



# The **lwarp** package

$\text{\LaTeX}$  to HTML

v0.43 — 2017/11/08

© 2016–2017 Brian Dunn  
bd@BDTechConcepts.com

## Abstract

The **lwarp** package allows  $\text{\LaTeX}$  to directly produce HTML5 output, using external utility programs only for the final conversion of text and images. Math may be represented by svg files or MathJax.

Documents may be produced by pdf $\text{\LaTeX}$ , Lua $\text{\LaTeX}$ , or Xe $\text{\LaTeX}$ . A `texlua` script removes the need for system utilities such as `make` and `gawk`, and also supports `xindy` and `latexmk`. Configuration is automatic at the first manual compile.

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

A modular package-loading system uses the **lwarp** version of a package for HTML when available. More than a hundred  $\text{\LaTeX}$  packages are supported with these high-level source compatibility replacements, and many others work as-is.

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

To update existing projects, see [section 2, Updates](#).

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

## License:

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

# 1 Supporting T<sub>E</sub>X development

T<sub>E</sub>X development:

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

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

Donations may be directed towards individual projects:

**TUG Bursary Fund:** Assistance for attending annual TUG meetings.

**CTAN:** The Comprehensive T<sub>E</sub>X Archive Network — Central storage for T<sub>E</sub>X.

**TeX Development Fund:** Support for specific projects.

**EduTeX:** Teaching and using T<sub>E</sub>X in schools and universities.

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

**LaTeX Project:** Modernizing the L<sup>A</sup>T<sub>E</sub>X core.

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

**LuaTeX:** Combining the pdfT<sub>E</sub>X engine and the Lua language.

**MetaPost:** Postscript graphics.

**MacTeX:** T<sub>E</sub>X for Mac.

**PDF Accessibility:** Modern PDF standards.

**Other:** Additional projects may be specified.

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

For users of MiKTeX: <https://miktex.org/donations.html>

For country-specific T<sub>E</sub>X users groups: <http://tug.org/usergroups.html>

# Contents

<b>1</b>	<b>Supporting TeX development</b>	<b>2</b>
	<b>List of Figures</b>	<b>31</b>
	<b>List of Tables</b>	<b>31</b>
<b>2</b>	<b>Updates</b>	<b>32</b>
<b>3</b>	<b>Introduction</b>	<b>40</b>
3.1	Supported packages and features . . . . .	41
<b>4</b>	<b>Alternatives</b>	<b>45</b>
4.1	Internet class . . . . .	45
4.2	TeX4ht . . . . .	45
4.3	Translators . . . . .	45
4.4	AsciiDoc . . . . .	46
4.5	Pandoc . . . . .	46
4.6	Word processors . . . . .	46
4.7	Commercial systems . . . . .	47
4.8	Comparisons . . . . .	47
<b>5</b>	<b>Installation</b>	<b>48</b>
5.1	Installing the lwarp package . . . . .	48
5.2	Installing the lwarpmk utility . . . . .	51
5.2.1	Using a local copy of lwarpmk . . . . .	52
5.3	Installing additional utilities . . . . .	53
<b>6</b>	<b>Tutorial</b>	<b>55</b>
6.1	Starting a new project . . . . .	55
6.2	Compiling the print version with lwarpmk . . . . .	59
6.3	Compiling the HTML version with lwarpmk . . . . .	60
6.4	Generating the SVG images . . . . .	61
6.5	Using MathJax for math . . . . .	62
6.6	Changing the CSS style . . . . .	63
6.7	Customizing the HTML output . . . . .	63
6.8	Using latexmk . . . . .	64
6.9	Using XeLaTeX or LuaLaTeX . . . . .	65
6.10	Using a glossary . . . . .	66
6.11	Cleaning auxiliary files . . . . .	67
6.12	Cleaning auxiliary and output files . . . . .	67
6.13	Processing multiple projects in the same directory . . . . .	67
6.14	Using the make utility . . . . .	67
<b>7</b>	<b>Additional details</b>	<b>68</b>
7.1	Font and UTF-8 support . . . . .	68

7.1.1	Indexes and UTF-8 . . . . .	69
7.2	lwarp package loading and options . . . . .	70
7.3	Customizing the HTML output . . . . .	72
7.3.1	Example HTML file naming . . . . .	76
7.4	Customizing the CSS . . . . .	77
7.5	Selecting the operating system . . . . .	78
7.6	Selecting actions for print or HTML output . . . . .	78
7.7	Commands to be placed into the warpprint environment . . . . .	80
7.8	Title page . . . . .	80
7.9	HTML page meta descriptions . . . . .	81
7.10	HTML page meta author . . . . .	81
7.11	Modifying xindy index processing . . . . .	82
<b>8</b>	<b>Special cases and limitations</b>	<b>82</b>
8.1	Formatting . . . . .	82
8.1.1	Text formatting . . . . .	82
8.1.2	Horizontal space . . . . .	83
8.1.3	Text alignment . . . . .	83
8.1.4	Accents . . . . .	83
8.1.5	Textcomp . . . . .	83
8.1.6	Superscripts and other non-math uses of math mode . . . . .	83
8.1.7	Empty \item followed by a new line of text or a nested list: . . . . .	83
8.1.8	Filenames in lists . . . . .	83
8.1.9	relsize package . . . . .	84
8.2	Boxes and minipages . . . . .	84
8.2.1	Save Boxes . . . . .	84
8.2.2	Minipages . . . . .	84
8.2.3	Side-by-side minipages . . . . .	85
8.2.4	Framed minipages and other environments . . . . .	85
8.2.5	fancybox package . . . . .	86
8.2.6	mdframed package . . . . .	88
8.3	Cross-references . . . . .	88
8.3.1	Page references . . . . .	88
8.3.2	cleveref and varioref packages . . . . .	89
8.3.3	Hyperlinks, hyperref, and url . . . . .	89
8.3.4	Footnotes and page notes . . . . .	89
8.4	Front and back matter . . . . .	90
8.4.1	Starred chapters and sections . . . . .	90
8.4.2	abstract package . . . . .	90
8.4.3	titling and authblk . . . . .	91
8.4.4	tocloft package . . . . .	91
8.4.5	appendix package . . . . .	91
8.4.6	pagenote package . . . . .	91
8.4.7	endnotes package . . . . .	91
8.4.8	glossaries package . . . . .	91
8.4.9	Index and the tocbibind package . . . . .	92

8.5	Math	93
8.5.1	Rendering tradeoffs	93
8.5.2	SVG option	94
8.5.3	MathJax option	94
8.5.4	ntheorem package	96
8.5.5	siunitx package	96
8.5.6	units and nicefrac packages	96
8.5.7	newtxmath package	97
8.6	Graphics	97
8.6.1	grffile package	98
8.6.2	color package	98
8.6.3	xcolor package	98
8.6.4	overpic package	99
8.7	Tabular	99
8.7.1	longtable package	102
8.7.2	supertabular and xtab packages	102
8.7.3	bigdelim package	102
8.8	Floats	103
8.8.1	float, trivfloat, and/or algorithmicx together	103
8.8.2	caption and subcaption packages	103
8.8.3	subfig package	103
8.8.4	floatrow package	103
8.8.5	keyfloat package	104
8.9	Miscellaneous	104
8.9.1	verse and memoir	104
8.9.2	newclude package	105
8.9.3	babel package	105
8.9.4	todonotes package	106
<b>9</b>	<b>EPUB conversion</b>	<b>107</b>
<b>10</b>	<b>Word-processor conversion</b>	<b>109</b>
10.1	Activating word-processor conversion	109
10.2	Additional modifications	111
10.3	Recommendations	112
10.4	Limitations	113
<b>11</b>	<b>Modifying lwarp</b>	<b>115</b>
11.1	Creating an lwarp version of a package	115
11.2	Adding a package to the lwarp.dtx file	116
11.3	Testing lwarp	116
11.4	Modifying lwarpmk	116
<b>12</b>	<b>Troubleshooting</b>	<b>118</b>
12.1	Using the lwarp.sty package	118
12.1.1	Debug tracing output	120

12.2	Compiling the <code>lwarp.dtx</code> file . . . . .	121
<b>1</b>	<b>lwarp.sty</b>	<b>121</b>
<b>13</b>	<b>Implementation</b>	<b>121</b>
<b>14</b>	<b>Section depths and HTML headings</b>	<b>122</b>
<b>15</b>	<b>Source Code</b>	<b>124</b>
<b>16</b>	<b>Detecting the T<sub>E</sub>X Engine — pdf<sub>l</sub>atex, lua<sub>l</sub>atex, xel<sub>l</sub>atex</b>	<b>125</b>
<b>17</b>	<b>Unicode Input Characters</b>	<b>125</b>
<b>18</b>	<b>Early package requirements</b>	<b>126</b>
<b>19</b>	<b>Operating-System portability</b>	<b>127</b>
19.1	Common portability code . . . . .	127
19.2	Unix, Linux, and Mac OS . . . . .	127
19.3	MS-Windows . . . . .	127
<b>20</b>	<b>Package options</b>	<b>128</b>
20.1	Conditional compilation . . . . .	131
<b>21</b>	<b>Misplaced packages</b>	<b>132</b>
<b>22</b>	<b>Required packages</b>	<b>137</b>
<b>23</b>	<b>Loading packages</b>	<b>141</b>
<b>24</b>	<b>File handles</b>	<b>144</b>
<b>25</b>	<b>Include a file</b>	<b>144</b>
<b>26</b>	<b>Copying a file</b>	<b>145</b>
<b>27</b>	<b>Debugging messages</b>	<b>146</b>
<b>28</b>	<b>HTML-conversion output modifications</b>	<b>147</b>
28.1	User-level controls . . . . .	147
28.2	Heading adjustments . . . . .	149
<b>29</b>	<b>Remembering original formatting macros</b>	<b>151</b>
<b>30</b>	<b>Accents</b>	<b>152</b>
<b>31</b>	<b>Configuration Files</b>	<b>154</b>
31.1	<code>project_html.tex</code> . . . . .	154

