 \@bslabeldelim}%
12 }
13
14 \if@titlepage
15 \renewenvironment{abstract}{%
16 % \titlepage
17 \null\vfil
18 \@beginparpenalty\@lowpenalty
19 \if@bsrunin
20 \else
21 \if@bsstyle
22 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
23 \else
24 \ifnumber@bs
25 \num@bs
26 \else
27 \begin{\absnamepos}%
28 \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}

```

---

```

29         \@endparpenalty\@M
30         \end\absnamepos%
31 %%         \vspace{\abstitleskip}%
32         \fi
33     \fi
34     \vspace{\abstitleskip}%
35 \fi
36 \put@bsintoc%
37 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
38 {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
39 }
40 \else
41 \renewenvironment{abstract}{%
42 \if@bsrunin
43 \else
44 \if@bsstyle
45 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
46 \else
47 \ifnumber@bs
48 \num@bs
49 \else
50 \begin{\absnamepos}%
51 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
52 \end\absnamepos%
53 %%         \vspace{\abstitleskip}%
54         \fi
55     \fi
56     \vspace{\abstitleskip}%
57 \fi
58 \put@bsintoc%
59 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
60 {\par\end{@bstr@ctlist}}
61 \fi
62

```

## Package 4

# lwarp-afterpage.sty

## 71 Afterpage

Pkg `afterpage` Not used.

for **HTML output**: Discard all options for `lwarp-afterpage`:

```
1 \LWR@ProvidesPackageDrop{afterpage}
```

```
2 \newcommand{\afterpage}[1]{#1}
```

## Package 5


# lwarp-algorithmicx.sty

## 72 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 [153.1](#).

for HTML output: 2 \begin{warpHTML}

```

3 \AtBeginEnvironment{algorithmic}{%
4 %
5 \let\origALG@doentity\ALG@doentity%
6 %
7 \renewcommand*{\ALG@doentity}{%
8 \origALG@doentity%
9 \uselengthunit{PT}%
10 \LWR@htmltagc{%
11 span style="width:\rndprintlength{\ALG@thistlm}; display:inline-block;"%
12 }%
13 \LWR@htmltagc{/span}%
14 }%
15 %
16 \let\origComment\Comment%
17 %
18 \renewcommand{\Comment}[1]{\InlineClass{floatright}{\origComment{#1}}}%
19 }

20 \end{warpHTML}
```

## Package 6

# lwarp-alltt.sty

## 73 Alltt

Pkg alltt alltt is patched for use by lwarp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{alltt}

2 \AfterEndPreamble{
3 \AtBeginEnvironment{alltt}{\LWR@atbeginverbatim{alltt}\unskip\vspace*{-\baselineskip}}
4 \AfterEndEnvironment{alltt}{\unskip\vspace*{-\baselineskip}\LWR@afterendverbatim}
5 }
```

## Package 7

# lwarp-amsthm.sty

## 74 AMSthm

(Based on original code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)

Pkg amsthm amsthm is patched for use by lwarp.

### CSS styling of theorems and proofs:

**Theorem:** <div> of class amsthmbody<theoremstyle>  
**Theorem Name:** <span> of class amsthmname<theoremstyle>  
**Theorem Number:** <span> of class amsthmnumber<theoremstyle>  
**Theorem Note:** <span> of class amsthmnote<theoremstyle>  
**Proof:** <div> of class amsthmproof  
**Proof Name:** <span> of class amsthmproofname  
 where <theoremstyle> is plain, definition, etc.

for HTML output: 1 \LWR@ProvidesPackagePass{amsthm}

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
3 \renewcommand{\theoremstyle}[1]{%
4   \@ifundefined{th@#1}{%
5     \PackageWarning{amsthm}{Unknown theoremstyle '#1'}%
6     \thm@style{plain}%
7   \renewcommand{\LWR@newtheoremstyle}{plain}% new
8   }{%
9     \thm@style{#1}%
10  \renewcommand{\LWR@newtheoremstyle}{#1}% new
11  }%
12 }
```

Patched to remember the style for this theorem type:

```

13 \def\xnthm#1#2{%
14   \csedef{LWR@thmstyle#2}{\LWR@newtheoremstyle}% new
15   \let\@tempa\relax
16   \exp\@ifdefinable\csname #2\endcsname{%
17     \global\exp\let\csname end#2\endcsname\@endtheorem
18     \ifx *#1% unnumbered, need to get one more mandatory arg
19       \edef\@tempa##1{%
20         \gdef\exp\@nx\csname#2\endcsname{%
21           \@nx\@thm{\exp\@nx\csname th@\the\thm@style\endcsname}%
22           }\{##1\}}%
23       \else % numbered theorem, need to check for optional arg
24         \def\@tempa{\@oparg{\ynthm{#2}}{}}%
25       \fi
26 \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#2}}}% new
27 }%
28 \@tempa
29 }

```

Patched to enclose with CSS:

```

30 \newcommand{\LWR@haveamsthmname}{
31 \renewcommand{\thmname}[1]{\InlineClass{amsthmname\LWR@thisthmstyle}{##1}}
32 }
33
34 \newcommand{\LWR@haveamsthmnumber}{
35 \renewcommand{\thmnumber}[1]{\InlineClass{amsthmnumber\LWR@thisthmstyle}{##1}}
36 }
37
38 \newcommand{\LWR@haveamsthmnote}{
39 \renewcommand{\thmnote}[1]{\InlineClass{amsthmnote\LWR@thisthmstyle}{##1}}
40 }
41
42 \LWR@haveamsthmname
43 \LWR@haveamsthmnumber
44 \LWR@haveamsthmnote

```

Patches for CSS:

```

45 \def\@begintheorem#1#2[#3]{%
46   \BlockClass{amsthmbody\LWR@thisthmstyle}% new
47   \deferred@thm@head{
48 \the\thm@headfont \thm@indent
49   \@ifempty{#1}{\let\thmname\@gobble}{\LWR@haveamsthmname}% new
50   \@ifempty{#2}{\let\thmnumber\@gobble}{\LWR@haveamsthmnumber}% new
51   \@ifempty{#3}{\let\thmnote\@gobble}{\LWR@haveamsthmnote}% new
52   \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
53   \the\thm@headpunct~
54   \thmheadnl % possibly a newline.
55   \hskip\thm@headsep
56 }%

```

```
57 \ignorespaces}
```

Patched for CSS:

```
58 \def\@endtheorem{\endBlockClass\endtrivlist\@endpefalse }
```

Proof QED symbol:

```
59 \AtBeginDocument{
60 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
61 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
62 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
63 }
```

Patched for CSS:

```
64 \renewenvironment{proof}[1][\proofname]{\par
65 \BlockClass{amsthmproof}% new
66 \pushQED{\qed}%
67 \normalfont \topsep6\p@\@plus6\p@\relax
68 \trivlist
69 \item[\hskip\labelsep
70 \InlineClass{amsthmproofname}{#1\@addpunct{.}}]\ignorespaces% changes
71 }{%
72 \InlineClass{theoremendmark}{\popQED}\endtrivlist%
73 \endBlockClass% new
74 \@endpefalse
75 }
```

## Package 8

# lwarp-bookmark.sty

## 75 Bookmark

`\Pkg bookmark` bookmark is emulated during HTML output, and the `bookmark` package is ignored.

**for HTML output:** Discard all options for `lwarp-bookmark`:

```
1 \LWR@ProvidesPackageDrop{bookmark}

2 \newcommand*{\bookmarksetup}[1]{ }
3 \newcommand*{\bookmarksetupnext}[1]{ }
4 \newcommand*{\bookmark}[2][ ]{ }
```



---

```
5 \newcommand*{\bookmarkdefinestyle}[2]{}  
6 \newcommand*{\bookmarkget}[1]{}  
7 \newcommand{\BookmarkAtEnd}[1]{}  

```

## Package 9

# lwarp-booktabs.sty

## 76 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 10

# lwarp-ccaption.sty

## 77 Ccaption

Pkg ccaption ccaption is not used. The user is recommended to use caption instead.

for HTML output: 1 \LWR@loadnever{ccaption}{caption}

## Package 11

# lwarp-changepage.sty

## 78 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 12

# lwarp-cutwin.sty

## 79 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 13

# lwarp-dcolumn.sty

## 80 Dcolumn

Pkg dcolumn dcolumn is emulated during HTML output, and the dcolumn package is ignored.

```
1 \LWR@ProvidesPackageDrop{dcolumn}
```

## Package 14

# lwarp-draftwatermark.sty

## 81 Draftwatermark

Pkg `draftwatermark` `draftwatermark` is emulated during HTML output, and the `draftwatermark` package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{draftwatermark}

2 \newcommand{\SetWatermarkAngle}[1]{}
3 \newcommand{\SetWatermarkColor}[1]{}
4 \newcommand{\SetWatermarkLightness}[1]{}
5 \newcommand{\SetWatermarkFontSize}[1]{}
6 \newcommand{\SetWatermarkScale}[1]{}
7 \newcommand{\SetWatermarkHorCenter}[1]{}
8 \newcommand{\SetWatermarkVertCenter}[1]{}
9 \newcommand{\SetWatermarkText}[1]{}

```

## Package 15

# lwarp-ellipsis.sty

## 82 Ellipsis

Pkg `ellipsis` `ellipsis` is emulated during HTML output, and the `ellipsis` package is ignored.

```
1 \LWR@ProvidesPackageDrop{ellipsis}
2
3 \newcommand{\ellipsisgap}{0.1em}

```

## Package 16

# lwarp-emptypage.sty

## 83 Emptypage

Pkg emptypage emptypage is ignored.

for **HTML output**: Discard all options for lwarp-emptypage:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

## Package 17

# lwarp-endnotes.sty

## 84 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 18

# lwarp-enumerate.sty

## 85 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 19

# lwarp-epigraph.sty

## 86 Epigraph

Pkg epigraph epigraph is emulated during HTML output, and the epigraph package is ignored.

```
for HTML output: 1 \LWR@ProvidesPackageDrop{epigraph}

2 \newcommand{\qitem}[2]
3 {
4 \begin{BlockClass}{qitem}
5 #1
6 \begin{BlockClass}{epigraphsource}
7 #2
8 \end{BlockClass}
9 \end{BlockClass}
10 }

11 \newcommand{\epigraph}[2]
12 {
13 \begin{BlockClass}{epigraph}
14 \qitem{#1}{#2}
15 \end{BlockClass}
16 }
17
18 \newenvironment*{epigraphs}
19 {\BlockClass{epigraph}}
20 {\endBlockClass}
```

Use CSS to format epigraphs.

The following are null commands for source compatibility:

```
21 \newlength{\epigraphwidth}
22 \setlength{\epigraphwidth}{.5\linewidth}
23 \newenvironment*{flushepinormal}{}{}
24 \newcommand{\textflush}[1]{flushepinormal}
25 \newcommand{\epigraphflush}[1]{flushright}
26 \newcommand{\sourceflush}[1]{flushright}
27 \newcommand*{\epigraphsize}{\small}
28 \newlength{\epigraphrule}
29 \newlength{\beforeepigraphskip}
30 \newlength{\afterepigraphskip}
31 \newcommand{\epigraphhead}[2][0]{#2}
```

```

32 \newcommand{\dropchapter}[1]{}
33 \newcommand*{\undodrop}{}
34 \newcommand{\cleartoevenpage}[1] [] {}

```

## Package 20

# lwarp-eso-pic.sty

## 87 Eso-pic

Pkg eso-pic eso-pic is emulated during HTML output, and the eso-pic package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{eso-pic}

2 \newcommand*{\LenToUnit}{}
3 \newcommand{\AtPageUpperLeft}[1]{}
4 \newcommand{\AtPageLowerLeft}[1]{}
5 \newcommand{\AtPageCenter}[1]{}
6 \newcommand{\AtStockLowerLeft}[1]{}
7 \newcommand{\AtStockUpperLeft}[1]{}
8 \newcommand{\AtStockCenter}[1]{}
9 \newcommand{\AtTextUpperLeft}[1]{}
10 \newcommand{\AtTextLowerLeft}[1]{}
11 \newcommand{\AtTextCenter}[1]{}
12 \NewDocumentCommand{\AddToShipoutPictureBG}{s +m}{}
13 \let\AddToShipoutPicture\AddToShipoutPictureBG
14 \NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
15 \newcommand*{\ClearShipoutPictureBG}{}
16 \newcommand*{\ClearShipoutPicture}{}
17 \newcommand*{\ClearShipoutPictureFG}{}
18 \newcommand{\gridSetup}[6] [] {}

```

## Package 21

# lwarp-everypage.sty

## 88 Everypage

Pkg `everypage` `everypage` is emulated during HTML output, and the `everypage` package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{everypage}

2 \newcommand*{\AddEverypageHook}[1]{}
3 \newcommand*{\AddThispageHook}[1]{}

```

## Package 22

# lwarp-extramarks.sty

## 89 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 23

# lwarp-fancyhdr.sty

## 90 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 24

# lwarp-float.sty

## 91 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 52.2 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 25

# lwarp-floatflt.sty

## 92 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 \DeclareDocumentEnvironment{KFLT@marginfloat}{0{-1.2ex} m}
3 {% start
4 \LWR@maybeinthisfloat%
5 % \BlockClass{marginblock}
6 \LWR@stoppars%
7 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
8 \LWR@startpars%
9 \captionsetup{type=#2}%
10 }
11 {
12 % \endBlockClass
13 \LWR@htmldivclassend{div}
14 }
15
16 \DeclareDocumentEnvironment{floatingfigure}{o m}
17 {\begin{KFLT@marginfloat}{figure}}
18 {\end{KFLT@marginfloat}}
19
20 \DeclareDocumentEnvironment{floatingtable}{o +m}
21 {\begin{KFLT@marginfloat}{table}#2}
22 {\end{KFLT@marginfloat}}
```




## Package 26


# lwarp-floatrow.sty

## 93 Floatrow

`Pkg floatrow` floatrow is emulated during HTML output, and the floatrow package is ignored.

**for HTML output:** `1 \LWR@ProvidesPackageDrop{floatrow}`

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

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

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

After everything has loaded, remember whether subcaption was loaded. If not, it is assumed that subfig is used instead:

```
2 \newbool{LWR@subcaptionloaded}
3
4 \AtBeginDocument{
5 \@ifpackageloaded{subcaption}
6 {\booltrue{LWR@subcaptionloaded}}
7 {\boolfalse{LWR@subcaptionloaded}}
8 }
```

```
\floatbox [1 preamble] {2 captype} [3 width] [4 height] [5 vert pos]
{6 caption} {7 object}
```

Only parameters for captype, width, caption, and object are used.

LWR@insubfloatrow is true if inside a subfloatrow environment.

There are two actions, depending on the use of `subcaption` or `subfig`.

```

9 \NewDocumentCommand{\floatbox}{o m o o +m +m}{%
10 \ifbool{LWR@subcaptionloaded}%
11 {% subcaption

```

For `subcaption`:

```

12 \ifbool{LWR@insubfloatrow}%
13 {% subcaption in a subfloatrow

```

`subfigure` and `subtable` environments take width as an argument.

```

14 \IfValueTF{#3}%
15 {\@nameuse{sub#2}{#3}}%
16 {\@nameuse{sub#2}{\linewidth}}%
17 }% subcaption in a subfloatrow
18 {% subcaption not in subfloatrow

```

`figure` and `table` environments do not take a width argument.