31.2	lwarpmk.conf . . . . .	154
31.3	project.lwarpmkconf . . . . .	155
31.4	lwarp.css . . . . .	156
31.5	lwarp_sagebrush.css . . . . .	183
31.6	lwarp_formal.css . . . . .	188
31.7	sample_project.css . . . . .	192
31.8	lwarp.xdy . . . . .	193
31.9	lwarp_mathjax.txt . . . . .	193
31.10	lwarpmk option . . . . .	195
<b>32</b>	<b>Stacks</b>	<b>206</b>
32.1	Assigning depths . . . . .	207
32.2	Closing actions . . . . .	207
32.3	Closing depths . . . . .	208
32.4	Pushing and popping the stack . . . . .	208
<b>33</b>	<b>Data arrays</b>	<b>210</b>
<b>34</b>	<b>HTML entities</b>	<b>210</b>
<b>35</b>	<b>HTML filename generation</b>	<b>211</b>
<b>36</b>	<b>Homepage link</b>	<b>214</b>
<b>37</b>	<b>\PrintStack diagnostic tool</b>	<b>214</b>
<b>38</b>	<b>Closing stack levels</b>	<b>215</b>
<b>39</b>	<b>PDF pages and styles</b>	<b>215</b>
<b>40</b>	<b>HTML tags, spans, divs, elements</b>	<b>217</b>
40.1	Mapping $\LaTeX$ Sections to HTML Sections . . . . .	217
40.2	Babel-French . . . . .	217
40.3	HTML tags . . . . .	218
40.4	Block tags and comments . . . . .	221
40.5	Div class and element class . . . . .	222
40.6	Single-line elements . . . . .	223
40.7	HTML5 semantic elements . . . . .	223
40.8	High-level block and inline classes . . . . .	224
40.9	Closing HTML tags . . . . .	226
<b>41</b>	<b>Paragraph handling</b>	<b>226</b>
<b>42</b>	<b>Paragraph start/stop handling</b>	<b>230</b>
<b>43</b>	<b>Page headers and footers</b>	<b>232</b>
<b>44</b>	<b>CSS</b>	<b>233</b>

<b>45</b>	<b>HTML meta description and author</b>	<b>234</b>
<b>46</b>	<b>Footnotes</b>	<b>235</b>
46.1	Regular page footnotes . . . . .	235
46.2	Minipage footnotes . . . . .	235
46.3	Titlepage thanks . . . . .	235
46.4	Regular page footnote implementation . . . . .	236
46.5	Minipage footnote implementation . . . . .	238
46.6	Printing pending footnotes . . . . .	239
<b>47</b>	<b>Marginpars</b>	<b>239</b>
<b>48</b>	<b>Splitting HTML files</b>	<b>240</b>
<b>49</b>	<b>Sectioning</b>	<b>245</b>
49.1	User-level starred section commands . . . . .	245
49.2	Book class commands . . . . .	246
49.3	Sectioning support macros . . . . .	247
49.4	\section and friends . . . . .	253
<b>50</b>	<b>Starting a new file</b>	<b>255</b>
<b>51</b>	<b>Starting HTML output</b>	<b>258</b>
<b>52</b>	<b>Ending HTML output</b>	<b>261</b>
<b>53</b>	<b>Title page</b>	<b>262</b>
53.1	Setting the title, etc. . . . .	263
53.2	Changes for \affiliation . . . . .	264
53.3	Printing the thanks . . . . .	265
53.4	Printing the title, etc. in HTML . . . . .	265
53.5	Printing the title, etc. in print form . . . . .	267
53.6	\maketitle for HTML output . . . . .	267
53.7	\published and \subtitle . . . . .	270
<b>54</b>	<b>Abstract</b>	<b>272</b>
<b>55</b>	<b>Quote and verse</b>	<b>272</b>
55.1	Citations and attributions . . . . .	272
55.2	Quotes, quotations . . . . .	273
55.3	Verse . . . . .	273
<b>56</b>	<b>Verbatim and tabbing</b>	<b>274</b>
<b>57</b>	<b>Theorems</b>	<b>277</b>
<b>58</b>	<b>Lists</b>	<b>278</b>
58.1	Itemize . . . . .	278



58.2	Enumerate	279
58.3	Description	280
<b>59</b>	<b>Tabular</b>	<b>281</b>
59.1	Limitations	281
59.2	Token lookahead	284
59.3	Booleans	285
59.4	Handling &, @, !, and bar	286
59.4.1	Localizing & catcodes	288
59.4.2	Handling &	289
59.4.3	Filling an unfinished row	290
59.5	Handling \\	291
59.6	Variables	292
59.7	Parsing @, >, <, !, bar columns	294
59.8	Parsing 'l', 'c', or 'r' columns	296
59.9	Parsing 'p', 'm', or 'b' columns	297
59.10	Parsing 'D' columns	297
59.11	Parsing the column specifications	298
59.12	Starting a new row	302
59.13	Printing vertical bar tags	303
59.14	Printing at or bang tags	303
59.15	Data opening tag	304
59.16	Midrules	306
59.17	Multicolumns	311
59.17.1	Parsing multicolumns	311
59.17.2	Multicolumn factored code	314
59.17.3	Multicolumn	315
59.17.4	Longtable captions	316
59.17.5	Counting HTML tabular columns	319
59.18	Multicolumnrow	320
59.19	Utility macros inside a table	321
59.20	Special-case tabular markers	322
59.21	Checking for a new table cell	323
59.22	\mrowcell	325
59.23	\mcolrowcell	325
59.24	New \tabular definition	326
59.25	Array	331
<b>60</b>	<b>Cross-references</b>	<b>331</b>
60.1	Setup	331
60.2	Zref setup	333
60.3	Labels	334
60.4	References	336
60.5	Hyper-references	338
<b>61</b>	<b>Floats</b>	<b>341</b>

61.1	Float captions . . . . .	342
61.1.1	Caption inside a float environment . . . . .	344
61.1.2	Caption and LOF linking and tracking . . . . .	344
<b>62</b>	<b>Table of Contents, LOF, LOT</b>	<b>347</b>
62.1	Reading and printing the TOC . . . . .	348
62.2	High-level TOC commands . . . . .	351
62.3	Side TOC . . . . .	351
62.4	Low-level TOC line formatting . . . . .	352
<b>63</b>	<b>Index and glossary</b>	<b>355</b>
<b>64</b>	<b>Restoring original formatting</b>	<b>357</b>
<b>65</b>	<b>Math</b>	<b>359</b>
65.1	Limitations . . . . .	359
65.1.1	Rendering tradeoffs . . . . .	359
65.1.2	SVG option . . . . .	360
65.1.3	MathJax option . . . . .	360
65.2	Inline and display math . . . . .	361
65.3	MathJax support . . . . .	364
65.4	Equation environment . . . . .	367
65.5	AMS Math environments . . . . .	369
65.5.1	Support macros . . . . .	369
65.5.2	Environment patches . . . . .	370
<b>66</b>	<b>Lateximages</b>	<b>377</b>
66.1	Description . . . . .	377
66.2	Support counters and macros . . . . .	378
66.3	Font size . . . . .	378
66.4	Sanitizing math expressions for HTML . . . . .	379
66.5	Equation numbers . . . . .	380
66.6	HTML <alt> tags . . . . .	381
66.7	lateximage . . . . .	382
<b>67</b>	<b>center, flushleft, flushright</b>	<b>386</b>
<b>68</b>	<b>Pre-loaded packages</b>	<b>387</b>
<b>69</b>	<b>Siunitx</b>	<b>387</b>
<b>70</b>	<b>Graphics print-mode modifications</b>	<b>388</b>
70.1	General limitations . . . . .	388
70.2	Print-mode modifications . . . . .	389
<b>71</b>	<b>Xcolor boxes</b>	<b>389</b>

<b>72</b>	<b>Cleveref</b>	<b>393</b>
<b>73</b>	<b>Picture</b>	<b>395</b>
<b>74</b>	<b>Boxes and Minipages</b>	<b>395</b>
74.1	Counters and lengths . . . . .	396
74.2	Footnote handling . . . . .	396
74.3	Minipage handling . . . . .	396
74.4	Parbox, makebox, framebox, fbox, raisebox . . . . .	401
<b>75</b>	<b>Direct formatting</b>	<b>406</b>
<b>76</b>	<b>Skips, spaces, font sizes</b>	<b>412</b>
<b>77</b>	<b>\phantomsection</b>	<b>420</b>
<b>78</b>	<b>\LaTeX and other logos</b>	<b>420</b>
78.1	HTML logos . . . . .	420
78.2	Print logos . . . . .	422
<b>79</b>	<b>\AtBeginDocument, \AtEndDocument</b>	<b>423</b>
<b>80</b>	<b>Trademarks</b>	<b>424</b>
<b>2</b>	<b>lwarp-a4.sty</b>	<b>425</b>
<b>81</b>	<b>a4</b>	<b>425</b>
<b>3</b>	<b>lwarp-a4wide.sty</b>	<b>425</b>
<b>82</b>	<b>a4wide</b>	<b>425</b>
<b>4</b>	<b>lwarp-a5comb.sty</b>	<b>425</b>
<b>83</b>	<b>a5comb</b>	<b>425</b>
<b>5</b>	<b>lwarp-abstract.sty</b>	<b>425</b>
<b>84</b>	<b>abstract</b>	<b>425</b>
<b>6</b>	<b>lwarp-adjmulticol.sty</b>	<b>427</b>
<b>85</b>	<b>adjmulticol</b>	<b>427</b>