```

19 \@nameuse{#2}%
20 }% subcaption not in subfloatrow
21 #6
22
23 #7

```

End the environments:

```

24 \ifbool{LWR@insubfloatrow}%
25 {\@nameuse{endsub#2}}%
26 {\@nameuse{end#2}}%
27 }% subcaption
28 {% assume subfig

```

For `subfig`:

```

29 \ifbool{LWR@insubfloatrow}%
30 {% subfig in a subfloatrow

```

`\subfloat` is a macro, not an environment.

Package `subfig`'s `\subfloat` command takes an optional argument which is the caption, but `\floatbox` argument `#6` contains commands to create the caption and label, not the caption itself. Thus, `\caption` is temporarily disabled to return its own argument without braces.

```

31 \begingroup
32 \let\caption\@firstofone

```

```

33 \subfloat[#6]{#7}
34 \endgroup
35 }% subfig in a subfloatrow
36 {% subfig package, but not a subfig

```

figure and table are environments:

```

37 \@nameuse{#2}
38 #6
39
40 #7
41 \@nameuse{end#2}
42 }% subfig package, but not a subfig
43 }% assume subfig
44 }

```

Not used:

```

45 \newcommand*{\nocapbeside}{}
46 \newcommand*{\capbeside}{}
47 \newcommand*{\captop}{}
48 \newlength{\FBwidth}
49 \setlength{\FBwidth}{.3\linewidth}
50 \newlength{\FBheight}
51 \setlength{\FBheight}{2in}
52 \newcommand*{\useFCwidth}{}
53 \newcommand{\floatsetup}[2][{}]{
54 \newcommand{\thisfloatsetup}[1]{
55 \newcommand{\clearfloatsetup}[1]{
56 \newcommand*{\killfloatstyle}{}

```

Preamble and default width are ignored.

```

57 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
58 \@namedef{#1}{
59 \floatbox{#2}
60 }
61 }

```

Preamble and default width are ignored.

```

62 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
63 \@namedef{#1}{%
64 \floatbox{#2}
65 }
66 }

67 \newfloatcommand{ffigbox}{figure}[\nocapbeside] []

```

```
68 \newfloatcommand{ttabbox}{table}[\captop][\FBwidth]
```

```
69 \newfloatcommand{fcapside}{figure}[\capbeside][]
```

The row of floats is placed into a <div> of class floatrow.

```
70 \newenvironment*{floatrow}[1][2]
71 {
72 \BlockClass{floatrow}
```

While inside the floatrow, divide the \linewidth by the number of floats.

```
73 \booltrue{LWR@infloatrow}
74 \setlength{\linewidth}{6in/#1}
75 }
76 {
77 \boolfalse{LWR@infloatrow}
78 \endBlockClass
79 }
```

Keys for \DeclareNewFloatType:

```
80 \newcommand*{\LWR@frowkeyplacement}{}
81 \newcommand*{\LWR@frowkeyname}{}
82 \newcommand*{\LWR@frowkeyfileext}{}
83 \newcommand*{\LWR@frowkeywithin}{}
84 \newcommand*{\LWR@frowkeycapstyle}{}
85
86 \define@key{frowkeys}{placement}{}%
87 \define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}%
88 \define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}%
89 \define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}%
90 \define@key{frowkeys}{relatedcapstyle}{}%
```

Use \listof{type}{Title} to print a list of the floats.

```
91 \newcommand*{\DeclareNewFloatType}[2]{%
```

Reset key values:

```
92 \renewcommand*{\LWR@frowkeyplacement}{}
93 \renewcommand*{\LWR@frowkeyname}{}
94 \renewcommand*{\LWR@frowkeyfileext}{}
95 \renewcommand*{\LWR@frowkeywithin}{}
96 \renewcommand*{\LWR@frowkeycapstyle}{}%
```

Read new key values:

```

97 \LWR@traceinfo{about to setkeys frowkeys}%
98 \setkeys{frowkeys}{#2}%
99 \LWR@traceinfo{finished setkeys frowkeys}%

```

Create a new float with optional [within]:

```

100 \ifthenelse{\equal{\LWR@frowkeywithin}{}}{
101 {
102 \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\
103 \LWR@frowkeyfileext}%
104 \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}
105 }%
106 {%
107 \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\
108 \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
109 \newfloat{#1}{\LWR@frowkeyplacement}%
110 {\LWR@frowkeyfileext}[\LWR@frowkeywithin}%
111 \LWR@traceinfo{finished newfloat #1}
112 }%

```

Rename the float if a name was given:

```

113 \ifthenelse{\equal{\LWR@frowkeyname}{}}{
114 {}
115 {\floatname{#1}{\LWR@frowkeyname}}%
116 }

```

Not used:

```

117 \newcommand{\buildFBBBOX}[2]{}
118 \newcommand*{\CenterFloatBoxes}{}
119 \newcommand*{\TopFloatBoxes}{}
120 \newcommand*{\BottomFloatBoxes}{}
121 \newcommand*{\PlainFloatBoxes}{}
122
123 \newcommand{\capsubrowsettings}{}
124
125 \NewDocumentCommand{\RawFloats}{o o}{}

```

To be used inside a minipage or parbox.

```

126 \newcommand{\RawCaption}[1]{#1}

```

Places additional text inside a float, inside a CSS <div> of class floatfoot.

```

127 \NewDocumentCommand{\floatfoot}{s +m}{%
128 \begin{BlockClass}{floatfoot}
129 #2
130 \end{BlockClass}

```

```
131 }
```

Used to compute `\linewidth`.

```
132 \newbool{LWR@insubfloatrow}  
133 \boolfalse{LWR@insubfloatrow}
```

```
134 \newenvironment*{subfloatrow}[1][2]  
135 {
```

The row of floats is placed into a `<div>` of class `floatrow`:

```
136 \BlockClass{floatrow}
```

While inside the `floatrow`, `LWR@insubfloatrow` is set true, which tells `\floatbox` to use `\subfigure` or `\subtable`.

```
137 \begingroup  
138 \booltrue{LWR@insubfloatrow}  
139 }  
140 {  
141 \endgroup  
142 \endBlockClass  
143 \boolfalse{LWR@insubfloatrow}  
144 }
```

## Package 27

# lwarp-fontenc.sty

## 94 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 28

# lwarp-fontspec.sty

## 95 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 29

# lwarp-footmisc.sty

## 96 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 30

# lwarp-footnote.sty

## 97 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 31

# lwarp-footnotehyper.sty

## 98 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 32

# lwarp-framed.sty

## 99 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 \BlockClass{framed}%
5 }
6 {\endBlockClass}
7
8 \renewenvironment{oframed}{%
9 \BlockClass{framed}%
10 }
11 {\endBlockClass}
12
13
14 \renewenvironment{shaded}{%
15 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
16 \BlockClass{framed}[background: \#\LWR@tempcolor]%
17 }
18 {\endBlockClass}
19
20 \renewenvironment{shaded*}{%
21 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
22 \BlockClass{framed}[background: \#\LWR@tempcolor]%
23 }
24 {\endBlockClass}
25
26
27 \renewenvironment{leftbar}{%
28 \BlockClass{framedleftbar}
29 \def\FrameCommand{}%
30 \MakeFramed {}
31 }%
32 {\endMakeFramed\endBlockClass}
33
34
```

```

35 \renewenvironment{snugshade}{%
36 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
37 \BlockClass{snugframed}[background: \#\LWR@tempcolor]%
38 }
39 {\endBlockClass}
40
41 \renewenvironment{snugshade*}{%
42 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
43 \BlockClass{snugframed}[background: \#\LWR@tempcolor]%
44 }
45 {\endBlockClass}
46
47 \let\oframed\framed
48 \let\endoframed\endframed
49
50
51 \RenewEnviron{titled-frame}[1]{%
52 \CustomFBox{#1}{\Opt}{0pt}{0pt}{0pt}{\BODY}
53 }

\CustomFBox {<toptitle>} {<bottitle>} {<thicknesstop>} {<bottom>} {<left>} {<right>}
{<text contents>}

54 \renewcommand{\CustomFBox}[7]{%
55 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
56 \begin{BlockClass}{framed}[border: 3px solid \#\LWR@tempcolor]%
57 \ifthenelse{\isempty{#1}}{\}% not empty
58 \begin{BlockClass}{framedtitle}[background: \#\LWR@tempcolor]%
59 \textcolor{TFTitleColor}{\textbf{#1}}%
60 \end{BlockClass}
61 }% not empty
62
63 #7
64
65 \ifthenelse{\isempty{#2}}{\}% not empty
66 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
67 \begin{BlockClass}{framedtitle}[background: \#\LWR@tempcolor]%
68 \textcolor{TFTitleColor}{\textbf{#2}}%
69 \end{BlockClass}
70 }% not empty
71 \end{BlockClass}
72 }

\TitleBarFrame [{<marker>}] {<title>} {<contents>}

73 \renewcommand\TitleBarFrame[3][{}{
74 \CustomFBox
75   {#2}{}%
76   \fboxrule\fboxrule\fboxrule\fboxrule

```

```
77     {#3}%  
78 }  
  
79 \renewcommand{\TF@Title}[1]{#1}  
  
MakeFramed {\langle settings \rangle}  
  
80 \let\MakeFramed\relax  
81 \let\endMakeFramed\relax  
82  
83 \NewEnviron{MakeFramed}[1]{%  
84 \FrameCommand{\begin{minipage}{\linewidth}\BODY\end{minipage}}%  
85 }  
  
\fb@put@frame {\langle frame cmd no split \rangle} {\langle frame cmd split \rangle}  
  
86 \renewcommand*\fb@put@frame[2]{%  
87 \relax%  
88 \@tempboxa%  
89 }
```

## Package 33

# lwarp-ftnright.sty

## 100 Ftnright

Pkg `ftnright` `ftnright` is ignored.

for **HTML output:** Discard all options for `lwarp-ftnright`:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

## Package 34

# lwarp-geometry.sty

## 101 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 35

# lwarp-graphics.sty

## 102 Graphics

Pkg graphics graphics is not used. The user is required to use graphicx instead.

for HTML output: 1 \LWR@loadnever{graphics}{graphicx}

## Package 36

# lwarp-graphicx.sty

## 103 Graphicx

Pkg graphicx graphicx is emulated during HTML output, and the graphicx package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphicx}

## Package 37

# lwarp-hyperref.sty

## 104 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}{0{label} +m}{%
33 \LWR@startref{#1}%
34 #2%
35 \LWR@htmltag{/a}%
36 \endgroup%
37 }

```

```

38 \newcommand{\hyperref}{%
39 \LWR@ensuredoingapar%
40 \begingroup\catcode'\_ =12
41 \@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
42 }

```

Creates an anchor to name with the given text.

```

43 \NewDocumentCommand{\hypertarget}{m +m}{%
44 \label{#1}%
45 #2%
46 }

```

Creates a link to the anchor created by `hypertarget`, with the given link text.

```

47 \NewDocumentCommand{\hyperlink}{m +m}{%
48 \hyperref{#1}{#2}%
49 }

```

For HTML, `\cleverref` is used instead.

```

50 \NewDocumentCommand{\autoref}{s m}{%
51 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
52 }

```

For HTML, `\cleverref` is used instead.

```

53 \NewDocumentCommand{\autopageref}{s m}{%
54 \IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
55 }

```

```

56 \newcommand{\pdfstringdef}[2]{}

```

```

57 \newcommand{\pdfbookmark}[3][]{ }

```

```
58 \newcommand{\currentpdfbookmark}[2]{}{}
```

```
59 \newcommand{\subpdfbookmark}[2]{}{}
```

```
60 \newcommand{\belowpdfbookmark}[2]{}{}
```

```
61 \newcommand{\texorpdfstring}[2]{#2}
```

From hyperref.

```
62 \def\hypercalcbp#1{%  
63 \strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax  
64 }%
```

```
65 \newcommand{\Acrobatmenu}[2]{}{}
```

```
66 \newcommand*{\TextField}[2][{}]{}
```

```
67 \newcommand*{\CheckBox}[2][{}]{}
```

```
68 \newcommand{\ChoiceMenu}[3][{}]{}
```

```
69 \newcommand*{\PushButton}[2][{}]{}
```

```
70 \newcommand*{\Submit}[2][{}]{}
```

```
71 \newcommand*{\Reset}[2][{}]{}
```

```
72 \newcommand*{\LayoutTextField}[2]{}{}
```

```
73 \newcommand*{\LayoutChoiceField}[2]{}{}
```

```
74 \newcommand*{\LayoutCheckField}[2]{}{}
```

```
75 \newcommand*{\MakeRadioField}[2]{}{}
```

```
76 \newcommand*{\MakeCheckField}[2]{}{}
```

```
77 \newcommand*{\MakeTextField}[2]{}{}
```

```
78 \newcommand*{\MakeChoiceField}[2]{}{}
```

```
79 \newcommand{\MakeFieldButton}[1]{}{}
```

## Package 38

# lwarp-indentfirst.sty

## 105 Indentfirst

Pkg indentfirst indentfirst is ignored.

Discard all options for lwarp-indentfirst:

for HTML output: 1 \LWR@ProvidesPackageDrop{indentfirst}

## Package 39

# lwarp-inputenc.sty

## 106 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

## 107 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 % \BlockClass{marginblock}
18 \LWR@stoppars%
19 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}}{
20 \LWR@startpars%
21 \captionsetup{type=#2}%
22 }
23 {
24 % \endBlockClass
25 \LWR@htmldivclassend{div}
26 }

27 }% AtBeginDocument

```

## Package 41

# lwarp-layout.sty

## 108 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

## 109 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

## 110 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

## 111 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

## 112 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}{lol}{#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@atbeginverbatim{programlisting}%
18
19 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
20 \else%
```



Inline, so open a `<span>`

```
21 \ifbool{LWR@verbtags}{\LWR@htmltag{span class="inlineprogramlisting"}}{}%
22 \fi%
23 }
```

```
24 \renewcommand*\lst@DeInit{%
25 \lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the `</pre>`, then reenable line numbers:

```
26 \let\lsthk@EveryPar\relax%
27
28 \LWR@afterendverbatim%
29 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
30 \else%
```

Inline, so create the closing `</span>`:

```
31 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
32 \fi%
```

Final listings deinit:

```
33 \LWR@origlst@DeInit%
34 }
```

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```
35 \def\lst@MakeCaption#1{%
36 \LWR@traceinfo{MAKING CAPTION at #1}%
37 \lst@ifdisplaystyle
38 \LWR@traceinfo{making a listings display caption}%
39 \ifx #1t%
40 \ifx\lst@@caption\@empty\expandafter\lst@HRefStepCounter \else
41 \expandafter\refstepcounter
42 \fi {lstlisting}%
43 \LWR@traceinfo{About to assign label: !\lst@label!}%
44 % \ifx\lst@label\@empty\else
45 % \label{\lst@label}\fi
46 \LWR@traceinfo{Finished assigning the label.}%
47 \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
48 \global\let\lst@name\lst@arg \global\let\lst@name\lst@name
49 \lst@ifnolol\else
50 \ifx\lst@@caption\@empty
```

```

51             \ifx\lst@caption\@empty
52             \ifx\lst@intname\@empty \else \def\lst@temp{ }%
53             \ifx\lst@intname\lst@temp \else

```

This code places a contents entry for a non-float. This would have to be modified for lwarp:

```

54 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
55 %             \addcontentsline{lol}{lstlisting}{\lst@name}
56             \fi\fi
57             \fi
58             \else

```

This would have to be modified for lwarp:

```

59 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
60             \addcontentsline{lol}{lstlisting}%
61 {\protect\numberline{\thelstlisting}%
62 {\protect\ignorespaces \lst@@caption \protect\relax}}%
63             \fi
64             \fi
65             \fi
66             \ifx\lst@caption\@empty\else
67 \LWR@traceinfo{lst@caption not empty-}%
68             \lst@ifsubstring #1\lst@captionpos
69             {\begingroup
70 \LWR@traceinfo{at the selected position}%

```

These space and box commands are not needed for HTML output:

```

71 %             \let\@vskip\vskip
72 %             \def\vskip{\afterassignment\lst@vskip \@tempskipa}%
73 %             \def\lst@vskip{\nobreak\@vskip\@tempskipa\nobreak}%
74 %             \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
75 %             \ifx #1t\allowbreak \fi
76             \ifx\lst@title\@empty

```

New lwarp code to create a caption:

```

77             \lst@makecaption\fnnum\lstlisting{\ignorespaces \lst@caption}
78             \else

```

New lwarp code to create a title:

```

79 %             \lst@maketitle\lst@title % (AS)
80 \LWR@traceinfo{Making title: \lst@title}%
81 \begin{BlockClass}{lstlistingtitle}% lwarp
82 \lst@maketitle\lst@title% lwarp
83 \end{BlockClass}% lwarp

```

```

84          \fi
85 \LWR@traceinfo{About to assign label: !\lst@label!}%
86          \ifx\lst@label\@empty\else
87 \leavevmode% gets rid of bad space factor error
88 \GetTitleStringExpand{\lst@caption}%
89 \edef\LWR@lntemp{\GetTitleStringResult}%
90 \edef\@currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
91 \label{\lst@label}\fi
92 \LWR@traceinfo{Finished assigning the label.}%

```

Not needed for lwarp:

```

93 %          \ifx #1b\allowbreak \fi
94          \endgroup}{}%
95 \fi
96 \LWR@traceinfo{end of making a listings display caption}%
97 \else
98 \LWR@traceinfo{INLINE}%
99 \fi
100 \LWR@traceinfo{DONE WITH CAPTION at #1}%
101 }

```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```

102 \lst@Key{numbers}{none}{%
103   \let\lst@PlaceNumber\@empty
104   \lstKV@SwitchCases{#1}%
105   {none&\\%
106     left&\def\lst@PlaceNumber{%
107 % \llap{
108 \LWR@orignormalfont%
109 \lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
110 % }
111 }
112 \\%
113   right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont
114     \kern\VerbatimHTMLWidth \kern\lst@numbersep
115     \lst@numberstyle{\thelstnumber}}}%
116   }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}
117 \end{warpHTML}

```

## Package 46

# lwarp-longtable.sty

## 113 Longtable

Pkg `longtable` `longtable` is emulated during HTML output, and the `longtable` package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{longtable}`

To emulate `longtable`:



For `longtable` `\endhead`, `\endfoot`, and `\endlastfoot` rows, 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.

See:

<http://tex.stackexchange.com/questions/43006/why-is-input-not-expandable>

Env `longtable` \* `[\horizontalalignment] {\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{\LTcaption}%

```

---

```

17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{\LTcaptype}%
21 \let\captionlistentry\LWR@LTcaptionlistentry%
22 \LWR@tabular{#2}
23 }
24 {\endLWR@tabular\LWR@floatend}
25

```

Provided for compatibility, but ignored:

```

26 \newcounter{LTchunksize}
27 \def\endhead{\LWR@tabularendofline}% 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}

\kill is ignored, place a \kill line inside
    \begin{warpprint} ... \end{warpprint} or \warpingprintonly.

38 \renewcommand*{\kill}{\LWR@tabularendofline}

```

## Package 47

# lwarp-lscape.sty

## 114 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

## 115 Ltcaption

Pkg ltcaption ltcaption is emulated during HTML output, and the ltcaption package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{ltcaption}

\LTcaptype is already defined by lwarp.

longtable\* is already defined by lwarp-longtable.

```
2 \newlength{\LTcapskip}
```

```
3 \newlength{\LTcapleft}
```

```
4 \newlength{\LTcapright}
```

```
5 \newcommand*{\LTcapmarginfalse}{}

```

## Package 49

## lwarp-marginfix.sty

## 116 Marginfix

Pkg marginfix Not used.

for HTML output: Discard all options for lwarp-marginfix:

```
1 \LWR@ProvidesPackageDrop{marginfix}

2 \newcommand*{\marginskip}[1]{}
3 \newcommand*{\clearmargin}{}
4 \newcommand*{\softclearmargin}{}
5 \newcommand*{\extendmargin}[1]{}
6 \newcommand*{\mparshift}[1]{}
7 \newdimen\marginheightadjustment
8 \newdimen\marginposadjustment
9 \newcommand*{\blockmargin}[1][\marginheightadjustment]{}
10 \newcommand*{\unblockmargin}[1][\marginposadjustment]{}
11 \newcommand*{\marginphantom}[2][\marginheightadjustment]{}

```

## Package 50

# lwarp-marginnote.sty

## 117 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

## 118 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

## 119 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**.

For title font, use

`frametitlefont=\textbf,`

**font** 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 }

```

$[\langle \textit{numberedlike} \rangle]$   $\{\langle \textit{caption} \rangle\}$   $[\langle \textit{within} \rangle]$

Modified from the original to remember the environment.

```

139 \let\LWR@origmdtheorem\mdtheorem
140
141 \DeclareDocumentCommand{\LWR@mdtheorem}{0}{ m o m o }{%
142 \LWR@origmdtheorem[#1]{#2}[#3]{#4}[#5}%
143 \BeforeBeginEnvironment{#2}{\renewcommand*\LWR@mdthisenv}{md#2}}%
144 }
145
146 \let\mdtheorem\LWR@mdtheorem

```

$[\langle \textit{numberedlike} \rangle]$   $\{\langle \textit{caption} \rangle\}$   $[\langle \textit{within} \rangle]$

Modified from the original to remember the environment.

```

147 \DeclareDocumentCommand\newmdtheoremenv{0}{ m o m o }{%
148 \ifboolexpr{ test {\IfNoValueTF {#3}} and test {\IfNoValueTF {#5}} }%
149   {\newtheorem{#2}{#4}}{%
150     \IfValueTF{#3}{\newtheorem{#2}[#3]{#4}}{%
151       \IfValueTF{#5}{\newtheorem{#2}{#4}[#5]}{%
152         }%
153     }%
154   \BeforeBeginEnvironment{#2}{%
155     \renewcommand*\LWR@mdthisenv}{md#2}%
156     \begin{mdframed}[#1]}%
157   \AfterEndEnvironment{#2}{%
158     \end{mdframed}}%
159 }

```

## Package 53

# lwarp-microtype.sty

## 120 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

## 121 Mparhack

Pkg mparhack Not used.

for **HTML output:** Discard all options for lwarp-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```

## Package 55

# lwarp-multicol.sty

## 122 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 {\BlockClass{multicols}
```

Optional HTML div class for the heading:

```
5 \IfValueTF{#3}{\begin{BlockClass}{multicolsheading}#3\end{BlockClass}}{}}
```

When done with the environment, close the div:

```
6 {\endBlockClass}
```

Emulated null functions which are not used in HTML:

```
7 \newcommand*{\columnbreak}{}
8 \newcommand*{\RLmulticolcolumns}{}
9 \newcommand*{\LRmulticolcolumns}{}
10
11 \newlength{\premulticols}
12 \newlength{\postmulticols}
13 \newlength{\multicolsep}
14 \newlength{\multicolbaselineskip}
15 \newlength{\multicoltolerance}
16 \newlength{\multicolpretolerance}
17 \newcommand*{\columnseprulecolor}{\normalcolor}
```

---

```
18 \newcounter{columnbadness}
19 \newcounter{finalcolumnbadness}
20 \newcounter{collectmore}
21 \newcounter{unbalance}
22 \newlength{\multicolovershoot}
23 \newlength{\multicolundershoot}

24 \end{warpHTML}
```

## Package 56

# lwarp-multirow.sty

## 123 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

## 124 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

## 125 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

## 126 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

## 127 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

## 128 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

## 129 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

## 130 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{ntheorem}

```

Storage for the style being used for new theorems.

```
24 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```

25 \gdef\theoremstyle#1{%
26   \ifundefined{th@#1}{\@warning
27     {Unknown theoremstyle ‘#1’. Using ‘plain’}%
28     \theorem@style{plain}
29     \renewcommand{\LWR@newtheoremstyle}{plain}% new
30 }%
31   {
32 \theorem@style{#1}
33   \renewcommand{\LWR@newtheoremstyle}{#1}% new
34 }
35 }

```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

36
37 \gdef\xnthm#1#2[#3]{%
38   \ifthm@tempif
39     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
40     \expandafter\@ifundefined{c@#1}%
41       {\@definecounter{#1}}{}%
42     \@newctr{#1}[#3]%
43     \expandafter\xdef\csname the#1\endcsname{%
44       \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
45       {\noexpand\csname\the\theoremnumbering\endcsname{#1}}}%
46     \expandafter\gdef\csname mkheader@#1\endcsname

```

```

47     {\csname setparms@#1\endcsname
48       \@thm{#1}{#1}{#2}
49 }%
50   \global\@namedef{end#1}{\@endtheorem}
51   \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
52   \fi
53 }
54
55 \gdef\@ynthm#1#2{%
56   \ifthm@tempif
57     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
58     \expandafter\@ifundefined{c@#1}%
59       {\@definecounter{#1}}{}%
60     \expandafter\xdef\csname the#1\endcsname
61       {\noexpand\csname\the\theoremnumbering\endcsname{#1}}%
62     \expandafter\gdef\csname mkheader@#1\endcsname
63       {\csname setparms@#1\endcsname
64         \@thm{#1}{#1}{#2}
65 }%
66     \global\@namedef{end#1}{\@endtheorem}
67     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
68     \fi
69 }
70
71 \gdef\@othm#1[#2]#3{%
72   \@ifundefined{c@#2}{\@nocounterr{#2}}%
73   {\ifthm@tempif
74     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
75     \global\@namedef{the#1}{\@nameuse{the#2}}%
76     \expandafter\protected@xdef\csname num@addtheoremeline#1\endcsname{%
77       \noexpand\@num@addtheoremeline{#1}{#3}}%
78     \expandafter\protected@xdef\csname nonum@addtheoremeline#1\endcsname{%
79       \noexpand\@nonum@addtheoremeline{#1}{#3}}%
80     \theoremkeyword{#3}%
81     \expandafter\protected@xdef\csname #1Keyword\endcsname
82       {\the\theoremkeyword}%
83     \expandafter\gdef\csname mkheader@#1\endcsname
84       {\csname setparms@#1\endcsname
85         \@thm{#1}{#2}{#3}
86 }%
87     \global\@namedef{end#1}{\@endtheorem}
88     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
89     \fi}
90 }

```

Mimics a float by incrementing the float counter and generating an HTML anchor. These are used for list-of-theorem cross-references.

```

91 \newcommand{\LWR@inctheorem}{%

```

```

92 \addtocounter{LWR@thisfloat}{1}%
93 \LWR@stoppars%
94 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
95 \LWR@startpars%
96 }

```

The following are patched for CSS.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc. They are gathered together here.

Each theorem is encased in a `BlockClass` environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader<style>`.

```

97 \gdef\newtheoremstyle#1#2#3{%
98   \expandafter\@ifundefined{th@#1}%
99   {\expandafter\gdef\csname th@#1\endcsname{%
100     \def\@begintheorem####1####2{%
101       \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
102       \LWR@inctheorem% new
103     }#2}%
104     \def\@opargbegintheorem####1####2####3{%
105       \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
106       \LWR@inctheorem% new
107     }#3}%
108   }%
109 }%
110 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
111 }
112
113 \renewtheoremstyle{plain}%
114   {\item[\hskip\labelsep \theorem@headerfont
115 \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
116   {\item[\hskip\labelsep \theorem@headerfont
117 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
118
119 \renewtheoremstyle{break}%
120   {\item[
121 % \rlap{\vbox{\hbox{
122 \hskip\labelsep \theorem@headerfont
123 \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
124 % }\hbox{\strut}}}}
125 ]}%
126   {\item[
127 % \rlap{\vbox{\hbox{
128 \hskip\labelsep \theorem@headerfont
129 \InlineClass{theoremheaderbreak}{##1\ ##2\ (##3)\theorem@separator}\newline
130 % }\hbox{\strut}}}}
131 ]}

```

```

132
133 \renewtheoremstyle{change}%
134   {\item[\hskip\labelsep
135 \theorem@headerfont
136 \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]]}%
137   {\item[\hskip\labelsep
138 \theorem@headerfont
139 \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]]}
140
141 \renewtheoremstyle{changebreak}%
142   {\item[
143 % \rlap{\vbox{\hbox{
144 \hskip\labelsep \theorem@headerfont
145 \InlineClass{theoremheaderchangebreak}{##2\ ##1\theorem@separator}\newline
146 % }\hbox{\strut}}}}
147 ]}%
148   {\item[
149 % \rlap{\vbox{\hbox{
150 \hskip\labelsep \theorem@headerfont
151 \InlineClass{theoremheaderchangebreak}{##2\ ##1\ (##3)\theorem@separator}\newline
152 % }\hbox{\strut}}}}
153 ]}
154
155 \renewtheoremstyle{margin}%
156   {\item[\hskip\labelsep\theorem@headerfont
157 \InlineClass{theoremheadermargin}{##2 \quad ##1\theorem@separator}
158 ]}%
159   {\item[\hskip\labelsep\theorem@headerfont
160 \InlineClass{theoremheadermargin}{##2 \quad ##1\ (##3)\theorem@separator}
161 ]}
162
163 \renewtheoremstyle{marginbreak}%
164   {\item[\hskip\labelsep\theorem@headerfont
165 \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\theorem@separator}\newline
166 ]}%
167   {\item[\hskip\labelsep\theorem@headerfont
168 \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\ (##3)\theorem@separator}\newline
169 ]}
170
171 \renewtheoremstyle{nonumberplain}%
172   {\item[\theorem@headerfont\hskip\labelsep
173 \InlineClass{theoremheaderplain}{##1\theorem@separator}]]}%
174   {\item[\theorem@headerfont\hskip \labelsep
175 \InlineClass{theoremheaderplain}{##1\ (##3)\theorem@separator}]]}
176
177 \renewtheoremstyle{nonumberbreak}%
178   {\item[
179 % \rlap{\vbox{\hbox{
180 \hskip\labelsep \theorem@headerfont
181 \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline

```

```

182 % }\hbox{\strut}}
183 ]}%
184 {\item[
185 % \rlap{\vbox{\hbox{
186 \hskip\labelsep \theorem@headerfont
187 \InlineClass{theoremheaderbreak}{##1\ (##3)\theorem@separator}\newline
188 % }\hbox{\strut}}
189 ]}
190
191 \renewtheoremstyle{empty}%
192 {\item[]}%
193 {\item[\theorem@headerfont \hskip\labelsep\relax
194 \InlineClass{theoremheaderplain}{##3}]}
195
196 \renewtheoremstyle{emptybreak}%
197 {\item[]}%
198 {\item[\theorem@headerfont \hskip\labelsep\relax
199 \InlineClass{theoremheaderplain}{##3}] \ \newline}

```

The following manually adjust the CSS for the standard configuration objects which are not a purely plain style:

```

200 \ifbool{LWR@theoremamsthm}{-}{-}
201 % upright text via CSS
202 \newtheoremstyle{plainupright}%
203 {\item[\hskip\labelsep \theorem@headerfont
204 \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
205 {\item[\hskip\labelsep \theorem@headerfont
206 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
207
208 % upright text and small caps header via CSS
209 \newtheoremstyle{nonumberplainuprightsc}%
210 {\item[\theorem@headerfont\hskip\labelsep
211 \InlineClass{theoremheadersc}{##1\theorem@separator}]}%
212 {\item[\theorem@headerfont\hskip \labelsep
213 \InlineClass{theoremheadersc}{##1\ (##3)\theorem@separator}]}

```

The following standard configuration is renewed using the new CSS:

```

214 \theoremstyle{plainupright}
215 \theorembodyfont{\upshape}
216 \theoremsymbol{\ensuremath{\_ \Box}}
217 \renewtheorem{Example}{Example}
218 \renewtheorem{example}{Example}
219 \renewtheorem{Beispiel}{Beispiel}
220 \renewtheorem{beispiel}{Beispiel}
221 \renewtheorem{Bemerkung}{Bemerkung}
222 \renewtheorem{bemerkung}{Bemerkung}
223 \renewtheorem{Anmerkung}{Anmerkung}

```

```

224 \renewtheorem{anmerkung}{Anmerkung}
225 \renewtheorem{Remark}{Remark}
226 \renewtheorem{remark}{Remark}
227 \renewtheorem{Definition}{Definition}
228 \renewtheorem{definition}{Definition}
229
230 \theoremstyle{nonumberplainuprightsc}
231 \theoremsymbol{\ensuremath{\_\blacksquare}}
232 \renewtheorem{Proof}{Proof}
233 \renewtheorem{proof}{Proof}
234 \renewtheorem{Beweis}{Beweis}
235 \renewtheorem{beweis}{Beweis}
236 \qedsymbol{\ensuremath{\_\blacksquare}}
237
238 \theoremsymbol{}
239 }% not amsthm

```

Only if the amsthm option was given:

```

240 \ifbool{LWR@theoremamsthm}{
241
242 \gdef\th@plain{%
243   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
244   \def\@begintheorem##1##2{%
245     \BlockClass{theorembodyplain}% new
246     \LWR@intheorem% new
247     \item[\hskip\labelsep
248 % \theorem@headerfont
249     \InlineClass{theoremheaderplain}{##1\ ##2.}
250 ]}%
251   \def\@opargbegintheorem##1##2##3{%
252     \BlockClass{theorembodyplain}% new
253     \LWR@intheorem% new
254     \item[\hskip\labelsep
255 % \theorem@headerfont
256     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
257 ]}}
258
259 \gdef\th@nonumberplain{%
260   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
261   \def\@begintheorem##1##2{%
262     \BlockClass{theorembodyplain}% new
263     \LWR@intheorem% new
264     \item[\hskip\labelsep
265 % \theorem@headerfont
266     \InlineClass{theoremheaderplain}{##1.}
267 ]}%
268   \def\@opargbegintheorem##1##2##3{%
269     \BlockClass{theorembodyplain}% new
270     \LWR@intheorem% new

```

```

271 \item[\hskip\labelsep
272 % \theorem@headerfont
273 \InlineClass{theoremheaderplain}{##1\ (##3).}
274 ]}]
275
276 \gdef\th@definition{%
277 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
278 \def\@begintheorem##1##2{%
279 \BlockClass{theorembodydefinition}% new
280 \LWR@intheorem% new
281 \item[\hskip\labelsep
282 % \theorem@headerfont
283 \InlineClass{theoremheaderdefinition}{##1\ ##2.}
284 ]}]%
285 \def\@opargbegintheorem##1##2##3{%
286 \BlockClass{theorembodydefinition}% new
287 \LWR@intheorem% new
288 \item[\hskip\labelsep
289 % \theorem@headerfont
290 \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
291 ]}]
292
293 \gdef\th@nonumberdefinition{%
294 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
295 \def\@begintheorem##1##2{%
296 \BlockClass{theorembodydefinition}% new
297 \LWR@intheorem% new
298 \item[\hskip\labelsep
299 % \theorem@headerfont
300 \InlineClass{theoremheaderdefinition}{##1.}
301 ]}]%
302 \def\@opargbegintheorem##1##2##3{%
303 \BlockClass{theorembodydefinition}% new
304 \LWR@intheorem% new
305 \item[\hskip\labelsep
306 % \theorem@headerfont
307 \InlineClass{theoremheaderdefinition}{##1\ (##3).}
308 ]}]
309
310 \gdef\th@remark{%
311 \def\theorem@headerfont{\itshape}\normalfont%
312 \def\@begintheorem##1##2{%
313 \BlockClass{theorembodyremark}% new
314 \LWR@intheorem% new
315 \item[\hskip\labelsep
316 % \theorem@headerfont
317 \InlineClass{theoremheaderremark}{##1\ ##2.}
318 ]}]%
319 \def\@opargbegintheorem##1##2##3{%
320 \BlockClass{theorembodyremark}% new

```

---

```

321 \LWR@inctheorem% new
322     \item[\hskip\labelsep
323 % \theorem@headerfont
324 \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
325 ]}}
326
327 \gdef\th@nonumberremark{%
328 \def\theorem@headerfont{\itshape}\normalfont%
329 \def\@begintheorem##1##2{%
330 \BlockClass{theorembodyremark}% new
331 \LWR@inctheorem% new
332     \item[\hskip\labelsep
333 % \theorem@headerfont
334 \InlineClass{theoremheaderremark}{##1.}
335 ]}%
336 \def\@opargbegintheorem##1##2##3{%
337 \BlockClass{theorembodyremark}% new
338 \LWR@inctheorem% new
339     \item[\hskip\labelsep
340 % \theorem@headerfont
341 \InlineClass{theoremheaderremark}{##1\ (##3).}
342 ]}}
343
344 \gdef\th@proof{%
345 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
346 \def\@begintheorem##1##2{%
347 \BlockClass{theorembodyproof}% new
348 \LWR@inctheorem% new
349     \item[\hskip\labelsep
350 % \theorem@headerfont
351 \InlineClass{theoremheaderproof}{##1.}
352 ]}%
353 \def\@opargbegintheorem##1##2##3{%
354 \BlockClass{theorembodyproof}% new
355 \LWR@inctheorem% new
356     \item[\hskip\labelsep
357 % \theorem@headerfont
358 \InlineClass{theoremheaderproof}{##1\ (##3).}
359 ]}}
360
361
362
363 \newcounter{proof}%
364 \if@thmmarks
365 \newcounter{currproofctr}%
366 \newcounter{endproofctr}%
367 \fi
368
369 \gdef\proofSymbol{\openbox}
370

```



```

371 \newcommand{\proofname}{Proof}
372
373 \newenvironment{proof}[1][\proofname]{
374 \th@proof
375 \def\theorem@headerfont{\itshape}%
376 \normalfont
377 \theoremsymbol{\ensuremath{\_\blacksquare}}
378 \@thm{proof}{proof}{#1}
379 }%
380 {\@endtheorem}
381
382 }{}% amsthm option

```

Patched for CSS:

```

383 \let\LWR@origendtheorem\@endtheorem
384 \renewcommand{\@endtheorem}{%
385 \ifbool{LWR@theoremmarks}{%
386 \ifsetendmark%
387 \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
388 \setendmarkfalse%
389 \fi%
390 }{}%
391 \LWR@origendtheorem%
392 \ifbool{LWR@theoremmarks}{\global\setendmarktrue}{}%
393 \endBlockClass%
394 }

395 \gdef\NoEndMark{\global\setendmarkfalse}

```

Redefined to reuse the float mechanism to add list-of-theorem links:

```

\thm@thmline {\langle 1: printed type\rangle} {\langle 2: \#\rangle} {\langle 3: optional\rangle} {\langle 4: page\rangle}

396 \renewcommand{\thm@@thmline@noname}[4]{%
397 \hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
398 }
399
400 \renewcommand{\thm@@thmline@name}[4]{%
401 \hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
402 }

```

This was redefined by ntheorem when loaded, so it is now redefined for lwarp:

```

403 \def\thm@@thmline{\thm@@thmline@name}

```

Patch for CSS:

```

404 \def\listtheorems#1{

```

---

```

405 \LWR@htmlelementclass{nav}{lothm}%
406 \begingroup
407 \c@tocdepth=-2%
408 \def\thm@list{#1}\thm@processlist
409 \endgroup
410 \LWR@htmlelementclassend{nav}{lothm}%
411 }

```

Proof QED symbol:

```

412
413 \newcommand{\qed}{\quad\the\qedsymbol}
414
415 \AtBeginDocument{
416 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
417 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
418 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
419 }

```

`\thref {<label>}`

```

420 \newcommand*{\thref}[1]{\cref{#1}}

```

## Package 64

# lwarp-pagenote.sty

## 131 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

## 132 Parskip

Pkg `parskip` `parskip` is ignored.  
for **HTML output**: Discard all options for `lwarp-parskip`.  
1 `\LWR@ProvidesPackageDrop{parskip}`

## Package 66

# lwarp-placeins.sty

## 133 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

## 134 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

## 135 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

## 136 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 {\BlockClass{singlespace}}
11 {\endBlockClass}
12
13 \newenvironment*{singlespace*}
14 {\BlockClass{singlespace}}
15 {\endBlockClass}
16
17 \newenvironment*{spacing}[1]{
18
19 }{
20
21 }
22
23 \newenvironment*{onehalfspace}
24 {\BlockClass{onehalfspace}}
25 {\endBlockClass}
26
27 \newenvironment*{doublespace}
28 {\BlockClass{doublespace}}
29 {\endBlockClass}

clearpage

```

## Package 70

# lwarp-showidx.sty

## 137 Showidx

Pkg `showidx` `showidx` is ignored.

for **HTML output**: Discard all options for `lwarp-showidx`:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

## Package 71

# lwarp-showkeys.sty

## 138 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

## 139 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

## 140 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 \DeclareDocumentEnvironment{KFLT@marginfloat}{0{-1.2ex} m}
23 {% start
24 \LWR@maybeinthisfloat%
25 % \BlockClass{marginblock}
26 \LWR@stoppars%
27 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}}{}
```

---

```

28 \LWR@startpars%
29 \captionsetup{type=#2}%
30 }
31 {
32 % \endBlockClass
33 \LWR@htmldivclassend{div}
34 }
35
36 \RenewDocumentEnvironment{marginfigure}{o}
37   {\begin{KFLT@marginfloat}{figure}}
38   {\end{KFLT@marginfloat}}
39
40 \RenewDocumentEnvironment{margintable}{o}
41   {\begin{KFLT@marginfloat}{table}}
42   {\end{KFLT@marginfloat}}

```

The following were changed by sidenotes, and now are reset back to their lwarp-supported originals:

Restoring the definition from the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> `article.cls` source:

```

43 \renewenvironment{figure*}
44     {\@dblfloat{figure}}
45     {\end@dblfloat}
46
47 \renewenvironment{table*}
48     {\@dblfloat{table}}
49     {\end@dblfloat}

```

## Package 74

# lwarp-soul.sty

## 141 Soul

(Based on original code by MELCHIOR FRANZ.)

Pkg soul Emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{soul}

Storage for the colors to use:

```
2 \newcommand*\LWR@soululcolor{}\{}
3
4 \newcommand*\LWR@soulstcolor{}\{}
5
6 % \definecolor{LWR@soulhlcolordefault}{HTML}{F8E800}
7 % \newcommand*\LWR@soulhlcolor{LWR@soulhlcolordefault}
8 \newcommand*\LWR@soulhlcolor{}\{}

```

Basic markup with CSS:

```
9 \newcommand{\so}[1]{\InlineClass{letterspacing}{#1}}
10 \newcommand{\caps}[1]{\InlineClass{capsspacing}{#1}}

```

Add colors if not empty:

```
11 \newcommand{\LWR@soulcolor}[4]{%
12 \ifcsempy{#2}%
13 {%
14 \InlineClass{#3}{#1}}%
15 {%
16 \convertcolorspec{named}{\csuse{#2}}{HTML}\LWR@tempcolor%
17 \InlineClass{#3}{#4: \#\LWR@tempcolor}{#1}%
18 }%
19 }
20
21 \newcommand{\ul}[1]{%
22 \LWR@soulcolor{#1}{LWR@soululcolor}{uline}{text-decoration-color}%
23 }
24
25 \newcommand{\st}[1]{
26 \LWR@soulcolor{#1}{LWR@soulstcolor}{sout}{text-decoration-color}%
27 }

```

```

28
29 \newcommand{\hl}[1]{
30 \LWR@soulcolor{#1}{\LWR@soulhlcolor}{highlight}{background-color}%
31 }

```

Nullified:

```

32 \newcommand*\soulaccent}[1]{}
33 \newcommand*\soulregister}[2]{}
34 \newcommand*\sloppyword}[1]{#1}
35 \newcommand*\sodef}[5]{\DeclareRobustCommand*#1[1]{\so{##1}}}
36 \newcommand*\resetso{}
37 \newcommand*\capsdef}[5]{}
38 \newcommand*\capsreset{}
39 \newcommand*\capssave}[1]{}
40 \newcommand*\capsselect}[1]{}
41 \newcommand*\setul}[2]{}
42 \newcommand*\resetul{}
43 \newcommand*\setuldepth}[1]{}
44 \newcommand*\setuloverlap}[1]{}

```

Set colors:

```

45 \newcommand*\setulcolor}[1]{\renewcommand{\LWR@soululcolor}{#1}}
46 \newcommand*\setstcolor}[1]{\renewcommand{\LWR@soulstcolor}{#1}}
47 \newcommand*\sethlcolor}[1]{\renewcommand{\LWR@soulhlcolor}{#1}}

```

Long versions of the user-level macros:

```

48 \let\textso\so
49 \let\textul\ul
50 \let\texthl\hl
51 \let\textcaps\caps

```


## Package 75

# lwarp-subfig.sty

## 142 Subfig

(Based on original code by STEVEN DOUGLAS COCHRAN.)

Pkg subfig subfig is supported and patched by lwarp.

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

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

for HTML output: Accept all options for lwarp-subfig:

```
1 \LWR@ProvidesPackagePass{subfig}

\sf@@@subfloat {<1 type>} [<2 lof entry>] [<3 caption>] {<4 contents>}
```

The outer minipage allows side-by-side subfloats with \hfill between.