---

<b>7</b>	<b>lwarp-addlines.sty</b>	<b>428</b>
86	addlines	428
<b>8</b>	<b>lwarp-afterpage.sty</b>	<b>428</b>
87	afterpage	428
<b>9</b>	<b>lwarp-algorithmicx.sty</b>	<b>429</b>
88	algorithmicx	429
<b>10</b>	<b>lwarp-alltt.sty</b>	<b>430</b>
89	alltt	430
<b>11</b>	<b>lwarp-amsthm.sty</b>	<b>430</b>
90	amsthm	430
<b>12</b>	<b>lwarp-anonchap.sty</b>	<b>433</b>
91	anonchap	433
<b>13</b>	<b>lwarp-anysize.sty</b>	<b>433</b>
92	anysize	433
<b>14</b>	<b>lwarp-appendix.sty</b>	<b>433</b>
93	appendix	433
<b>15</b>	<b>lwarp-arabicfront.sty</b>	<b>434</b>
94	arabicfront	434
<b>16</b>	<b>lwarp-authblk.sty</b>	<b>434</b>
95	authblk	434

<b>17</b>	<b>lwarp-balance.sty</b>	<b>435</b>
96	balance	435
<b>18</b>	<b>lwarp-bigdelim.sty</b>	<b>435</b>
97	bigdelim	435
<b>19</b>	<b>lwarp-bigstrut.sty</b>	<b>437</b>
98	bigstrut	437
<b>20</b>	<b>lwarp-bookmark.sty</b>	<b>437</b>
99	bookmark	437
<b>21</b>	<b>lwarp-booktabs.sty</b>	<b>438</b>
100	booktabs	438
<b>22</b>	<b>lwarp-boxedminipage.sty</b>	<b>438</b>
101	boxedminipage	438
<b>23</b>	<b>lwarp-boxedminipage2e.sty</b>	<b>438</b>
102	boxedminipage2e	438
<b>24</b>	<b>lwarp-breakurl.sty</b>	<b>439</b>
103	breakurl	439
<b>25</b>	<b>lwarp-caption2.sty</b>	<b>439</b>
104	caption2	439
<b>26</b>	<b>lwarp-ccaption.sty</b>	<b>440</b>
105	ccaption	440

---

<b>27</b>	<b>lwarp-changepage.sty</b>	<b>440</b>
106	change page	440
<b>28</b>	<b>lwarp-chngpage.sty</b>	<b>440</b>
107	chngpage	440
<b>29</b>	<b>lwarp-chappg.sty</b>	<b>440</b>
108	chappg	440
<b>30</b>	<b>lwarp-color.sty</b>	<b>441</b>
109	color	441
<b>31</b>	<b>lwarp-crop.sty</b>	<b>441</b>
110	crop	441
<b>32</b>	<b>lwarp-cuted.sty</b>	<b>441</b>
111	cuted	441
<b>33</b>	<b>lwarp-cutwin.sty</b>	<b>442</b>
112	cutwin	442
<b>34</b>	<b>lwarp-dblfnote.sty</b>	<b>442</b>
113	dblfnote	442
<b>35</b>	<b>lwarp-dcolumn.sty</b>	<b>443</b>
114	dcolumn	443
<b>36</b>	<b>lwarp-draftwatermark.sty</b>	<b>443</b>
115	draftwatermark	443

---

<b>37</b>	<b>lwarp-ebook.sty</b>	<b>444</b>
116	ebook	444
<b>38</b>	<b>lwarp-ellipsis.sty</b>	<b>444</b>
117	ellipsis	444
<b>39</b>	<b>lwarp-emptypage.sty</b>	<b>444</b>
118	emptypage	444
<b>40</b>	<b>lwarp-endnotes.sty</b>	<b>445</b>
119	endnotes	445
<b>41</b>	<b>lwarp-enumerate.sty</b>	<b>445</b>
120	enumerate	445
<b>42</b>	<b>lwarp-enumitem.sty</b>	<b>446</b>
121	enumitem	446
<b>43</b>	<b>lwarp-epigraph.sty</b>	<b>446</b>
122	epigraph	446
<b>44</b>	<b>lwarp-eso-pic.sty</b>	<b>447</b>
123	eso-pic	447
<b>45</b>	<b>lwarp-everypage.sty</b>	<b>448</b>
124	everypage	448
<b>46</b>	<b>lwarp-everyshi.sty</b>	<b>448</b>
125	everyshi	448

---

<b>47</b>	<b>lwarp-extramarks.sty</b>	<b>449</b>
126	extramarks	449
<b>48</b>	<b>lwarp-fancybox.sty</b>	<b>449</b>
127	fancybox	449
<b>49</b>	<b>lwarp-fancyhdr.sty</b>	<b>453</b>
128	fancyhdr	453
<b>50</b>	<b>lwarp-fancyvrb.sty</b>	<b>454</b>
129	fancyvrb	454
<b>51</b>	<b>lwarp-figcaps.sty</b>	<b>459</b>
130	figcaps	459
<b>52</b>	<b>lwarp-fix2col.sty</b>	<b>460</b>
131	fix2col	460
<b>53</b>	<b>lwarp-float.sty</b>	<b>460</b>
132	float and \newfloat	460
<b>54</b>	<b>lwarp-floatflt.sty</b>	<b>461</b>
133	floatflt	461
<b>55</b>	<b>lwarp-floatpag.sty</b>	<b>462</b>
134	floatpag	462
<b>56</b>	<b>lwarp-floatrow.sty</b>	<b>462</b>
135	floatrow	462



---

<b>57</b>	<b>lwarp-flushend.sty</b>	<b>468</b>
136	flushend	468
<b>58</b>	<b>lwarp-fncychap.sty</b>	<b>468</b>
137	fncychap	468
<b>59</b>	<b>lwarp-fnpos.sty</b>	<b>469</b>
138	fnpos	469
<b>60</b>	<b>lwarp-fontenc.sty</b>	<b>469</b>
139	fontenc	469
<b>61</b>	<b>lwarp-fontspec.sty</b>	<b>470</b>
140	fontspec	470
<b>62</b>	<b>lwarp-footmisc.sty</b>	<b>470</b>
141	footmisc	470
<b>63</b>	<b>lwarp-footnote.sty</b>	<b>471</b>
142	footnote	471
<b>64</b>	<b>lwarp-footnotehyper.sty</b>	<b>472</b>
143	footnotehyper	472
<b>65</b>	<b>lwarp-framed.sty</b>	<b>473</b>
144	framed	473
<b>66</b>	<b>lwarp-ftnright.sty</b>	<b>475</b>
145	ftnright	475

<b>67</b>	<b>lwarp-fullpage.sty</b>	<b>475</b>
146	fullpage	475
<b>68</b>	<b>lwarp-fullwidth.sty</b>	<b>476</b>
147	fullwidth	476
<b>69</b>	<b>lwarp-geometry.sty</b>	<b>476</b>
148	geometry	476
<b>70</b>	<b>lwarp-glossaries.sty</b>	<b>477</b>
149	glossaries	477
<b>71</b>	<b>lwarp-graphics.sty</b>	<b>478</b>
<b>150</b>	<b>graphics</b>	<b>478</b>
150.1	Graphics extensions . . . . .	478
150.2	Length conversions and graphics options . . . . .	479
150.3	Printing HTML styles . . . . .	481
150.4	\includegraphics . . . . .	482
150.5	Boxes . . . . .	486
<b>72</b>	<b>lwarp-graphicx.sty</b>	<b>489</b>
151	graphicx	489
<b>73</b>	<b>lwarp-grffile.sty</b>	<b>489</b>
152	grffile	489
<b>74</b>	<b>lwarp-hyperref.sty</b>	<b>490</b>
153	hyperref	490

---

<b>75</b>	<b>lwarp-hyperxmp.sty</b>	<b>494</b>
154	hyperxmp	494
<b>76</b>	<b>lwarp-idxlayout.sty</b>	<b>494</b>
155	idxlayout	494
<b>77</b>	<b>lwarp-indentfirst.sty</b>	<b>495</b>
156	indentfirst	495
<b>78</b>	<b>lwarp-inputenc.sty</b>	<b>495</b>
157	inputenc	495
<b>79</b>	<b>lwarp-keyfloat.sty</b>	<b>496</b>
158	keyfloat	496
<b>80</b>	<b>lwarp-layout.sty</b>	<b>497</b>
159	layout	497
<b>81</b>	<b>lwarp-letterspace.sty</b>	<b>497</b>
160	letterspace	497
<b>82</b>	<b>lwarp-lettrine.sty</b>	<b>498</b>
161	lettrine	498
<b>83</b>	<b>lwarp-lips.sty</b>	<b>499</b>
162	lips	499

---

<b>84</b>	<b>lwarp-listings.sty</b>	<b>499</b>
163	listings	499
<b>85</b>	<b>lwarp-longtable.sty</b>	<b>503</b>
164	longtable	503
<b>86</b>	<b>lwarp-lscape.sty</b>	<b>505</b>
165	lscape	505
<b>87</b>	<b>lwarp-ltcaption.sty</b>	<b>505</b>
166	ltcaption	505
<b>88</b>	<b>lwarp-luatodonotes.sty</b>	<b>505</b>
167	luatodonotes	505
<b>89</b>	<b>lwarp-marginfit.sty</b>	<b>506</b>
168	marginfit	506
<b>90</b>	<b>lwarp-marginfix.sty</b>	<b>506</b>
169	marginfix	506
<b>91</b>	<b>lwarp-marginnote.sty</b>	<b>507</b>
170	marginnote	507
<b>92</b>	<b>lwarp-mcaption.sty</b>	<b>507</b>
171	mcaption	507