```
2 \long\def\sf@@@subfloat#1[#2][#3]#4{%
3 \begin{minipage}{\linewidth}% new
4 \LWR@stoppars% new
5   \ifundefined{FBsc@max}{}%
6     {\FB@readaux{\let\FBsuboheight\relax}}%
7   \@tempcnta=\@ne
8   \if@minipage
9     \@tempcnta=\z@
10  \else\ifdim \lastskip=\z@ \else
11    \@tempcnta=\tw@
12  \fi\fi
13  \ifmaincaptiontop
14    \sf@top=\sf@nearskip
15    \sf@bottom=\sf@farskip
16  \else
17    \sf@top=\sf@farskip
18    \sf@bottom=\sf@nearskip
19  \fi
20  \leavevmode
21  \setbox\@tempboxa \hbox{#4}%
22    \@tempdima=\wd\@tempboxa
```

```

23 \ifundefined{FBsc@max}{}%
24 {\global\advance\Xhsize-\wd\@tempboxa
25 \dimen@=\ht\@tempboxa
26 \advance\dimen@\dp\@tempboxa
27 \ifdim\dimen@>\FBsc@max
28 \global\FBsc@max\dimen@
29 \fi}%
30 \vtop\bgroup
31 \vbox\bgroup
32 \ifcase\@tempcnta
33 \@minipagefalse
34 \or
35 \vskip\sf@top
36 \or
37 \ifdim \lastskip=\z@ \else
38 \@tempskipb\sf@top\relax\@xaddvskip
39 \fi
40 \fi
41 \sf@ifpositiontop{%
42 \ifx \@empty#3\relax \else
43 \sf@subcaption{#1}{#2}{#3}%
44 \vskip\sf@capskip
45 \vskip\sf@captopadj
46 \fi\egroup
47 \hrule width0pt height0pt depth0pt
48 \LWR@startpars% new
49 % \box\@tempboxa
50 #4
51 \LWR@stoppars% new
52 }{%
53 \LWR@startpars% new
54 \ifundefined{FBsc@max}%
55 {
56 % \box\@tempboxa
57 #4
58 }%
59 {\ifx\FBsuboheight\relax
60 % \box\@tempboxa
61 #4
62 \else
63 % \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
64 #4
65 \fi}%
66 \LWR@stoppars% new
67 \egroup
68 \ifx \@empty#3\relax \else
69 \vskip\sf@capskip
70 \hrule width0pt height0pt depth0pt
71 \sf@subcaption{#1}{#2}{#3}%
72 \fi

```

```

73      }%
74      \vskip\sf@bottom
75      \egroup
76      \@ifundefined{FBsc@max}{}%
77      {\addtocounter{FRobj}{-1}%
78      \ifnum\c@FRobj=0\else
79      \subfloatrowsep
80      \fi}%
81      \ifmaincaptiontop\else
82      \global\advance\@nameuse{c@\@capttype}\m@ne
83      \fi
84 \end{minipage}% new
85 \LWR@startpars% new
86 \endgroup\ignorespaces%
87 }%

\sf@subcaption {\langle 1 type\rangle} {\langle 2 lof entry\rangle} {\langle 3 caption\rangle}

88 \long\def\sf@subcaption#1#2#3{%
89 \LWR@stoppars% new
90 \ifx \relax#2\relax \else
91 \bgroup
92 \let\label=\@gobble
93 \let\protect=\string
94 \def\@subcaplabel{%
95 \caption@lstfmt{\@nameuse{p#1}}{\@nameuse{the#1}}}%
96 \sf@updatecaptionlist{#1}{#2}{\the\value{\@capttype}}{\the\value{#1}}%
97 \egroup
98 \fi
99 \bgroup
100 \ifx \relax#3\relax
101 \let\captionlabelsep=\relax
102 \fi
103 % \setbox0\vbox{%
104 % \hb@xt@\the\@tempdima{%
105 %
106 % % \hss
107 % % \parbox[t]{\the\@tempdima}{%
108 % % \caption@make
109 % % {\@nameuse{sub\@capttype name}}%
110 % % {\@nameuse{thesub\@capttype}}%
111 % % {\#3}
112 % % }%
113 % % \hss
114 % % }
115 % %}%
116 \@ifundefined{FBsc@max}%
117 % {\box0}%
118 {
119 % \parbox[t]{\the\@tempdima}{%

```



```

120 \LWR@traceinfo{sfsubcap B1}% new
121 \LWR@htmlblocktag{figcaption}% new
122 \caption@make
123 {\@nameuse{sub\@capttype name}}}%
124 {\@nameuse{thesub\@capttype}}}%
125 {#3}
126 \LWR@htmlblocktag{/figcaption}% new
127 \LWR@traceinfo{sfsubcap B2}% new
128 % }%
129 }%
130     {\dimen@ \ht0%
131     \advance \dimen@ \dp0%
132     \ifdim \dimen@ > \FBsc@max
133     \global \FBsc@max \dimen@
134     \fi
135     \FB@readaux{\let \FBsubcheight \relax}%
136     \ifx \FBsubcheight \relax
137     \def \next{
138 % \parbox[t]{\the \@tempdima}
139 }%
140     \else
141     \def \next{
142 % \parbox[t][\FBsubcheight][t]{\the \@tempdima}
143 }%
144     \fi
145     \vbox{%
146 % \hb@xt@\the \@tempdima{%
147
148 % \hss
149 % \next{%
150 \LWR@traceinfo{sfsubcap C1}% new
151     \caption@make
152     {\@nameuse{sub\@capttype name}}}%
153     {\@nameuse{thesub\@capttype}}}%
154     {#3}
155 \LWR@traceinfo{sfsubcap C1}% new
156 % }%
157 % \hss
158
159 % }
160 }
161 }%
162 \egroup
163 \LWR@startpars% new
164 }

\caption@@@make {\langle caption label \rangle} {\langle caption text \rangle}

165 \renewcommand \caption@@@make [2] {%
166 \LWR@startpars% new

```

```

167 \sbox\@tempboxa{#1}%
168 \ifdim\wd\@tempboxa=\z@
169   \let\caption@lsep\relax
170 \fi
171 \caption@ifempty{#2}{%
172   \let\caption@lsep\@empty
173   \let\caption@tfmt\@firstofone
174 }%
175 %   \@setpar{\@par\caption@@par}\caption@@par
176 \renewcommand{\@par}{\LWR@closeparagraph\LWR@orig@@par}% new
177 \caption@applyfont
178 \caption@fmt
179   {\ifcaption@star\else
180     \begingroup
181       \captionlabelfont
182       #1%
183     \endgroup
184   \fi}%
185   {\ifcaption@star\else
186     \begingroup
187       \caption@iflf\captionlabelfont
188       \relax\caption@lsep
189     \endgroup
190   \fi}%
191   {{\captiontextfont
192     \caption@ifstrut
193     {\vrule\@height\ht\strutbox\@width\z@}%
194     {}}%
195     \nobreak\hskip\z@skip % enable hyphenation
196     \caption@tfmt{#2}
197 \LWR@ensuredoingapar% new
198   \caption@ifstrut
199     {\ifhmode\@finalstrut\strutbox\fi}%
200     {}}%
201   \par}}
202 \LWR@stoppars% new
203 }

```

Patches for \sf@sub@label:

```

204 \def\subfloat@label{%
205 \LWR@ensuredoingapar% new
206 \ifnextchar(%  % match left parenthesis
207   {\sf@sub@label}
208   {\sf@sub@label(Sub\@captype\space
209     \ifundefined{thechapter}{\@nameuse{thechapter}\space}%
210     \@nameuse{p@sub\@captype}%
211     \@nameuse{thesub\@captype}.)}}

```

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

## 143 Tabularx

Pkg `tabularx` `tabularx` is emulated by `lwarp`.

for **HTML output**: Discard all options for `lwarp-tabularx`:

```
1 \LWR@ProvidesPackageDrop{tabularx}

2 \newenvironment{tabularx}[2]
3 {\tabular{#2}}
4 {\endtabular}
5
6 \newenvironment{tabularx*}[2]
7 {\tabular{#2}}
8 {\endtabular}
```

## Package 77

# lwarp-textpos.sty

## 144 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 78

# lwarp-theorem.sty

## 145 Theorem

(Based on original code by FRANK MITTELBAACH.)

Pkg theorem theorem is patched for use by lwarp.

### CSS styling of theorems and proofs:

**Theorem:** <div> of class theorembody<theoremstyle>

**Theorem Header:** <span> of class theoremheader

where <theoremstyle> is plain, break, etc.

for HTML output: 1 \LWR@ProvidesPackagePass{theorem}

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
3 \gdef\theoremstyle#1{%
4   \@ifundefined{th@#1}{\@warning
5     {Unknown theoremstyle ‘#1’. Using ‘plain’}%
6     \theorem@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% new
8   }%
9   {%
10    \theorem@style{#1}%
11    \renewcommand{\LWR@newtheoremstyle}{#1}% new
12  }%
13  \begingroup
14    \csname th@the\theorem@style \endcsname
15  \endgroup}
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```
16 \gdef\xnthm#1#2[#3]{%
17   \expandafter\@ifdefinable\csname #1\endcsname
```

```

18  {%
19  \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
20  \@definecounter{#1}\@newctr{#1}[#3]%
21  \expandafter\xdef\csname the#1\endcsname
22    {\expandafter \noexpand \csname the#3\endcsname
23     \@thmcountersep \@thmcounter{#1}}%
24  \def\@tempa{\global\@namedef{#1}}%
25  \expandafter \@tempa \expandafter{%
26    \csname th@the \theorem@style
27      \expandafter \endcsname \the \theorem@bodyfont
28    \@thm{#1}{#2}}%
29  \global \expandafter \let \csname end#1\endcsname \@endtheorem
30  \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
31  }}
32
33 \gdef\@ynthm#1#2{%
34 \expandafter\@ifdefinable\csname #1\endcsname
35   {
36 \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
37 \@definecounter{#1}%
38   \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
39   \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
40   \expandafter{\csname th@the \theorem@style \expandafter
41     \endcsname \the\theorem@bodyfont \@thm{#1}{#2}}%
42   \global \expandafter \let \csname end#1\endcsname \@endtheorem
43   \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
44   }}
45
46 \gdef\@othm#1[#2]#3{%
47 \expandafter\ifx\csname c@#2\endcsname\relax
48   \@nocounterr{#2}%
49   \else
50   \expandafter\@ifdefinable\csname #1\endcsname
51   {
52 \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
53 \expandafter \xdef \csname the#1\endcsname
54   {\expandafter \noexpand \csname the#2\endcsname}%
55   \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
56   \expandafter{\csname th@the \theorem@style \expandafter
57     \endcsname \the\theorem@bodyfont \@thm{#2}{#3}}%
58   \global \expandafter \let \csname end#1\endcsname \@endtheorem
59   \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
60 }%
61 \fi}

```

The following are patched for CSS.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc. They are gathered together here.

Each theorem is encased in a BlockClass environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader`.

```

62 \gdef\th@plain{%\normalfont\itshape
63   \def\@begintheorem##1##2{%
64     \BlockClass{theorembody\LWR@thisthmstyle}% new
65       \item[\hskip\labelsep
66 \InlineClass{theoremheader}{##1\ ##2}
67 ]}%
68 \def\@opargbegintheorem##1##2##3{%
69 \BlockClass{theorembody\LWR@thisthmstyle}% new
70   \item[\hskip\labelsep
71 \InlineClass{theoremheader}{##1\ ##2\ (##3)}
72 ]}
73 }
74
75 \gdef\th@break{%\normalfont\slshape
76   \def\@begintheorem##1##2{%
77 \BlockClass{theorembody\LWR@thisthmstyle}% new
78 \item[\hskip \labelsep
79 \InlineClass{theoremheader}{##1\ ##2}\newline%
80 ]}%
81 \def\@opargbegintheorem##1##2##3{%
82 \BlockClass{theorembody\LWR@thisthmstyle}% new
83   \item[\hskip \labelsep
84 \InlineClass{theoremheader}{##1\ ##2\ (##3)}\newline
85 ]}
86 }
87
88 \gdef\th@marginbreak{%\normalfont\slshape
89   \def\@begintheorem##1##2{
90 \BlockClass{theorembody\LWR@thisthmstyle}% new
91 \item[\hskip\labelsep %
92 \InlineClass{theoremheader}{##2 \quad ##1}\newline
93 ]}%
94 \def\@opargbegintheorem##1##2##3{%
95 \BlockClass{theorembody\LWR@thisthmstyle}% new
96 \item[\hskip\labelsep %
97 \InlineClass{theoremheader}{##2 \quad ##1\ %
98 (##3)}\newline
99 ]}
100 }
101
102 \gdef\th@changebreak{%\normalfont\slshape
103   \def\@begintheorem##1##2{
104 \BlockClass{theorembody\LWR@thisthmstyle}% new
105 \item[\hskip\labelsep
106 \InlineClass{theoremheader}{##2\ ##1}\newline
107 ]}%

```



```

108 \def\@opargbegintheorem##1##2##3{%
109 \BlockClass{theorembody\LWR@thisthmstyle}% new
110 \item[\hskip\labelsep
111 \InlineClass{theoremheader}{ ##2\ ##1\ %
112 (##3)}\newline
113 ]}
114 }
115
116 \gdef\th@change{%\normalfont\slshape
117 \def\@begintheorem##1##2{
118 \BlockClass{theorembody\LWR@thisthmstyle}% new
119 \item[\hskip\labelsep
120 \InlineClass{theoremheader}{##2\ ##1}
121 ]}%
122 \def\@opargbegintheorem##1##2##3{%
123 \BlockClass{theorembody\LWR@thisthmstyle}% new
124 \item[\hskip\labelsep
125 \InlineClass{theoremheader}{##2\ ##1\ (##3)}
126 ]}
127 }
128
129 \gdef\th@margin{%\normalfont\slshape
130 \def\@begintheorem##1##2{
131 \BlockClass{theorembody\LWR@thisthmstyle}% new
132 \item[\hskip\labelsep
133 \InlineClass{theoremheader}{##2 \quad ##1}
134 ]}%
135 \def\@opargbegintheorem##1##2##3{%
136 \BlockClass{theorembody\LWR@thisthmstyle}% new
137 \item[\hskip\labelsep
138 \InlineClass{theoremheader}{##2 \quad ##1\ (##3)}
139 ]}
140 }