---

<b>93</b>	<b>lwarp-mdframed.sty</b>	<b>507</b>
<b>172</b>	<b>mdframed</b>	<b>507</b>
172.1	Package loading . . . . .	508
172.2	Limitations . . . . .	508
172.3	Color and length HTML conversion . . . . .	509
172.4	Environment encapsulation . . . . .	509
172.5	Titles and subtitles . . . . .	511
172.6	New environments . . . . .	513
<b>94</b>	<b>lwarp-metalogo.sty</b>	<b>516</b>
<b>173</b>	<b>metalogo</b>	<b>516</b>
<b>95</b>	<b>lwarp-microtype.sty</b>	<b>517</b>
<b>174</b>	<b>microtype</b>	<b>517</b>
<b>96</b>	<b>lwarp-midfloat.sty</b>	<b>518</b>
<b>175</b>	<b>midfloat</b>	<b>518</b>
<b>97</b>	<b>lwarp-moreverb.sty</b>	<b>518</b>
<b>176</b>	<b>moreverb</b>	<b>518</b>
<b>98</b>	<b>lwarp-morewrites.sty</b>	<b>519</b>
<b>177</b>	<b>morewrites</b>	<b>519</b>
<b>99</b>	<b>lwarp-mparhack.sty</b>	<b>520</b>
<b>178</b>	<b>mparhack</b>	<b>520</b>
<b>100</b>	<b>lwarp-multicol.sty</b>	<b>520</b>
<b>179</b>	<b>multicol</b>	<b>520</b>

---

<b>101</b>	<b>lwarp-multirow.sty</b>	<b>521</b>
<b>180</b>	<b>multirow</b>	<b>521</b>
180.1	Multirow . . . . .	522
180.2	Combined multicolumn and multirow . . . . .	523
<b>102</b>	<b>lwarp-nameref.sty</b>	<b>524</b>
<b>181</b>	<b>nameref</b>	<b>524</b>
<b>103</b>	<b>lwarp-needspace.sty</b>	<b>524</b>
<b>182</b>	<b>needspace</b>	<b>524</b>
<b>104</b>	<b>lwarp-newclude.sty</b>	<b>525</b>
<b>183</b>	<b>newclude</b>	<b>525</b>
<b>105</b>	<b>lwarp-newunicodechar.sty</b>	<b>525</b>
<b>184</b>	<b>newunicodechar</b>	<b>525</b>
<b>106</b>	<b>lwarp-nextpage.sty</b>	<b>525</b>
<b>185</b>	<b>nextpage</b>	<b>525</b>
<b>107</b>	<b>lwarp-nonumonpart.sty</b>	<b>526</b>
<b>186</b>	<b>nonumonpart</b>	<b>526</b>
<b>108</b>	<b>lwarp-nopageno.sty</b>	<b>526</b>
<b>187</b>	<b>nopageno</b>	<b>526</b>
<b>109</b>	<b>lwarp-nowidow.sty</b>	<b>526</b>
<b>188</b>	<b>nowidow</b>	<b>526</b>

<b>110</b>	<b>lwarp-ntheorem.sty</b>	<b>527</b>
<b>189</b>	<b>ntheorem</b>	<b>527</b>
189.1	Limitations . . . . .	527
189.2	Options . . . . .	527
189.3	Remembering the theorem style . . . . .	528
189.4	HTML cross-referencing . . . . .	530
189.5	\newtheoremstyle . . . . .	530
189.6	Standard styles . . . . .	531
189.7	Additional objects . . . . .	533
189.8	Renewed standard configuration . . . . .	533
189.9	amsthm option . . . . .	534
189.10	Ending a theorem . . . . .	537
189.11	\NoEndMark . . . . .	537
189.12	List-of . . . . .	537
189.13	Symbols . . . . .	538
189.14	Cross-referencing . . . . .	538
<b>111</b>	<b>lwarp-overpic.sty</b>	<b>538</b>
<b>190</b>	<b>overpic</b>	<b>538</b>
<b>112</b>	<b>lwarp-pagenote.sty</b>	<b>539</b>
<b>191</b>	<b>pagenote</b>	<b>539</b>
<b>113</b>	<b>lwarp-paralist.sty</b>	<b>539</b>
<b>192</b>	<b>paralist</b>	<b>539</b>
<b>114</b>	<b>lwarp-parskip.sty</b>	<b>540</b>
<b>193</b>	<b>parskip</b>	<b>540</b>
<b>115</b>	<b>lwarp-pdfscape.sty</b>	<b>540</b>
<b>194</b>	<b>pdfscape</b>	<b>540</b>
<b>116</b>	<b>lwarp-pdfsync.sty</b>	<b>540</b>
<b>195</b>	<b>pdfsync</b>	<b>540</b>

---

<b>117</b>	<b>lwarp-pfnote.sty</b>	<b>541</b>
196	pfnote	541
<b>118</b>	<b>lwarp-placeins.sty</b>	<b>541</b>
197	placeins	541
<b>119</b>	<b>lwarp-prelim2e.sty</b>	<b>541</b>
198	prelim2e	541
<b>120</b>	<b>lwarp-quotchap.sty</b>	<b>542</b>
199	quotchap	542
<b>121</b>	<b>lwarp-ragged2e.sty</b>	<b>543</b>
200	ragged2e	543
<b>122</b>	<b>lwarp-realscripts.sty</b>	<b>544</b>
201	realscripts	544
<b>123</b>	<b>lwarp-reysize.sty</b>	<b>544</b>
202	resize	544
<b>124</b>	<b>lwarp-romanbar.sty</b>	<b>546</b>
203	romanbar	546
<b>125</b>	<b>lwarp-romanbarpagenumber.sty</b>	<b>546</b>
204	romanbarpagenumber	546



---

<b>126</b>	<b>lwarp-rotating.sty</b>	<b>546</b>
205	rotating	546
<b>127</b>	<b>lwarp-rotfloat.sty</b>	<b>547</b>
206	rotfloat	547
<b>128</b>	<b>lwarp-savetrees.sty</b>	<b>548</b>
207	savetrees	548
<b>129</b>	<b>lwarp-scalefnt.sty</b>	<b>548</b>
208	scalefnt	548
<b>130</b>	<b>lwarp-sectsty.sty</b>	<b>548</b>
209	sectsty	548
<b>131</b>	<b>lwarp-setspace.sty</b>	<b>549</b>
210	setspace	549
<b>132</b>	<b>lwarp-shadow.sty</b>	<b>550</b>
211	shadow	550
<b>133</b>	<b>lwarp-showidx.sty</b>	<b>550</b>
212	showidx	550
<b>134</b>	<b>lwarp-showkeys.sty</b>	<b>551</b>
213	showkeys	551
<b>135</b>	<b>lwarp-sidecap.sty</b>	<b>551</b>
214	sidecap	551

---

<b>136</b>	<b>lwarp-sidenotes.sty</b>	<b>552</b>
215	sidenotes	552
<b>137</b>	<b>lwarp-siunitx.sty</b>	<b>553</b>
216	siunitx	553
<b>138</b>	<b>lwarp-soul.sty</b>	<b>555</b>
217	soul	555
<b>139</b>	<b>lwarp-stabular.sty</b>	<b>557</b>
218	stabular	557
<b>140</b>	<b>lwarp-subfig.sty</b>	<b>557</b>
219	subfig	557
<b>141</b>	<b>lwarp-supertabular.sty</b>	<b>563</b>
220	supertabular	563
<b>142</b>	<b>lwarp-syntonly.sty</b>	<b>564</b>
221	syntonly	564
<b>143</b>	<b>lwarp-tables.sty</b>	<b>565</b>
222	tables	565
<b>144</b>	<b>lwarp-tabularx.sty</b>	<b>565</b>
223	tabularx	565
<b>145</b>	<b>lwarp-tabulary.sty</b>	<b>566</b>
224	tabulary	566

<b>146</b>	<b>lwarp-textarea.sty</b>	<b>566</b>
225	textarea	566
<b>147</b>	<b>lwarp-textcomp.sty</b>	<b>567</b>
226	textcomp	567
226.1	Limitations . . . . .	567
226.2	Package loading . . . . .	567
226.3	Remembering original defintions . . . . .	567
226.4	HTML symbols . . . . .	568
226.4.1	pdf $\TeX$ symbols . . . . .	568
226.4.2	X $\TeX$ and Lua $\TeX$ symbols . . . . .	568
226.5	HTML dicritics . . . . .	569
226.6	Inside a lateximage . . . . .	570
<b>148</b>	<b>lwarp-textpos.sty</b>	<b>570</b>
227	textpos	570
<b>149</b>	<b>lwarp-theorem.sty</b>	<b>571</b>
228	theorem	571
228.1	Remembering the theorem style . . . . .	571
228.2	CSS patches . . . . .	573
<b>150</b>	<b>lwarp-threeparttable.sty</b>	<b>575</b>
229	threeparttable	575
<b>151</b>	<b>lwarp-tikz.sty</b>	<b>576</b>
230	tikz	576
<b>152</b>	<b>lwarp-titleps.sty</b>	<b>577</b>
231	titleps	577

<b>153</b>	<b>lwarp-titleref.sty</b>	<b>580</b>
232	titleref	580
<b>154</b>	<b>lwarp-titlesec.sty</b>	<b>580</b>
233	titlesec	580
<b>155</b>	<b>lwarp-titletoc.sty</b>	<b>582</b>
234	titletoc	582
<b>156</b>	<b>lwarp-titling.sty</b>	<b>584</b>
235	titling	584
<b>157</b>	<b>lwarp-tocbibind.sty</b>	<b>589</b>
236	tocbibind	589
<b>158</b>	<b>lwarp-tocloft.sty</b>	<b>591</b>
237	tocloft	591
<b>159</b>	<b>lwarp-todonotes.sty</b>	<b>596</b>
238	todonotes	596
<b>160</b>	<b>lwarp-transparent.sty</b>	<b>597</b>
239	transparent	597
<b>161</b>	<b>lwarp-trivfloat.sty</b>	<b>597</b>
240	trivfloat	597
240.1	Combining \newfloat, \trivfloat, and algorithmicx . . . . .	598

<b>162</b>	<b>lwarp-typearea.sty</b>	<b>599</b>
241	typearea	599
<b>163</b>	<b>lwarp-ulem.sty</b>	<b>599</b>
242	ulem	599
<b>164</b>	<b>lwarp-upref.sty</b>	<b>601</b>
243	upref	601
<b>165</b>	<b>lwarp-verse.sty</b>	<b>601</b>
244	verse	601
<b>166</b>	<b>lwarp-wallpaper.sty</b>	<b>603</b>
245	wallpaper	603
<b>167</b>	<b>lwarp-wrapfig.sty</b>	<b>604</b>
246	wrapfig	604
<b>168</b>	<b>lwarp-xcolor.sty</b>	<b>605</b>
247	xcolor	605
247.1	Limitations . . . . .	605
247.2	Xcolor definitions: location and timing . . . . .	606
247.3	Package loading . . . . .	608
247.4	Remembering and restoring original definitions . . . . .	608
247.5	HTML color style . . . . .	608
247.6	HTML border . . . . .	609
247.7	High-level macros . . . . .	609
<b>169</b>	<b>lwarp-xfrac.sty</b>	<b>613</b>
248	xfrac	613

<b>170</b>	<b>lwarp-xltextra.sty</b>	<b>615</b>
------------	---------------------------	------------

249	xltextra	615
-----	----------	-----

<b>171</b>	<b>lwarp-xmpincl.sty</b>	<b>616</b>
------------	--------------------------	------------

250	xmpincl	616
-----	---------	-----

<b>172</b>	<b>lwarp-xtab.sty</b>	<b>616</b>
------------	-----------------------	------------

251	xtab	616
-----	------	-----

<b>173</b>	<b>lwarp-zwpagelayout.sty</b>	<b>618</b>
------------	-------------------------------	------------

252	zwpagelayout	618
-----	--------------	-----

<b>Change History and Index</b>	<b>620</b>
---------------------------------	------------

## List of Figures

1	tutorial.tex listing . . . . .	56
---	--------------------------------	----

## List of Tables

1	$\text{\LaTeX}$ -HTML generation — lwarp package — Supported functions . .	41
2	Required software programs . . . . .	49
3	Files created along with the print version . . . . .	58
4	Package options . . . . .	70
5	HTML settings . . . . .	72
6	Section HTML headings for word-processor conversion . . . . .	113
7	Section depths and HTML headings . . . . .	122
8	Tabular baseline . . . . .	298
9	Tabular HTML column conversions . . . . .	299
10	Cross-referencing data structures . . . . .	332
11	Float data structures . . . . .	341
12	AMStm package — CSS styling of theorems and proofs . . . . .	430
13	Ntheorem package — CSS styling of theorems and proofs . . . . .	527
14	Theorem package — CSS styling of theorems and proofs . . . . .	571

## 2 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 620.

### v0.43:

- Docs: Reorganized HTML customization, added an HTML settings table. See section 7.3.
- footnotes
  - Added FootnoteDepth to control the placement of pending footnotes before section breaks. By default, pending footnotes are printed before each \subparagraph or higher.
- sectioning
  - Fix: Expansion in section name.
- tabular
  - Fix: Ignore spaces in tabular column specification.
  - Fix: Tabular rules at bottom or when finishing incomplete rows.
  - Fix: \multicolumn at/bang/before/after specifications, trim, and vertical rules.
  - Fix: supertabular and xtab column misalignment.
- math
  - Fix: equation\*.
  - Fix: svg math in a section name.
  - Fix: \ref and \eqref in svg math.
- packages
  - Added todonotes and luatodonotes (but only disabled).
  - Added breakurl.
  - hyperref: Fix: Several macros were made robust, \Gauge added.

### v0.42:

- Support  $\TeX$ !
  - Added  $\TeX$  development support page.
- word-processor conversion
  - Improved assistance for word-processor conversions when boolean FormatWP is set true. See section 10.
    - The boolean FormatWordProcessor has been renamed FormatWP.
    - The boolean HTMLMarkFloats has been renamed WMarkFloats.
    - New booleans control whether to place additional marks around minipages, at the table of contents, at the LOF and LOT, and whether to print math as  $\TeX$  source for copy/paste into the LibreOffice Writer TeXMaths extension.
    - Improved formatting for numerous objects. See section 10.


⚠ name change

⚠ name change




- tabbing
  - Add: tabbing environment.
- overpic
  - Add: overpic package. See section 190.
- math
  - Fix: Text copy/paste of  $\mathcal{AMS}$  math environment numbers and names.
  - Improved `\ensuremath`.
  - MathJax with siunitx: Updated script and documentation.
- symbols
  - textcomp: Improved `\interrobangdown`.
  - realscripts: Fix for subscripts in a `lateximage`.
- load order
  - morewrites: Enforces loading before lwarp.

**v0.41:**





- tabular
  - Added tabular vertical rules, subject to some limitations. See the rules section of section 8.7.
  - Improved booktabs: Width and trim are honored.
-  new syntax
  - Added `\mcolrowcell` for empty cells inside a `\multicolumnrow`. Use `\mcolrowcell` instead of `\mrowcell` for two-dimensional cells created by `\multicolumnrow`. Continue to use `\mrowcell` for empty cells in a `\multirow`. See section 180.2 on page 523.
  - Fix: Unfinished tabular rows are automatically filled.
  - Fix for tabular column specifiers while using babel-french. (`\NoAutoSpacing` is activated then nullified inside the tabular, due to a conflict with the tabular column parsing code.)

**v0.40:**

- graphics, graphicx
  - graphics and graphicx have been moved from the lwarp core, and are only loaded if requested with `\usepackage`.
- `\includegraphics path`
  - Improved graphics `\graphicspath` support. Multiple image directories may now be used. Refer to .pdf files without a file extension to allow the HTML version to use a .svg, .png, .jpg, or .gif version instead. See section 8.6.
-  image file extensions
  - grffile is now directly supported instead of emulated.
- bigdelim
  - Fix for bigdelim, and improved documentation. See section 97.
- symbols
  - Improved  $\TeX$  and textcomp symbols.
  - Fix for  $\TeX$  logos and `\InlineClass`, etc. inside a `lateximage`.
  - Fix for xltextra with  $\LaTeX$ .
  - Fixes for tocibind with `\simplechapter`, etc.
  - Fixes for `\multicolumnrow` and `\nullfonts` with older versions of `multirow` and `xparse`.
  - Added `\underline`.
- fixes
  -

- margins
  - columns
  - footnotes
  - tabular
  - sectioning
- Added adjmulticol.
  - Added cuted, midfloat.
  - Added pfnote, fnpos, dblfnote.
  - Added stabular, tabs.
  - Added sectsty, anonchap, quotchap.

**v0.39:**

- title pages
  -  `\published and \subtitle`
  -  load order
  - tabular
  - multi column/row cell
  -  macros inside tabular
  -  tabular defined inside another environment
  - tabular
  - margins
  - page layout
- Improved the titlepage HTML code, `\thanks` notes, and `\maketitle`. titling is no longer required, but is still supported. The `\published` and `\subtitle` fields are no longer provided, but `\AddSubtitlePublished` replicates them using titling. See section 53.7. `authblk` is added, and should be loaded before titling. See section 53.
  - `\multirow` now supports the new optional `vpos` argument.
  - Added `\multicolumnrow` for combined `\multicolumn` and `\multirow`. See section 180.2.
  - Tabular special cases:
    - Added `\TabularMacro` to mark custom macros inside tabular data cells, avoiding row corruption. See section 8.7.
    - Added `\ResumeTabular` for use when a tabular environment is defined inside another environment. See section 8.7.
  - Added supertabular, xtab, bigstrut, bigdelim.
  - Added fullwidth.
  - Added addlines, anysize, a4, a4wide, a5comb, textarea, zwpagelayout, typearea, ebook.

**v0.38:**

- forced single-pass compile
  - starred sections
  - updated tutorial
  - packages
  - font size
  - page numbering
- Added `lwarpmk print1` and `lwarpmk html1` actions to force a compile of the project a single time. Useful when multiple passes are not needed, or changes were not detected.
  - Added `\ForceHTMLPage` and `\ForceHTMLTOC` to force a starred sectional unit onto its own HTML page and with its own TOC entry. See section 8.4.1.
  - Modified the tutorial to use the new `\ForceHTMLPage` and `\ForceHTMLTOC` macros.
  - Added appendix, tocbibind, fncychap, fix2col.
  - Added relsize, scalefnt.
  - Added realscripts, metalogo, xltextra.
  - Added grffile, romanbar.
  - Added arabicfront, chappg, nonumonpart, nopageno, romanbarpagenumber.

## front &amp; back matter

- Docs: Improved description of the use of front/back matter. See section 8.4.
- Fix: color requests xcolor.
- Fix: \part for article class.

**v0.37:**

## \include for HTML

## latexmk

- \include now maintains independent .aux files for HTML versions.
- comment, used by lwarp, now maintains independent cut files for print and HTML versions, helping latexmk to better know whether to recompile.