```

Patched for CSS:

```

141 \gdef\@endtheorem{\endBlockClass\endtrivlist}

```

## Package 79

# lwarp-threeparttable.sty

## 146 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 \BlockClass{tnotes}%
```

```
7 \setlist[description]{format=\LWR@printtablenote}%
```

```
8 \description%
```

```
9 }
```

```
10 {%
```

```
11 \enddescription%
```

```
12 \endBlockClass%
```

```
13 }
```

```
14 \newcommand{\tnote}[1]{\textsuperscript{#1}}
```

## Package 80

# lwarp-tikz.sty

## 147 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 81

# lwarp-titles.sty

## 148 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 m m m m}{}
25 \NewDocumentCommand{\setfloatfoot}{s o o m m m m}{}

26 \NewDocumentCommand{\nextfloathead}{s o o m m m m}{}
27 \NewDocumentCommand{\nextfloatfoot}{s o o m m m m}{}

28 \newcommand{\newmarkset}[1]{}

29 \NewDocumentCommand{\newextramarkset}{s m 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 82

# lwarp-titlesec.sty

## 149 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 83

# lwarp-titletoc.sty

## 150 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 84

# lwarp-titling.sty

## 151 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 85

# lwarp-tocloft.sty

## 152 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*{\cftpafterpnum}{}

38 \newlength{\cftbeforechapskip}
39 \newlength{\cftchapindent}
40 \newlength{\cftchapnumwidth}
41 \newcommand*{\cftchapfont}{}
42 \newcommand*{\cftchappresnum}{}
43 \newcommand*{\cftchapaftersnum}{}
44 \newcommand*{\cftchapaftersnumb}{}
45 \newcommand*{\cftchapleader}{}
46 \newcommand*{\cftchapdotsep}{1}
47 \newcommand*{\cftchapfont}{}
48 \newcommand*{\cftchapafterpnum}{}

49 \newlength{\cftbeforesecskip}
50 \newlength{\cftsecindent}
51 \newlength{\cftsecnumwidth}
52 \newcommand*{\cftsecfont}{}
53 \newcommand*{\cftsecpresnum}{}
54 \newcommand*{\cftsecaftersnum}{}
55 \newcommand*{\cftsecaftersnumb}{}
56 \newcommand*{\cftsecleader}{}
57 \newcommand*{\cftsecdotsep}{1}
58 \newcommand*{\cftsecfont}{}
59 \newcommand*{\cftsecafterpnum}{}

60 \newlength{\cftbeforesubsecskip}
61 \newlength{\cftsubsecindent}
62 \newlength{\cftsubsecnumwidth}
63 \newcommand*{\cftsubsecfont}{}
64 \newcommand*{\cftsubsecpresnum}{}
65 \newcommand*{\cftsubsecaftersnum}{}
66 \newcommand*{\cftsubsecaftersnumb}{}
67 \newcommand*{\cftsubsecleader}{}
68 \newcommand*{\cftsubsecdotsep}{1}
```

```
69 \newcommand*{\cftsubsecpagefont}{}
70 \newcommand*{\cftsubsecafterpnum}{}

71 \newlength{\cftbeforesubsubsecskip}
72 \newlength{\cftsubsubsecindent}
73 \newlength{\cftsubsubsecnumwidth}
74 \newcommand*{\cftsubsubsecfont}{}
75 \newcommand*{\cftsubsubsecpresnum}{}
76 \newcommand*{\cftsubsubsecaftersnum}{}
77 \newcommand*{\cftsubsubsecaftersnumb}{}
78 \newcommand*{\cftsubsubsecleader}{}
79 \newcommand*{\cftsubsubsecdotsep}{1}
80 \newcommand*{\cftsubsubsecpagefont}{}
81 \newcommand*{\cftsubsubsecafterpnum}{}

82 \newlength{\cftbeforeparaskip}
83 \newlength{\cftparaindent}
84 \newlength{\cftparanumwidth}
85 \newcommand*{\cftparafont}{}
86 \newcommand*{\cftparapresnum}{}
87 \newcommand*{\cftparaftersnum}{}
88 \newcommand*{\cftparaftersnumb}{}
89 \newcommand*{\cftparaleader}{}
90 \newcommand*{\cftparadotsep}{1}
91 \newcommand*{\cftparapagefont}{}
92 \newcommand*{\cftparaafterpnum}{}

93 \newlength{\cftbeforesubparaskip}
94 \newlength{\cftsubparaindent}
95 \newlength{\cftsubparanumwidth}
96 \newcommand*{\cftsubparafont}{}
97 \newcommand*{\cftsubparapresnum}{}
98 \newcommand*{\cftsubparaftersnum}{}
99 \newcommand*{\cftsubparaftersnumb}{}
100 \newcommand*{\cftsubparaleader}{}
101 \newcommand*{\cftsubparadotsep}{1}
102 \newcommand*{\cftsubparapagefont}{}
103 \newcommand*{\cftsubparaafterpnum}{}

104 \newlength{\cftbeforefigskip}
105 \newlength{\cftfigindent}
106 \newlength{\cftfignumwidth}
107 \newcommand*{\cftfigfont}{}
108 \newcommand*{\cftfigpresnum}{}
109 \newcommand*{\cftfigaftersnum}{}
110 \newcommand*{\cftfigaftersnumb}{}
111 \newcommand*{\cftfigleader}{}
112 \newcommand*{\cftfigdotsep}{1}
113 \newcommand*{\cftfigpagefont}{}

```

```

114 \newcommand*\cftfigafterpnum){}

115 \newlength{\cftbeforesubfigskip}
116 \newlength{\cftsubfigindent}
117 \newlength{\cftsubfignumwidth}
118 \newcommand*\cftsubfigfont){}
119 \newcommand*\cftsubfigpresnum){}
120 \newcommand*\cftsubfigaftersnum){}
121 \newcommand*\cftsubfigaftersnumb){}
122 \newcommand*\cftsubfigleader){}
123 \newcommand*\cftsubfigdotsep}{1}
124 \newcommand*\cftsubfigpagefont){}
125 \newcommand*\cftsubfigafterpnum){}

126 \newlength{\cftbeforetabskip}
127 \newlength{\cfttabindent}
128 \newlength{\cfttabnumwidth}
129 \newcommand*\cfttabfont){}
130 \newcommand*\cfttabpresnum){}
131 \newcommand*\cfttabaftersnum){}
132 \newcommand*\cfttabaftersnumb){}
133 \newcommand*\cfttableader){}
134 \newcommand*\cfttabdotsep}{1}
135 \newcommand*\cfttabpagefont){}
136 \newcommand*\cfttabafterpnum){}

137 \newlength{\cftbeforesubtabskip}
138 \newlength{\cftsubtabindent}
139 \newlength{\cftsubtabnumwidth}
140 \newcommand*\cftsubtabfont){}
141 \newcommand*\cftsubtabpresnum){}
142 \newcommand*\cftsubtabaftersnum){}
143 \newcommand*\cftsubtabaftersnumb){}
144 \newcommand*\cftsubtableader){}
145 \newcommand*\cftsubtabdotsep}{1}
146 \newcommand*\cftsubtabpagefont){}
147 \newcommand*\cftsubtabafterpnum){}

148 \newcommand{\cftsetindents}[3]{}

149 \newcommand{\pagenumbersoff}[1]{}
150 \newcommand{\pagenumberson}[1]{}

```

Emulated through the \newfloat mechanism.

```

151 \NewDocumentCommand{\newlistof}{o m m}
152 {%
153 \IfValueTF{#1}

```

```

154 {\newfloat{#2}{tbp}{#3}[#1]}
155 {\newfloat{#2}{tbp}{#3}}
156 \@namedef{listof#2}{\listof{#2}{#4}}
157 \@namedef{#2depth}{1}
158 \expandafter\newlength\csuse{cftbefore#2skip}
159 \expandafter\newlength\csuse{cft#2indent}
160 \expandafter\newlength\csuse{cft#2numwidth}
161 \@namedef{cft#2font}{}
162 \@namedef{cft#2presnum}{}
163 \@namedef{cft#2aftersnum}{}
164 \@namedef{cft#2aftersnumb}{}
165 \@namedef{cft#2leader}{}
166 \@namedef{cft#2dotsep}{1}
167 \@namedef{cft#2pagefont}{}
168 \@namedef{cft#2afterpnum}{}
169 }

```

\cftchapterprecis from tocloft:

```

170 \newcommand{\cftchapterprecis}[1]{%
171   \cftchapterprecishere{#1}
172   \cftchapterprecistoc{#1}}
173 \newcommand{\cftchapterprecishere}[1]{%
174   \begin{quote}\textit{#1}\end{quote}}
175 \newcommand{\cftchapterprecistoc}[1]{
176   \addtocontents{toc}{%
177     {
178       \protect\begin{quote}#1\protect\end{quote}}
179   }
180 }

```

## Package 86

# lwarp-trivfloat.sty

## 153 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}`

### 153.1 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{tflfloatcnt}{1} % start trivfloats with .lob
\makeatletter
```

---

```
9 \end{warpall}
```

## Package 87

# lwarp-ulem.sty

## 154 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}}
```

`\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 88

# lwarp-verse.sty

## 155 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` 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@atbeginverbatim{verse}
10 \unskip\vspace{-\baselineskip}
11 }
```

After the end of the `verse` environment, which places the `pre` tag at the regular left margin:

```

12 \AfterEndEnvironment{verse}{
13 \unskip\vspace{-\baselineskip}
14 \LWR@afterendverbatim
15 }
```

Patch to place poemtitle inside an HTML span of class `poemtitle`:

```

16 \ifdef{\poemtitle}{
17 \DeclareDocumentCommand{\@vstypeptitle}{m}{%
18   \vspace{\beforepoemtitleskip}%
19   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
20   \vspace{\afterpoemtitleskip}%
21 }
22 }{}
23
24 }
```

## Package 89

# lwarp-wallpaper.sty

## 156 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 90

# lwarp-wrapfig.sty

## 157 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 % \BlockClass{marginblock}{%
19 % width:\rndprintlength{\LWR@wrapwidth} ; %
20 % \LWR@wrapposition%
21 % ]%
22 \LWR@stoppars%
23 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{\LWR@thisfloat}"
24 style="width:\rndprintlength{\LWR@wrapwidth} ; %
25 \LWR@wrapposition"%
26 }
27 \LWR@startpars
28 }
29
30
31 \NewDocumentEnvironment{wrapfigure}{o m o m}
32 {%
33 \LWR@subwrapfigure{#2}{#4}%
34 \captionsetup{type=figure}%
```

```
35 }
36 {
37 \LWR@htmldivclassend{div}
38 }
39
40
41 \NewDocumentEnvironment{wraptable}{o m o m}
42 {%
43 \LWR@subwrapfigure{#2}{#4}%
44 \captionsetup{type=table}%
45 }
46 {
47 \LWR@htmldivclassend{div}
48 }
49
50
51 \NewDocumentEnvironment{wrapfloat}{m o m o m}
52 {%
53 \LWR@subwrapfigure{#3}{#5}%
54 \captionsetup{type=#1}%
55 }
56 {
57 \LWR@htmldivclassend{div}
58 }
59
60 \newlength{\wrapoverhang}
```

## Package 91

# lwarp-xcolor.sty

## 158 Xcolor

Pkg	xcolor	xcolor is supported by lwarp.
support		Color definitions, models, and mixing are fully supported without any changes required.
tables		Colored tables are ignored so far. Use CSS to style tables.
colored text and boxes		<code>\textcolor</code> , <code>\colorbox</code> , and <code>\fcolorbox</code> are supported.
<code>\color</code> and <code>\pagecolor</code>		<code>\color</code> and <code>\pagecolor</code> are ignored. Use CSS or <code>\textcolor</code> where possible.
for HTML output:		<pre> 1 \LWR@ProvidesPackagePass{xcolor}  2 \newcommand*{\LWR@tempcolor}{}  defaulting to black.  3 \newcommand*{\LWR@currenttextcolor}{black}  \LWR@colorstyle {\langle 1: styletext \rangle} {\langle 2: model \rangle} {\langle 3: color \rangle} {\langle 4: spancontents \rangle}  Creates a styled span with a color converted to HTML hex colorspace. Uses LWR@spandepth to prevent paragraph tags inside the span. If used for \textcolor, with a styletext of color:, then the new color is copied into \LWR@currenttextcolor for possible re-use in \rule.  4 \NewDocumentCommand{\LWR@colorstyle}{m m m m}{%  Use the xcolor package to convert to an HTML color space:  5 \convertcolorspec{#2}{#3}{HTML}\LWR@tempcolor%  If is a \textcolor, save a copy of this color for use by \rule:  6 \ifthenelse{\equal{#1}{color:}}{% 7 {\renewcommand*{\LWR@currenttextcolor}{\#\LWR@tempcolor}}{}}%  Create the HTML &lt;span&gt; with the styled color: </pre>

```

8 \LWR@htmltagc{span style="#1\#\LWR@tempcolor"{} }%
9 \begin{LWR@nestspan}%

```

Prevent additional paragraph tags inside this span:

Print the contents then close the span:

```

10 #4%
11 \LWR@htmltagc{/span}%
12 \end{LWR@nestspan}%

```

For paragraph-tag handling:

```

13 \LWR@ensuredoingapar%
14 }

```

`\color` appears in the L<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 `\NewCSS` instead.

```

20 \newcommand*{\LWR@pagecolor}[2][named]{ }

```

`\colorbox` [*model*] {*color*} {*text*} is converted into an HTML hex background color span.

```

21 \NewDocumentCommand{\LWR@colorbox}{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 92

# lwarp-xfrac.sty

## 159 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. . . . .	57
General: 2016/03/08 Initial version	1		
v0.11		Docs: Commands into a warpprint environment. . . . .	56
General: 2016/03/11 . . . . .	1	Docs: Newclude limitations. . .	68
Added section:		Docs: Table: Cross-referencing data structures. . . . .	193
Operating-System portability.	92	Docs: Table: Float data structures. . . . .	203
Added section: Selecting the operating system. . . . .	55	Docs: Trademarks section. . .	271
Test Suite: limages and index in README.txt . . . . .	1	Docs: Troubleshooting cross-references. . . . .	71
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. . . . .	123	v0.15	
General: 2016/03/14 . . . . .	1	\printpublished: No div if empty. . . . .	144
Global: Uses \p@(type) in float captions. . . . .	1	\printsubtitle: No div if empty. . . . .	145
Test Suite: Sub-figures . . . . .	1	General: 2016/04/06 . . . . .	1
v0.13		Added . . . . .	350
\CaptionSeparator: Fix for newer babel package. . . . .	205	Ampersand (&): Fixed handling when passed as an argument. . . . .	168
\LWR@LwarpStart: \up and \fup . . . . .	135	Docs: Added warning icons for items needing special attention. . . . .	76
\published: Default to empty published. . . . .	141	Docs: Clarify print/HTML output. . . . .	55
\subtitle: Default to empty subtitle. . . . .	141	Docs: Moved the supported functions table to the introduction. . . . .	23
General: 2016/03/24 . . . . .	1	Files: lwarp_formal.css added. . . . .	1
Removed package: subfig . . . . .	1	Fix: steps counter . . . . .	349
Test Suite: Ordinals, Subcaption . . . . .	1	Fixed & handling. . . . .	347
tikzpicture: Fix dollar-redefined bug for newer package. . . . .	427	Test Suite: test_suite_formal.css file added. . . . .	1
v0.14		v0.16	
\LWR@htmlsectionfilename: Fix: Links to home page. . . . .	99	General: 2016/04/11 . . . . .	1
General: 2016/03/31 . . . . .	1		
floatrow: Added. . . . .	345		

\titlingpage: Improved print-output spacing. . . . .	146	\LWR@subhyperrefclass: Improved HTML output linebreaks. . .	200
Added XeLaTeX, LuaLaTeX support. . . . .	83	\LWR@subinlineimage: Surpress extra space. . . . .	201
Docs: Font and UTF-8 support. . . . .	54	General: 2016/05/19 . . . . .	1
Docs: Moved location of \usepackage{lwarp}. . . . .	55	File: lwarp.css: Improved TOC outline display. . . . .	1
Docs: Text not converting. . . . .	71	Files: lwarp.css and lwarp_formal.css: Improved responsive design. . . . .	1
Fix: amsmath options clash . . . . .	86	Microtype disabled during HTML generation . . . . .	84
Fix: newtxmath compatibility. . . . .	86	PDF Unicode input characters. . . . .	77
Lwarp no longer selects fonts. . . . .	54, 83	Test Suite: Verse package . . . . .	1
Removed package: suffix . . . . .	1	lateximage: pdfcrop: --hires added. . . . .	234
Test Suite: Improved titlingpage. . . . .	146	Reorganize \HomeHTMLFilename logic. . . . .	234
Test Suite: Lwarp no longer selects fonts. . . . .	1	Surpress extra space. . . . .	234
Test Suite: Supports XeLaTeX, LuaLaTeX. . . . .	1	verse: Supports verse, memoir packages. . . . .	442
xfrac: Adjusted for the use of any font: . . . . .	451	minipage: Fix: \linewidth, \textwidth, \textheight inside a minipage. . . . .	253
v0.17		v0.19	
\LWR@htmlsectionfilename: Fix: Links when entire doc is one HTML page. . . . .	99	\HTMLFileName: Docs: Escape filename underscores. . . . .	98
General: 2016/04/14 . . . . .	1	\HomeHTMLFileName: Docs: Escape filename underscores. . . . .	98
mdframed: Added. . . . .	377	\LWR@LwarpStart: Enabled \\ equal to \newline. . . . .	134
Test Suite: Fix: Print-version front-matter page numbers. . . . .	1	\LWR@doubledollar: MathJax support. . . . .	220
Test Suite: Mdframed . . . . .	1	\LWR@filestart: lwarp_mathjax.txt loaded. . . . .	132
v0.18		\LWR@hspace: Fix: \hspace length computations. . . . .	262
\LWR@hspace: \hspace supported. . . . .	262	\LWR@minipagestartpars: Surpresses paragraph tags between minipages. . . . .	261
\LWR@includegraphicsb: Add: svgz file extension. . . . .	243	\LWR@singledollar: MathJax support. . . . .	220
em, ex, %, px dimensions preserved. . . . .	243	\LWR@tabledatamultirowtag: Added optional args. . . . .	188
Fix: \linewidth, \textwidth, \textheight inside a minipage. . . . .	243	\LateximageFontSizeName: Add: User-adjustable math/lateximage font size. . . . .	232
Improved HTML output linebreaks. . . . .	243	\minipagefullwidth: Added: No width tag for the next minipage in HTML. . . . .	253
\LWR@myshorttoc: Reorganize \HomeHTMLFilename logic. . . . .	209		
\LWR@newhtmlfile: Sidetoc after title, improving responsive design. . . . .	122		
\LWR@requesttoc: Reorganize \HomeHTMLFilename logic. . . . .	137		
\LWR@subhyperref: Improved HTML output linebreaks. . . . .	200		

<code>\rule</code> : Added	266	<code>\LWR@htmldivclass</code> : Added	
<code>\warpHTMLonly</code> : Added	78	optional style	106
<code>\warpprintonly</code> : Replaces		<code>\LWR@htmlclass</code> : Added	
<code>\rowprintedonly</code>	78	optional style	105
General: 2016/06/08	1	<code>\LWR@htmlsectionfilename</code> :	
Added	450	HTMLFileName: removed	
Avoids MathJax	219	additional trailing ‘-’, and may	
cleveref: Loaded		be empty	99
<code>\AtEndPreamble</code>	249	Sections called “Index” or “index”	
CSS for table note item	426	have an underscore prepended	
Docs: <code>multirow</code> browser bug		to their filenames if no prefix	99
workaround	64, 165	<code>\LWR@includegraphicsb</code> : Fix:	
Docs: Math options	55	<code>\linewidth</code> in a floatrow	243
Docs: Table: Cross-referencing		Fix: Expands filename	243
data structures, updated	193	<code>\LWR@longtabledatacaptiontag</code> :	
File: <code>lwarp_mathjax.txt</code> added	1	Fix: Pars in captions	185
File: <code>lwarp.css</code> :		<code>\LWR@section</code> : Combined	
<code>tnoteitemheader</code> added	1	higher-level sections together	
Introduction: MathJax support		into files	126
mentioned	22	<code>\LWR@subhtmlclass</code> :	
MathJax support		Factored code	105
added	221, 225, 227	<code>\SetHTMLFileNumber</code> : Add:	
Options: <code>mathsvg</code> and <code>mathjax</code>	79	Control file numbers	98
Supports colored <code>\rule</code>	447	<code>\cpagerefFor</code> : User-redefinable	
titles: null <code>\pagestyle</code> and		word for page references	250
<code>\thispagestyle</code> for HTML	87	<code>\dotfill</code> : Inserts an ellipsis	259
<code>tikzpicture</code> : Adapts to <code>tikz</code>		<code>\hfill</code> : Inserts a <code>\qqquad</code>	259
version	427	<code>\hrulefill</code> : Inserts a short rule	259
<code>equation</code> : MathJax support	224	<code>\hyperindexref</code> : Print mode	
v0.20		provided in case <code>hyperref</code> not	
<code>\BlockClassSingle</code> : Renamed		used	216
from “ <code>LWR@htmldivclassline</code> ”	107	<code>\pageref</code> : Added	199
<code>\HTMLFileName</code> : No longer escape		<code>\tracinglwarp</code> : Added	91
underscores	98	<code>\warpOSwindows</code> : Auto-detects	
<code>\HomeHTMLFileName</code> : No longer		operating system	93
escape underscores	98	General: 2017/02/09	1
<code>\InlineClass</code> : Renamed from		<code>afterpage</code> : Added	323
“ <code>inlineclass</code> ”	108	<code>alltt</code> : Added	325
<code>\LWR@LwarpStart</code> : Fix: math cross		bookmark: Added	328
references	135	caption and subcaption	
<code>\LWR@closeparagraph</code> : <code>\unskip</code>		supported	1
extra spaces	111	cleveref and referencing patches:	
No break tags in the start/end of		Applied <code>\AfterEndPreamble</code>	249
a tabular	111	<code>draftwatermark</code> : Added	334
<code>\LWR@endoffline</code> : Fix: <code>\\</code>	261	<code>eso-pic</code> : Added	339
<code>\LWR@filestart</code> : Adds meta		<code>everypage</code> : Added	340
description	132	<code>extramarks</code> : Added	340
<code>\LWR@hspace</code> : Add: Supports		<code>fancyhdr</code> : Added	341
HTML thin breakable space	262	<code>hyperref</code> : Additional user	
		macros	360

keyfloat: Added. . . . .	364	Verbatim: Added. . . . .	157
letterspace: User-interface		verbatim: Added. . . . .	154
emulated. . . . .	365	BlockClass: Added optional style. . . . .	107
listings: Added. . . . .	368	Renamed from "blockclass". . . . .	107
lrcaption: Added. . . . .	374	LWR@nestspan: Fix: Minipages	
lwarp-newproject: Added. . . . .	272	inside a span. . . . .	103
microtype: User-interface		v0.21	
emulated. . . . .	384	\LWR@LwarpStart: Changed	
needspace: Added. . . . .	389	lateximages to a .txt file. . . . .	134
nowidow: Added. . . . .	391	\LWR@filestart: Skip title if not	
placeins: Added. . . . .	404	given. . . . .	132
ragged2e: Added. . . . .	405	\LWR@newhtmlfile: Skip title if not	
setspace: Improved support. . . . .	407	given. . . . .	122
textpos: Added. . . . .	421	\marginpar: Fixed source listing. . . . .	119
titles: Added. . . . .	428	General: 2017/02/23 . . . . .	1
titlesec: Added. . . . .	430	fontenc: Added. . . . .	351
titletoc: Added. . . . .	431	fontspec: Added. . . . .	351
titling: Improved compatibility. . . . .	432	inputenc: Added. . . . .	363
tocloft: Added. . . . .	433	newclue: Added. . . . .	389
wallpaper: Added. . . . .	444	newunicodechar: Added. . . . .	390
wrapfig: Added. . . . .	445	Add: Errors for misplaced	
Added @, <, > columns. . . . .	165	packages. . . . .	80
Added \NewHTMLdescription. . . . .	116	Docs: Added internet class. . . . .	28
Added single-expansion data		Docs: Added TeX2page,	
arrays. . . . .	97	GladTeX. . . . .	28
Code factored into independent		Docs: Installing on Windows. . . . .	34
lwarp_html files. . . . .	320	File lwarp_tutorial.txt added. . . . .	38
Docs: Examples for generating		lwarpmk: Fix: lwarpmk again	
HTML file names. . . . .	48	for Windows. . . . .	311
Docs: Improved index. . . . .	1	lwarpmk: Fix: lwarpmk limages	
Enhanced titling support. . . . .	143	for Windows. . . . .	311
File: lwarp.css: Minor fixes for		lwarpmk: Fix: lwarpmk uses	
validation. . . . .	1	lateximages text file instead of	
File: lwarpmk used to compile		shell script. . . . .	311
print, HTML, indexes, and		v0.22	
lateximages. . . . .	1	\LWR@parseDcolumn: Added	
Fix: \linewidth in a floatrow. . . . .	348	tabular D column. . . . .	174
Improved float caption type		\LWR@parsebangcolumn: Added	
handling. . . . .	342	tabular ! column. . . . .	171
Moved sidebar and example code		\LWR@parsetablecols: Unknown	
to test suite. . . . .	1	table column types become l.	
Page geometry set to 6in wide		Added tabular D, !, X columns. . . . .	175
with large margins. . . . .	84	\LWR@printmccoldata: Added	
Parallel versions of aux files for		tabular D, !, and X columns. . . . .	182
print/HTML. . . . .	1	General: 2017/03/02 . . . . .	1
Removed reliance on make, grep,		abstract: Added. . . . .	321
gawk. . . . .	1	changepage: Added. . . . .	331
Tabular: \unskip extra spaces. . . . .	165	dcolumn: Added. . . . .	333
Test Suite: HTML meta		enumerate: Added. . . . .	337
descriptions. . . . .	1	ftnright: Added. . . . .	358



geometry: Nullified commands.	358	lips: Added.	367
indentfirst: Added.	363	mdframed: Help avoid	
layout: Added.	365	hyphenation.	378
lscap: Added.	374	ntheorem: Added.	392
mcaption: Added.	376	showidx: Added.	408
nameref: Added.	388	theorem: Added.	422
nextpage: Added.	390	Basic L <sup>A</sup> T <sub>E</sub> X theorems: improved	
parskip: Added.	404	CSS.	160
showkeys: Added.	408	Docs: Adds credits for patched	
sidecap: Added.	409	code.	1
tabularx: Added.	420	Docs: Testing <code>lwarp</code> .	70
varioref: Supported.	67	Fix: Allows XeL <sup>A</sup> T <sub>E</sub> X and	
verse: Added.	442	LuaL <sup>A</sup> T <sub>E</sub> X to preload graphics	
		and graphicx.	81
v0.23		v0.26	
\LWR@parsetablecols: Fix for vert		General: 2017/03/31	1
bar column type.	175	cutwin: Added.	332
\LWR@printmccoldata: Fix for vert		endnotes: Added.	336
bar column type.	182	floatfit: Added.	344
General: 2017/03/02	1	footmisc: Added.	351
v0.24		footnotehyper: Added.	354
\LWR@hspace: Add: <code>\hspace \fill</code>		footnote: Added.	353
converts to 2em	262	marginfix: Added.	375
\LWR@htmlfileref: Fix: Index		marginnote: Added.	376
links while <code>\tracinglwarp</code> .	195	mparhack: Added.	385
\hypertocfloat: List of floats		pagenote: Supported as-is.	403
responds to <code>lofdepth</code> ,		sidenotes: Added.	410
<code>lotdepth</code> .	214	<code>lwarp.css</code> : Improved responsive	
General: 2017/03/15	1	<code>marginpar</code> and <code>marginblock</code> .	275
floatrow: Support for <code>subfig</code> .	345	Docs: Improved MiK <sub>T</sub> E <sub>X</sub> install	
subfig: Added.	414	instructions.	31, 34
tikz: For tikz v3.0.0 or later,		Dollar span avoided in a	
auto-loads tikz babel library if		<code>lateximage</code> .	219
necessary.	427	Footnotes now are L <sup>A</sup> T <sub>E</sub> X boxes	
Docs: Filename underscore.	47, 57	instead of pagenotes.	117
No longer preloads <code>subcaption</code> ;		<code>lateximage</code> : Labels track page	
conflicted with <code>subfig</code> .	86	numbers of <code>lateximages</code> .	234
picture: Fix for inline images.	250	Print mode now uses a <code>minipage</code>	
tikzpicture: Fix for inline		of <code>\linewidth</code> .	234
images.	427	picture: Fix for <code>\makebox</code> in	
v0.25		picture.	250
\LWR@loadnever: Added the ability		v0.27	
to prevent conflicting packages.	81	General: 2017/04/04	1
\addcontentsline: Handles		lettrine: Added.	366
theorems.	208	microtype: Fix with XeL <sup>A</sup> T <sub>E</sub> X,	
General: 2016/03/22	1	LuaL <sup>A</sup> T <sub>E</sub> X.	384
amsthm: Added.	326	soul: Added.	412
ccaption: Prevented.	330	ulem: Added.	440
ellipsis: Added.	334	Docs: Installing utilities for	
emptypage: Added.	335	MacOS.	36
framed: Added.	355		

Docs: Limitations of saveboxes. . . . .	65	Page geometry modified to	
Fix for table footnote par tags. . . . .	118	reduce line overflow. . . . .	84

## 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		
$\backslash \$$ . . . . .	<i>219</i>	algorithmic
$\backslash \&$ . . . . .	<i>97</i> , <u>2131</u>	with newfloat, trivfloat . . . . .
$\backslash ($ . . . . .	<u>3128</u>	algorithmicx (package) . . . . .
$\backslash )$ . . . . .	<u>3128</u>	align (environment) . . . . .
$\backslash @\setcpageref$ . . . . .	<u>3833</u>	align* (environment) . . . . .
$\backslash @\setcref$ . . . . .	<u>3829</u>	alltt (package) . . . . .
$\backslash @\setcrefrange$ . . . . .	<u>3830</u>	$\backslash \text{AmS}$ . . . . .