## accents and symbols

## babel-french

- Improved support for  $\TeX$  accents, textcomp, siunitx symbols.
- Improved babel-french handling for load order and ~ tilde.

**v0.36:**

## boxes and frames

## babel-french

## footnotes

## siunitx

- Recorganized the documentation section regarding special cases and limitations. (Section 8)
- Improved source formatting.
- \fbox and related now use \fboxsep and \fboxrule.
- \makebox and \framebox now use width and position.
- \fcolorbox and related now work inside a lateximage.
- babel-french: Improvements for French variants, load order, footnotes, ellipses.
- Improved footnote numbering. lateximage footnotes now appear as regular footnotes to match the numbering of the print version. Also fixed a regression with MathJax.
- Improved siunitx units.
- Fix for filenames while using MathJax.
- Fix for \rule when xcolor is not loaded.
- Added transparent, upref.

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

**v0.34:**

## ⚠ Optional arguments

- BlockClass's optional argument has been moved in front of the mandatory argument:

BlockClass[style]{class} (NEW)

instead of:

BlockClass{class}[style] (OLD)

This change makes it more consistent with  $\TeX$  standards, and avoids problems with space between arguments.

## ⚠ Optional arguments

spans with minipages

framing minipages

lateximage, SVG math,  
tabular

eqnarray

verbatim packages

framing packages

list packages

babel-french

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

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

- Improved compatibility between spans, minipages, lists, frames, and math. Handles minipages and lists inside an HTML span, such as an `\fbox` containing a minipage, although with minimal HTML formatting. See section 8.2.2. `\fboxBlock` is added to frame minipages, tables, and lists with full HTML formatting but no longer inline, and behaves as `\fbox` for print output. The `fminipage` environment is added for framed minipages, as an environment with full HTML formatting, and draws a framed minipage in print output. See section 8.2.4. `\fbox` and minipages now often work in SVG math and `lateximages`. MathJax supports `\fbox`, but not `\fboxBlock` nor `fminipage`.

- Improved compatibility between `lateximage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, `booktabs`.

- Improved font control for `lateximages` and SVG math.

- Added the `eqnarray` environments.

- `fancyvrb` is no longer required (preloaded), but is still supported.

- Added `verbatim` and `moreverb`.

- Added `fancybox`, `boxedminipage2e` and `shadow`.

- `enumitem` is no longer required, but is still supported.

- Added `enumerate` and `paralist`.

- `titleps` is no longer required, but is still supported.

- Added `crop`.

- Added `rotfloat`, `marginfit`, and several minor packages; see the change log.

- Adds fixed-width HTML spaces around punctuation when using `babel-french`. Lua<sub>T</sub><sub>E</sub>X does not yet use the extra punctuation spacing.

## v0.33:

- `Tabular @` and `!` columns now have their own HTML columns.
- `&` catcode changes are localized, perhaps causing errors about the tab alignment character `&`, so any definitions of macros or environments which themselves contain `tabular` and `&` must be enclosed within `\StartDefiningTabulars` and `\EndDefiningTabulars`. See section 59.4.1. This change is not required for the routine use of tables, but only when a table is defined inside another macro or environment, and while also using the `&` character inside the definition. This may include the use inside conditional expressions.

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

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

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

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

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

**v0.30:**

#### `lwarp-newproject`

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

To modify existing documents, remove from the document source:

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

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

#### HTML setup changes.

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

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

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

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

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

```

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

```
- The file `lwarp_html.xdy` has been renamed `lwarp.xdy`. To update each document's project:
  1. Make the changes shown above.
  2. Recompile the document in print mode. This updates the project's configuration files, and also generates the new file `lwarp.xdy`.
  3. The old file `lwarp_html.xdy` may be deleted.
- The new lwarp package option `xdyFilename` may be used to tell `lwarpmk` to use a custom `.xdy` file instead of `lwarp.xdy`. See section 7.11.
- Improvements in index processing:
  - `xindy`'s language is now used for index processing as well as glossary.
  - Print mode without `latexmk` now uses `xindy` instead of `makeindex`.
  - `texindy/xindy` usage depends on `pdflatex` vs `xelatex`, `lualatex`.
  - For `pdflatex` and `texindy`, the `-C utf8` option is used. This is supported in modern distributions, but a customized `lwarpmk.lua` may need to be created for use with older distributions.

**v0.29:**

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

**v0.28:**

- `\HTMLAuthor {<name>}` assigns HTML meta author if non-empty. Defaults to `\theauthor`.
- Boolean `HTMLDebugComments` controls whether HTML comments are added for closing `<div>`s, opening and closing sections, etc.
- Boolean `FormatEPUB` changes HTML output for easy EPUB conversion via an external program. Removes per-file headers, footers, and nav. Adds footnotes per chapter/section.
- Boolean `FormatWordProcessor` changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments. Name changed to `FormatWP` as of v0.42.
- Boolean `HTMLMarkFloats` adds text marks around floats only if `FormatWordProcessor`. These make it easier to identify float boundaries, which are to be manually converted to word-processor frames. Name changed to `WPMarkFloats` as of v0.42.
- Updated for the new MathJax CDN repository.
- Adds tabulary.
- Supports the options syntax for graphics.
- Improved index references, now pointing exactly to their target.
- Adds glossaries. `lwarpmk` is modified to add `printglossary` and `htmlglossary` actions.

### 3 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 a  $\text{\LaTeX}$ -generated PDF file which happens to have HTML5 tags, to a number of HTML5 plain-text files and accompanying images.

---

<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!



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

When requesting packages during HTML conversion, lwarp first looks to see if it has its own modified version to use instead of the usual  $\text{\LaTeX}$  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 41.)

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

### 3.1 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  $\text{\LaTeX}$  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  $\text{\LaTeX}$  code.

Table 1:  $\text{\LaTeX}$ -HTML generation — lwarp package — Supported functions

Category	Status
Engines:	pdf $\text{\LaTeX}$ , X <sub>Y</sub> $\text{\LaTeX}$ , Lua $\text{\LaTeX}$
Classes:	book, report, or article. memoir is planned.
Page layout:	Emulates geometry, fancyhdr, titleps, addlines, anysize, a4, a4wide, a5comb, textarea, zwpagelayout, typearea, ebook.

## lwarp Supported Functions — continued

Category	Status
Sectioning:	Supported, with hyperlinks. Honors <code>tocdepth</code> and <code>secnumdepth</code> . Adds <code>FileDepth</code> for splitting the HTML output. Files may be numbered sequentially or named according to section name. Common short words and punctuation is removed from the filenames. Emulates <code>titlesec</code> , <code>fncychap</code> , <code>sectsty</code> , <code>anonchp</code> , <code>quotchap</code> .
Table of contents, figures, tables:	Supported, with hyperlinks. Supports <code>tocbibind</code> . Emulates <code>titletoc</code> and <code>tocloft</code> .
Title page:	<code>\maketitle</code> , <code>titlepage</code> , <code>titling</code> , <code>authblk</code> .
Front & back matter:	<code>abstract</code> , <code>appendix</code>
Indexing:	<code>texindy</code> is used, with hyperlinks. <code>idxlayout</code> is emulated.
Glossary:	<code>glossaries</code> and <code>xindy</code> are used.
Bibliography:	Supported text-only.
Cross-references:	Emulated, with hyperlinks. <code>hyperref</code> , <code>cleveref</code> , <code>varioref</code> , <code>url</code> , <code>breakurl</code> .
Languages:	<code>babel</code> . ( <code>polyglossia</code> is untested.)
Margin notes:	<code>marginfit</code> , <code>marginfix</code> .
Footnotes:	Adds <code>FootnoteDepth</code> to print footnotes at section breaks. <code>footnote</code> , <code>footmisc</code> , <code>marginnote</code> , <code>sidenote</code> , <code>pagenote</code> , <code>endnotes</code> .
Math:	Supported. Converted to SVG images with HTML <code>&lt;alt&gt;</code> tags containing the $\text{\LaTeX}$ source for the math expression. MathJax supported as an alternative. $\mathcal{AMS}$ environments are supported. User-defined macros are available during conversion, due to native $\text{\LaTeX}$ processing.
Theorems:	Support for native $\text{\LaTeX}$ theorems, plus <code>theorem</code> , <code>amsthm</code> , <code>ntheorem</code> .
Add'l math:	<code>delarray</code> , <code>bm</code> , math fonts via SVG images.
Units and fractions:	<code>siunitx</code> , <code>xfrac</code> , <code>nicefrac</code> , <code>units</code>

## lwarp Supported Functions — continued

Category	Status
Floats:	Appear where declared. float, rotfloat, newfloat, caption and subcaption, subfig, capt-of, placeins, trivfloat, floatrow, subfloat, keyfloat, wrapfig, cutwin, floatflt.
Tabular	tabular environment, array, tabularx, tabulary, threeparttable, multirow, longtable, supertabular, xtab, booktabs.
Graphics	graphics and graphicx are emulated. \includegraphics supports width, height, origin, angle, and scale tags, and adds class. References to PDF files are changed to SVG, other image types are accepted as well. \rotatebox and \scalebox are supported as well as HTML can handle. rotating is emulated but all objects are unrotated. picture and tikz are converted to an SVG image. grffile and overpic are supported.
xcolor:	<b>Supported.</b> Full package color names, any color models, and <b>mixing</b> . <b>\textcolor</b> , <b>\colorbox</b> , <b>\fcolorbox</b> . Enhanced for HTML compatibility.
Lists:	Standard $\LaTeX$ environments are supported, along with enumitem, enumerate, paralist. Spacing is still controlled by CSS.
Environments:	Standard $\LaTeX$ environments are supported.
minipage:	Supported with some HTML5-imposed limitations. Nested minipages are supported. Footnotes appear at the bottom of the HTML page.
Quotations:	verse, csquotes, epigraph
Verbatim:	verbatim, moreverb, fancyvrb (except for verbatim footnotes).
Frames:	framed, fancybox, mdframed, boxedminipage2e, shadow.
multicol:	Emulated, with CSS3. Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported.