@author . . . . .	<i>139</i>	amsmath (package) . . . . .
$\backslash @begintheorem$ . . . . .	<u>1990</u>	amsthm (package) . . . . .
$\backslash @capytype$ . . . . .	<u>2849</u>	$\backslash and$ . . . . .
$\backslash @currentlabelname$ . . . . .	<u>2642</u>	array (package) . . . . .
@date . . . . .	<i>139</i>	AsciiDoc (program) . . . . .
$\backslash @dlbfloat$ . . . . .	<u>2832</u>	AsciiDoctor (program) . . . . .
$\backslash @endtheorem$ . . . . .	<u>1998</u>	Asciidoctor-LaTeX (program) . . . . .
$\backslash @float$ . . . . .	<u>2832</u>	$\backslash attrib$ . . . . .
$\backslash @fnsymbol$ . . . . .	<u>1724</u>	$\backslash attribution$ . . . . .
$\backslash @makecaption$ . . . . .	<u>2851</u>	$\backslash author$ . . . . .
$\backslash @maketitle$ . . . . .	<u>1735</u>	autosec . . . . .
$\backslash @nameedef$ . . . . .	<u>69</u>	
$\backslash @opargbegintheorem$ . . . . .	<u>1994</u>	
@published . . . . .	<i>139</i>	
@subtitle . . . . .	<i>139</i>	
@title . . . . .	<i>139</i>	
$\backslash \backslash$ . . . . .	<i>261</i>	
$\$$ . . . . .	<i>220</i>	
$\$ \$$ . . . . .	<i>220</i>	
A		B
abstract (environment) . . . . .	<u>1764</u>	babel (package) . . . . .
abstract (package) . . . . .	<i>321</i>	$\backslash backmatter$ . . . . .
$\backslash abstractname$ . . . . .	<i>49</i> , <u>1763</u>	BaseJobName (option) [lwrap-
accents . . . . .	<i>124</i>	newproject] . . . . .
$\backslash addcontentsline$ . . . . .	<u>2903</u>	$\backslash BaseJobname$ . . . . .
Adobe (program) . . . . .	<i>30</i>	BaseJobname (option) [lwrap] . . . . .
$\backslash affiliation$ . . . . .	<u>1433</u>	baseline
afterpage (package) . . . . .	<i>323</i>	tabular . . . . .
		$\backslash bfseries$ . . . . .
		$\backslash BibTeX$ . . . . .
		BlockClass (environment) . . . . .
		$\backslash BlockClassSingle$ . . . . .
		bookmark (package) . . . . .
		booktabs (package) . . . . .
		booleans:
		CombineHigherDepths . . . . .
		FileSectionNames . . . . .
		LWR@amsmultline . . . . .



endnotes (package) . . . . .	336	eso-pic (package) . . . . .	339
\enlargethispage . . . . .	4110	etoolbox (package) . . . . .	77
\enskip . . . . .	261, 4074	everyhook (package) . . . . .	85
enumerate (environment) . . . . .	2029	everypage (package) . . . . .	340
enumerate (package) . . . . .	69, 337	expl3 (package) . . . . .	85
enumitem (package) . . . . .	69, 86	extramarks (package) . . . . .	340
environ (package) . . . . .	87		
environments:		<b>F</b>	
abstract . . . . .	1764	fancyhdr (package) . . . . .	341
align . . . . .	3344	fancyvrb (package) . . . . .	85
align* . . . . .	3366	FAQ . . . . .	71
BlockClass . . . . .	721	fbox	
BVerbatim . . . . .	1976	around a minipage . . . . .	58
center . . . . .	3564	\fbbox . . . . .	3971
description . . . . .	2060	file	
displaymath . . . . .	3137	inaccessible . . . . .	47, 48
enumerate . . . . .	2029	underscore . . . . .	47, 57
equation . . . . .	3204	FileDepth (counter) . . . . .	47, 119
flalign . . . . .	3388	files:	
flalign* . . . . .	3410	glyphtounicode . . . . .	54
flushleft . . . . .	3570	lwarp.css . . . . .	60, 275
flushright . . . . .	3567	lwarp_formal.css . . . . .	304
gather . . . . .	3300	lwarp_html.xdy . . . . .	309
gather* . . . . .	3322	lwarp_mathjax.txt . . . . .	309
itemize . . . . .	2011	lwarp_sagebrush.css . . . . .	299
lateximage . . . . .	3480	lwarp_tutorial.txt . . . . .	38
longtable . . . . .	2	lwarpmk . . . . .	311
LVerbatim . . . . .	1987	lwarpmk.conf . . . . .	270
LWR@nestspan . . . . .	616	project.css . . . . .	60
LWR@tabular . . . . .	2600	project.lwarpmkconf . . . . .	274
math . . . . .	3136	project_html.tex . . . . .	274
minipage . . . . .	3900	sample_project.css . . . . .	60, 308
multline . . . . .	3253	tutorial.tex . . . . .	38
multline* . . . . .	3276	FileSectionNames (boolean) . . . . .	48, 99
picture . . . . .	250, 3850	\flagverse . . . . .	443
quote . . . . .	1788	flalign (environment) . . . . .	3388
theindex . . . . .	3063	flalign* (environment) . . . . .	3410
tikzpicture . . . . .	8	Flare (program) . . . . .	30
titlepage . . . . .	58, 1474	float (package) . . . . .	342
titlingpage . . . . .	58, 1477, 1490	floatflt (package) . . . . .	344
Verbatim . . . . .	1844	\floatname . . . . .	11
verbatim . . . . .	1803	\floatplacement . . . . .	14
VerbatimClass . . . . .	1836	floatrow (package) . . . . .	345
verse . . . . .	2	\floatstyle . . . . .	17
warpall . . . . .	34, 56	flushleft (environment) . . . . .	3570
warpHTML . . . . .	42, 50, 56	flushright (environment) . . . . .	3567
warpprint . . . . .	35, 50, 56	font	
epigraph (package) . . . . .	338	bitmapped . . . . .	54
equation (environment) . . . . .	3204	Computer Modern . . . . .	54
error messages . . . . .	71	Deja Vu . . . . .	54

ligatures	54, 55	HTML	
packages	54	commands for a successful conver-	
selection	54	sion	57
size — lateximage	61, 217, 232	conversion settings	47
size — math, SVG	61, 217, 232	filename generation	48
size — xfrac	450	headings	74
fontenc (package)	54, 351	meta tag	
fontspec (package)	54, 351	description	59, 117
footmisc (package)	352	generator	132
footnote (package)	353	viewport	133
footnotehyper (package)	354	tabular column conversion	176
footnotes	117	\HTMLentity	497
for		\HTMLFileName	47, 508
HTML & PRINT	76	HTMLleftmargini (length)	
HTML output	76		68, 153, 154, 442
PRINT output	76	\HTMLunicode	498
\framebox	3968	HTMLvleftskip (length)	68, 153, 154, 442
framed (package)	355	\hyperindexref	3080
FrameMaker (program)	30	hyperref (package)	200, 360
Frequently Asked Questions	71	\hypertoc	3010
\frontmatter	1121	\hypertocfloat	3023
ftnright (package)	358		
\fup	4005		
		<b>I</b>	
<b>G</b>		icon	
gather (environment)	3300	warning	76
gather* (environment)	3322	ifplatform (package)	78
GELLMU (program)	28	images	
generator		graphicx package	238
HTML meta tag	132	in strange places	72
geometry (package)	84, 358	\includegraphics	3765
getttitlestring (package)	85	indentfirst (package)	363
GladTeX (program)	29	InDesign (program)	30
glyphtounicode (file)	54	\InlineClass	741
graphics (package)	238, 359	inputenc (package)	54, 363
\graphicspath	3583	internet (class)	28
graphicx (package)	238, 359	item, list, empty	57
		itemize (environment)	2011
		\itshape	3997
		<b>J</b>	
<b>H</b>		JavaScript	
Hevea (program)	28	MathJax	61, 217
\hfill	4006		
\HomeHTMLFileName	509		
\HomeHTMLFilename	47		
horizontal space		<b>K</b>	
between minipages	261	keyfloat (package)	364
\href	2784	kvoptions (package)	78
\hrulefill	4007		
\hspace	261, 4103	<b>L</b>	
		label	
		in HTML	135

math environment	226	[lwarp]:	
language HTML metadata	132	BaseJobname (option)	79
\LaTeX	4158	mathjax (option)	55, 79
LaTeX2HTML (program)	28	mathsvg (option)	55, 79
\LaTeXe	4158	warpHTML (option)	55, 79
lateximage (environment)	3480	warpprint (option)	55, 78
\LateximageFontSizeName	3444	lwarp (package)	55
lateximages		[lwarp-newproject]:	
font size	61, 217, 232	BaseJobName (option)	272
LaTeXML (program)	28	lwarpmk (option)	272
layout (package)	365	warpHTML (option)	272
lengths:		warpprint (option)	272
HTMLleftmargini	68, 153, 154, 442	lwarp-newproject (package)	272
HTMLvleftskip	68, 153, 154, 442	lwarp.css (file)	60, 275
LWR@minipageheight	252	lwarp_formal.css (file)	304
LWR@minipagewidth	251	lwarp_html.xdy (file)	309
LWR@tempheight	262	lwarp_mathjax.txt (file)	309
LWR@tempraise	262	lwarp_sagebrush.css (file)	299
LWR@tempwidth	262	lwarp_tutorial.txt (file)	38
\VerbatimHTMLWidth	155	lwarpmk (file)	311
vleftmargini	68, 153, 442	lwarpmk (option) [lwarp-newproject]	272
vleftskip	68, 153, 442, 443	lwarpmk.conf (file)	270
letterspace (package)	365	\LWR@addmathjax	3193
lettrine (package)	366	\LWR@afterendverbatim	1828
LibreOffice (program)	29	LWR@amsmultiline (boolean)	225
ligatures	54, 55, 84	\LWR@atbeginverbatim	1820
line numbers	76	\LWR@beginhideamsmath	3238
\linebreak	4104	\LWR@botnavigation	567
\LinkHome	559	\LWR@caption@begin	2863
lips (package)	367	\LWR@caption@end	2873
list item, empty	57	\LWR@clearmidrules	2357
listings (package)	368	\LWR@closeparagraph	808
\listof	2977	\LWR@closeprevious	587
lmodern (package)	54	\LWR@closetabledatacell	2109
lofdepth (counter)	214	LWR@colafterspec	171
longtable (environment)	2	LWR@colatspec	171
longtable (package)	372	LWR@colbangspec	171
lotdepth (counter)	214	LWR@colbeforespec	171
lscape (package)	374	\LWR@copyfile	321
ltcaption (package)	374	\LWR@createautosec	1130
LuaLaTeX		\LWR@currentcss	909
detection	83	\LWR@currentHTMLdescription	924
file & section names	124	\LWR@descitem	2048
\LuaLaTeX	4166	\LWR@docmidrule	2376
LuaLaTeX (program) [requirement]	32	LWR@doingapar (boolean)	109
\LuaTeX	4166	LWR@doinghline (boolean)	166
LVerbatim (environment)	1987	LWR@doingstartpars (boolean)	109
lwarp		LWR@doingtbrule (boolean)	167
loading	55	\LWR@domulticolumn	2449
options	55	\LWR@doubledollar	3106

<code>\LWR@endhideamsmath</code> .....	3246	<code>LWR@lateximagenumber (counter)</code> ..	232
<code>\LWR@endminipage</code> .....	3864	<code>\LWR@lateximagenumberref</code> .....	2675
<code>\LWR@endofline</code> .....	4052	<code>LWR@LIpage (counter)</code> .....	232
<code>\LWR@endsubminipage</code> .....	3885	<code>\LWR@loadafter</code> .....	70
<code>\LWR@ensuredoingapar</code> .....	782	<code>\LWR@loadbefore</code> .....	81
<code>LWR@externalfilecnt (counter)</code> ..	219	<code>\LWR@loadnever</code> .....	86
<code>\LWR@filenamoblanks</code> .....	992	<code>\LWR@longtabledatacaptiontag</code> ..	2470
<code>\LWR@filestart</code> .....	1294	<code>\LWR@lookforpackagename</code> .....	260
<code>\LWR@findword</code> .....	257	<code>\LWR@LwarpEnd</code> .....	1414, 4202
<code>\LWR@floatbegin</code> .....	2821	<code>\LWR@LwarpStart</code> .....	1348, 4202
<code>\LWR@floatend</code> .....	2834	<code>\LWR@maketitlesetup</code> .....	1715
<code>LWR@freezethisfloat (boolean)</code> ..	205	<code>\LWR@maybeinthisfloat</code> .....	2846
<code>\LWR@futurenonspaclet</code> .....	2088	<code>\LWR@maybeewtablerow</code> .....	2306
<code>\LWR@getexparray</code> .....	493	<code>LWR@midrulecounter (counter)</code> ..	180
<code>\LWR@getmynexttoken</code> .....	2095	<code>LWR@midrules</code> .....	180
<code>\LWR@hidelatexequation</code> .....	3179	<code>LWR@minipagedepth (counter)</code> .....	251
<code>\LWR@hspace</code> .....	4082	<code>LWR@minipagefullwidth (boolean)</code> .	253
<code>\LWR@htmlblockcomment</code> .....	660	<code>LWR@minipageheight (length)</code> ....	252
<code>\LWR@htmlblocktag</code> .....	668	<code>\LWR@minipagestartpars</code> .....	4056
<code>\LWR@htmlblocktagcomment</code> .....	673	<code>\LWR@minipagestoppars</code> .....	4062
<code>\LWR@htmlclosecomment</code> .....	647	<code>LWR@minipagethispar (boolean)</code> ..	253
<code>\LWR@htmlcomment</code> .....	656	<code>LWR@minipagewidth (length)</code> .....	251
<code>\LWR@htmldivclass</code> .....	698	<code>\LWR@multicolother</code> .....	2401
<code>\LWR@htmldivclassend</code> .....	701	<code>\LWR@multicolpartext</code> .....	2396
<code>\LWR@htmllelement</code> .....	711	<code>\LWR@multicolskip</code> .....	2407
<code>\LWR@htmllelementclass</code> .....	687	<code>\LWR@mynexttoken</code> .....	2087
<code>\LWR@htmllelementclassend</code> .....	692	<code>\LWR@mynshorttoc</code> .....	2928
<code>\LWR@htmllelementclassline</code> .....	704	<code>\LWR@nameref</code> .....	2665
<code>\LWR@htmllelementend</code> .....	714	<code>LWR@nestspan (environment)</code> .....	616
<code>LWR@htmlfilenumber (counter)</code> ....	99	<code>\LWR@newhtmlfile</code> .....	1070
<code>\LWR@htmlfileref</code> .....	2668	<code>\LWR@newlabel</code> .....	2702
<code>\LWR@htmlmathlabel</code> .....	3224	<code>\LWR@newref</code> .....	2751
<code>\LWR@htmlloppcomment</code> .....	647	<code>LWR@nextautofloat (counter)</code> ....	206
<code>\LWR@htmlrefsectionfilename</code> ....	552	<code>LWR@nextautopage (counter)</code> .....	206
<code>\LWR@HTMLsanitize</code> .....	3447	<code>LWR@nextequation (counter)</code> .....	221
<code>\LWR@HTMLsanitizeexpand</code> .....	3465	<code>\LWR@openparagraph</code> .....	787
<code>\LWR@htmlsectionfilename</code> .....	519	<code>\LWR@origcolspec</code> .....	2162
<code>\LWR@htmlspan</code> .....	622	<code>\LWR@originname</code> .....	3777
<code>\LWR@htmlspanclass</code> .....	630	<code>\LWR@originnames</code> .....	3784
<code>\LWR@htmltag</code> .....	638	<code>\LWR@parseaftercolumn</code> .....	2219
<code>\LWR@htmltagc</code> .....	609	<code>\LWR@parseatcolumn</code> .....	2167
<code>\LWR@includegraphicsb</code> .....	3676	<code>\LWR@parsebangcolumn</code> .....	2189
<code>\LWR@indexitem</code> .....	3068, 3072, 3076	<code>\LWR@parsebeforecolumn</code> .....	2210
<code>\LWR@indexsection</code> .....	3052	<code>\LWR@parseDcolumn</code> .....	2246
<code>LWR@infloatrow (boolean)</code> .....	242	<code>\LWR@parsenormalcolumn</code> .....	2232
<code>LWR@intabularmetadata (boolean)</code> .	167	<code>\LWR@parsepcolumn</code> .....	2242
<code>\LWR@itemizeitem</code> .....	2004	<code>\LWR@parseskipcolumn</code> .....	2229
<code>LWR@latestautopage (counter)</code> ....	206	<code>\LWR@parsetablecols</code> .....	2251
<code>LWR@lateximagedepth (counter)</code> ..	232	<code>LWR@prevFileDepth (counter)</code> ....	126
<code>\LWR@lateximagedepthref</code> .....	2672	<code>\LWR@printmccoldata</code> .....	2410







epigraph	338	nameref	388
eso-pic	339	needspace	389
etoolbox	77	newclude	68, 389
everyhook	85	newfloat	86
everypage	340	newtxmath	68
expl3	85	newunicodechar	54, 390
extramarks	340	nextpage	390
fancyhdr	341	nowidow	391
fancyvrb	85	ntheorem	392
float	342	pagenote	403
floatflt	344	parskip	404
floatrow	345	placeins	404
fontenc	54, 351	ragged2e	405
fontspec	54, 351	refcount	86
footmisc	352	rotating	406
footnote	353	setspace	407
footnotehyper	354	showidx	408
framed	355	showkeys	408
ftnright	358	sidecap	409
geometry	84, 358	sidenotes	410
getttitlestring	85	siunitx	68, 238
graphics	238, 359	soul	412
graphicx	238, 359	subfig	414
hyperref	200, 360	tabularx	420
ifplatform	78	textcomp	55
indentfirst	363	textpos	421
inputenc	54, 363	theorem	422
keyfloat	364	threeparttable	426
kvoptions	78	tikz	427
layout	365	titleps	87, 428
letterspace	365	titlesec	430
lettrine	366	titletoc	431
lips	367	titling	86, 432
listings	368	tocloft	433
lmodern	54	trivfloat	438
longtable	372	ulem	440
lscope	374	verse	442, 443
ltcaption	374	wallpaper	444
lwrap	55	wrapfig	445
lwrap-newproject	272	xcolor	447
makeidx	85	xfrac	87, 450
marginfix	375	xifthen	85
marginnote	376	xparse	84
mcaption	376	xstring	85
mdframed	377	zref	86
memoir	443	page	
microtype	55, 84, 384	inaccessible	47, 48
mparhack	385	\pagebreak	4106
multicol	386	pagenote (package)	403
multirow	188, 388	\pagenumbering	249