## lwarp Supported Functions — continued

Category	Status
fullwidth:	Emulated.
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> , <code>soul</code> , <code>relsize</code> , <code>scalegnt</code> , and <code>realscripts</code> are supported.
Ordinals:	<code>nth</code> , <code>fmtcount</code> , and <code>engord</code> are supported.
Text ligatures:	Ligatures for symbols are supported. Ligatures for <code>f</code> , <code>q</code> , <code>t</code> are intentionally turned off because many simpler browsers do not display them correctly. Modern full-featured browsers re-create these ligatures on-the-fly.
Horizontal space:	HTML output for <code>thin-unbreakable</code> , <code>unbreakable</code> , <code>\enskip</code> , <code>\quad</code> , <code>\qqquad</code> , <code>\hspace</code> .
Rules:	<code>\rule</code> with <code>width</code> , <code>height</code> , <code>raise</code> , <code>text color</code> .
HTML reserved characters:	<code>\&amp;</code> , <code>\textless</code> , and <code>\textgreater</code> are converted to HTML entities.
Fonts:	Used as-is. Appear in math expressions or embedded image environments.
Symbols:	Native $\TeX$ diacriticals, <code>textcomp</code> .
Working as-is:	Various utility, calculation, file, and text-only packages, such as <code>calc</code> , <code>fileerr</code> , <code>somedefs</code> , <code>trace</code> , <code>xspace</code> .
Where:	
<b>Supported:</b>	The existing $\TeX$ package is used.
<b>Emulated:</b>	The $\TeX$ package is not used, but some/all of its functions are emulated. Null functions, lengths, and counters are provided for source compatibility.

## 4 Alternatives

Summarized below are several other ways to convert a  $\TeX$  or other document to HTML. Where an existing  $\TeX$  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.

### 4.1 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.

### 4.2 TeX4ht

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

This system uses native  $\TeX$  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.

### 4.3 Translators

These systems use external programs to translate a subset of  $\TeX$  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  $\TeX$  math into HTML:

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

## 4.4 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  $\TeX$  and HTML.

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

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

The Asciidoctor-LaTeX project is adding additional  $\TeX$ -related features.

### Asciidoctor-Latex:

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

## 4.5 Pandoc

Prog Pandoc

A markup system which also reads and writes  $\TeX$  and HTML.

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

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

## 4.6 Word processors

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

Prog LibreOffice

Prog OpenOffice

## 4.7 Commercial systems

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

## 4.8 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.

## 5 Installation

Table 2 shows the tools which are used for the  $\text{\LaTeX}$  to HTML conversion. In most cases, these will be available via the standard package-installation tools.

### 5.1 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.

**$\text{\TeX}$  Live:** If using a  $\text{\TeX}$  Live distribution, try installing via tlmgr:

```
Enter ⇒ tlmgr install lwarp
```

**MiK $\text{\TeX}$ :** If using MiK $\text{\TeX}$ , 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  $\text{\TeX}$ -related packages.

**CTAN TDS archive:** lwarp may be downloaded from the Comprehensive  $\text{\TeX}$  Archive:

1. See <http://ctan.org/pkg/lwarp> for the lwarp package.
2. Download the TDS archive: lwarp.tds.zip
3. Find the  $\text{\TeX}$  local directory:

**$\text{\TeX}$  Live:**

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

**MiK $\text{\TeX}$ :**

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 2: Required software programs

**Provided by your  $\text{\LaTeX}$  distribution:**

From  $\text{\TeX}$ Live: <http://tug.org/texlive/>.

**$\text{\LaTeX}$ :** pdf $\text{\LaTeX}$ , x $\text{\LaTeX}$ , or lua $\text{\LaTeX}$ .

**The lwarp package:** This package.

**The lwarpmk utility:** Provided along with this package. This should be an operating-system executable in the same way that pdf $\text{\LaTeX}$  or latexmk is. It is possible to have the lwarp package generate a local copy of lwarpmk called lwarpmk.lua. See table 3.

**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 $\text{\TeX}$  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  $\text{\LaTeX}$  code. On a MiK $\text{\TeX}$  system, Perl may need to be installed first.

**pdfcrop:** Used to pull images out of the  $\text{\LaTeX}$  PDF.

**Poppler PDF utilities:**

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

**pdfseparate:** Used to pull images out of the  $\text{\LaTeX}$  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 ⇒ `mktxlsr`  
     — 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 5.2.1 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 5.2.

**Project-local CTAN .dtx and .ins files:** The .dtx and .ins files may be downloaded to a project directory, then compiled right there, alongside the document source files. The resultant \*.sty and 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!

## 5.2 Installing the lwarpmk utility

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

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

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

Otherwise, continue with the following:

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

```
Enter ⇒ kpsewhich lwarpmk.lua
```

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

4. Create lwarpmk:

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

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

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

This will be something like:

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

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

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

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

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

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

- (c) Make the link executable:

```
Enter ⇒ chmod 0755 lwarpmk
```

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

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

**Windows 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.

### 5.2.1 Using a local copy of lwarpmk

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

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

#### Unix:

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

#### Windows:

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

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

Or:

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

### 5.3 Installing additional utilities

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

Enter the following in a command line. If each programs' version is displayed, then that utility is already installed. See table 2 on page 49.

```
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/pdftocairo
```

```
Prog  pdftotext
Prog  pdfseparate
Prog  pdftocairo
```

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

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

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

1. Install Homebrew from <https://brew.sh/>:  
Enter ⇒  
`/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`
2. Install the Poppler utilities:  
Enter ⇒ `brew install poppler`

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

1. See table 2 on page 49.
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.**

## 6 Tutorial

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

### 6.1 Starting a new project

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

File `tutorial.tex`

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

#### Copy from the documentation PDF:

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

#### Copy from the lwarp documentation directory:

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

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

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

File `lwarp_tutorial.txt`

 **Note: .txt suffix!**

 **Bad formatting!**

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

3. Compile the project:

Enter  $\Rightarrow$  `pdflatex tutorial.tex`

(several times)

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

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

A number of new files are created when `tutorial.tex` is compiled, as shown in table 3. These files are created by the `lwarp` package.

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

Figure 1: tutorial.tex listing

Note: There are two pages!

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

\documentclass{book}

\usepackage{iftex}

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

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

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

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

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



```

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

\begin{document}

\maketitle % Or titlepage/titlingpage environment.

% An article abstract would go here.

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

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

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

\end{document}

```

Table 3: Files created along with the print version

<b>tutorial.pdf:</b>	The PDF output from $\text{\LaTeX}$ . The print version of the document.
<b>tutorial_html.tex:</b>	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.
<b>Auxiliary files:</b>	The usual $\text{\LaTeX}$ files .aux, .log, .out, .toc, .lof, .idx. When an HTML version of the document is created, _html versions of the auxiliary files will also be generated.
<b>lwarpmk.conf:</b>	A configuration file for lwarpmk, which is used to automate the compilation of PDF or HTML versions of the document.
<b>tutorial.lwarpmkconf:</b>	Another configuration file used by lwarpmk, which is only useful if you wish to have several projects residing in the same directory.
<b>.css files:</b>	lwarp.css, lwarp_formal.css, lwarp_sagebrush.css These files are standard for lwarp, and are not meant to be modified by the user.
<b>sample_project.css:</b>	An example of a user-customized css file, which may be used for project-specific changes to the lwarp defaults.
<b>lwarp.xdy:</b>	Used by lwarp while creating an index. This file should not be modified by the user. A custom file may be used instead, if necessary.
<b>lwarp_mathjax.txt:</b>	Inserted into the HTML files when MathJax is used to display math. This file should not be modified by the user.
<b>comment.cut:</b>	A temporary file used by lwarp to conditionally process blocks of text. This file may be ignored.

---

When the lwarpmk option is given to the lwarp package:

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

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

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

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

## 6.2 Compiling the print version with `lwarpmk`

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

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

1. Re-compile the print version:

Enter  $\Rightarrow$  `lwarpmk print`

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

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

Enter  $\Rightarrow$  `lwarpmk print`

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

4. Force a recompile to occur.

Enter  $\Rightarrow$  `lwarpmk again`

Enter  $\Rightarrow$  `lwarpmk print`

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

5. Process the index.<sup>45</sup>

Enter  $\Rightarrow$  `lwarpmk printindex`

6. Recompile again to include the index.

Enter  $\Rightarrow$  `lwarpmk print`

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

Enter  $\Rightarrow$  `lwarpmk print1`

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

---

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

<sup>4</sup>A `lwarpmk printglossary` command is also available to process a glossary produced with the glossaries package. See section 8.4.8.

<sup>5</sup>Also see section 8.4.9 for index options.

### 6.3 Compiling the HTML version with lwarpmk

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

1. Compile the HTML version:

```
Enter ⇒ lwarpmk html
```

- (a) `lwarpmk` uses  $\text{\LaTeX}$  to process `tutorial_html.tex` to create `tutorial_html.pdf`.
- (b) `pdftotext` is then used to convert to the file `tutorial_html.html`. This file is a plain-text file containing HTML tags and content for the entire document.
- (c) `lwarpmk` manually splits `tutorial_html.html` into individual HTML files according to the HTML settings. For this tutorial, the result is `tutorial.html` (the home page), along with `First-chapter.html`<sup>6</sup>, `Some-math.html`, and the document's index in `_Index.html`.<sup>7</sup>

2. View the homepage in a web browser.

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

math

Note that math is still displayed as its plain-text  $\text{\LaTeX}$  source until the images of the math expressions have been generated. Math may be displayed as svg images or by a MathJax script, as seen in sections 6.4 and 6.5.

3. Force a recompile:

```
Enter ⇒ lwarpmk again
```

```
Enter ⇒ lwarpmk html
```

```
Enter ⇒ lwarpmk print
```

4. Process the HTML index and recompile:<sup>89</sup>

```
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.

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

```
Enter ⇒ lwarpmk html1
```

<sup>6</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>7</sup>`index.html` is commonly used as a homepage, so the document index is in `_Index.html`.

<sup>8</sup>A `lwarpmk htmlglossary` command is also available to process a glossary produced with the `glossaries` package. See section 8.4.8.

<sup>9</sup>Also see section 8.4.9 for index options.

## 6.4 Generating the SVG images

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


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


1. Create the svg images:


```
Enter ⇒ 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.

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

 **Lots of files!** Expressing math as svg images has the advantage of representing the math exactly as  $\text{\LaTeX}$  would, but has the disadvantage of requiring an individual file for each math expression. There is no attempt at reusing the same file each time the same expression occurs, so each time  $\$x\$$  is used, for example, yet another file is created. For a document with a large amount of math, see section 6.5 to use MathJax instead.

## 6.5 Using MathJax for math

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


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

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

2. Recompile

```
Enter ⇒ lwarpmk html
```

3. Reload the math page.

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

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

## 6.6 Changing the CSS style

For a formal css style, add to the preamble:

```
\usepackage{lwarp}  
...  
\CSSFilename{lwarp_formal.css}  
...  
\begin{document}
```

For a modern css style, `lwarp_sagebrush.css` is also provided:

```
\CSSFilename{lwarp_sagebrush.css}
```

See section 7.4 for more information about modifying the CSS styling of the document.

## 6.7 Customizing the HTML output

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

See section 7.3 for more information.

## 6.8 Using latexmk

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

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

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

2. Recompile the printed version of the document.

```
Enter ⇒ lwarpmk print
```

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

3. Recompile the document.

```
Enter ⇒ lwarpmk print
```

and/or

```
Enter ⇒ lwarpmk html
```

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

**forced single-pass recompile** A single recompile may be forced with:

```
Enter ⇒ lwarpmk print1
```

and/or

```
Enter ⇒ lwarpmk html1
```



## 6.9 Using XeLaTeX or LuaLaTeX

X<sub>g</sub>TeX or LuaTeX may be used instead of E<sub>g</sub>TeX.

1. Remove the auxiliary files for the project:

Enter  $\Rightarrow$  `lwarpmk cleanall`

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

Enter  $\Rightarrow$  `xelatex tutorial.tex`

-or-

Enter  $\Rightarrow$  `lualatex tutorial.tex`

When the recompile occurs, the configuration files for `lwarpmk` are modified to remember which T<sub>E</sub>X engine was used. X<sub>g</sub>TeX or LuaTeX will be used for future runs of `lwarpmk`.

3. To recompile the document:

Enter  $\Rightarrow$  `lwarpmk print`

-and-

Enter  $\Rightarrow$  `lwarpmk html`

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

## 6.10 Using a glossary

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

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

---

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

---

To process the glossary for the print version:

```
Enter ⇒   lwarpmk printglossary
```

To process the glossary for the HTML version:

```
Enter ⇒   lwarpmk htmlglossary
```

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

## 6.11 Cleaning auxiliary files

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

```
Enter ⇒ lwarpmk clean
```

## 6.12 Cleaning auxiliary and output files

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

```
Enter ⇒ lwarpmk cleanall
```

## 6.13 Processing multiple projects in the same directory

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

To create each project:

```
Enter ⇒ pdflatex project_a
```

```
Enter ⇒ pdflatex project_b
```

Each project is given its own configuration file:

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

To compile each project with `lwarkmk`:

```
Enter ⇒ lwarpmk print project_a
```

```
Enter ⇒ lwarpmk html project_b
```

## 6.14 Using the make utility

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

```
lwarpmk pdftohtml [project]
```

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

## 7 Additional details

### 7.1 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>TeX and LuaTeX 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>TeX or LuaTeX:

Pkg `fontspec`

ligatures

- `fontspec` and font choices

lwarp sets the following to turn off TeX 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`

Pkg `fontenc`

Pkg `inputenc`

Pkg `newunicodechar`

- `lmodern` or other font-related packages
- `fontenc`
- `inputenc`
- `newunicodechar`

File	glyphtounicode	- \input glyphtounicode.tex
		- \input glyphtounicode-cmr.tex% from the pdfx package
		- \pdfgentounicode=1
Pkg	cmap	- cmap
Pkg	textcomp	- textcomp
Pkg	microtype	- microtype is automatically used by lwarp to turn off f,q,t,T,Q ligatures for the same browser-related reasons shown above. Also, the monospaced font is used during HTML tag generation to turn off T <sub>E</sub> X ligatures.
	ligatures	

3. `\usepackage{lwarp}` (section 7.2) goes after any of the above, followed by:
4. ... the rest of the preamble and the main document.

### 7.1.1 Indexes and UTF-8

lwarp uses the xindy program to process indexes.

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

While using `pdflatex`, the `texindy` program is used with the `-C utf8` option, which is newly supported in recent distributions of  $\text{\TeX}$ . This option correctly sorts index entries into headings while using Latin languages, but will not work well with others.  $\text{\XeTeX}$  or  $\text{\LuaTeX}$  are recommended for non-Latin languages.

For an older distribution of  $\text{\TeX}$ , it may be necessary to generate a local version of `lwarpmk.lua` and modify it to remove the `-C utf8` option from the `texindy` call. See section 11.4.

Table 4: Package options

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

## 7.2 lwarp package loading and options

lwarp supports book, report, and article classes.

Pkg	<code>lwarp</code>	Load the lwarp package immediately after the font and UTF-8 setup commands.
Opt	<code>warpprint</code>	Usually controlled by <code>lwarpmk</code> , and not set in the document. Select the <code>warpprint</code> option to generate print output (default), or the <code>warpHTML</code> option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual <code>pdflatex</code> , etc. When lwarp 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	<code>warpHTML</code>	
Opt	<code>mathsvg</code>	For math display, select <code>mathsvg</code> (default), or <code>mathjax</code> . For more information about the math options, see section 8.5.
Opt	<code>mathjax</code>	
Opt	<code>OSWindows</code>	See section 7.5 if using Windows.


---


Opt	BaseJobname	Not intended for the user. Used internally by lwarp when creating the *_html.tex file used to compile the HTML version. See section 20.
Opt	HomeHTMLFilename	See section 7.3.
Opt	HTMLFilename	See section 7.3.
Opt	IndexLanguage	If using an index or glossary, see section 20.
	Opt latexmk	Has lwarpmk use latexmk to recompile the document several times if necessary. Otherwise, lwarpmk attempts to determine for itself whether to recompile. See section 7.3.
	Opt lwarpmk	If you wish to have lwarp generate a local copy of lwarpmk.lua for archival or local-installation purposes, compile the print version with the lwarpmk option set. See section 20.
Opt	xdyFilename	The default xindy filename is lwarp.xdy. If you wish to use a custom .xdy file for index generation, see section 20.

Table 5: HTML settings

Option	Description
SideTOCDepth	Sectioning depth of the sideroc.
FileDepth	Sectioning depth of the file splits.
CombineHigherDepths	Combine higher section levels.
FileSectionNames	Use section names for file names, else use numbers.
FootnoteDepth	Sectioning depth of footnotes.
\abstractname	The name of the abstract.
\CSSFilename	The css for the following files.
\HTMLLanguage	The html lang tag.
\HTMLAuthor	The HTML author meta tag.
\HTMLDescription	The HTML description meta tag.
\HTMLFirstPageTop	Heading for the home page.
\HTMLPageTop	Heading for the other pages.
\HTMLPageBottom	Footing for all pages.

### 7.3 Customizing the HTML output

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

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

1. Clear all the HTML, PDF, and auxiliary files:

```
Enter ⇒ lwarpmk clearall
```

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

```
Enter ⇒ lwarpmk print
```

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

```
Enter ⇒ lwarpmk html
```

#### Options for the lwarp package:

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



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

`HomeHTMLFilename=index`

#### filename underscores

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

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

See section 7.3.1 for examples of naming and numbering HTML files.

Opt `HTMLFilename` **HTMLFilename:** A filename prefix for the rest of the HTML web pages. Useful for numbered web pages with a common prefix. May be empty. See section 7.3.1 for examples of naming and numbering HTML files.

Opt `latexmk` **latexmk:** Controls whether lwarp uses latexmk to compile the document. This setting is written to `lwarpmk`’s configuration files.

Opt `mathsvg` **mathsvg:** Selects svg display for math output. (The default.)

Opt `mathjax` **mathjax:** Selects MathJax for math output.

Default: `false`

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

Ctr `tocdepth` **tocdepth:** Sectioning depth of the table of contents. See section 14 for a list of L<sup>A</sup>T<sub>E</sub>X stack depths.

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

#### sideroc

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

It is recommended to set:

`SideTOCDepth=FileDepth`

or

`SideTOCDepth=FileDepth+1`

#### ⚠ inaccessible pages

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

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

Default: `-5`

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



Bool `CombineHigherDepths`  
Default: `true`

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

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

**FileSectionNames:** If `true`, web page filenames are derived from a sanitized version of the section names. If `false`, web pages are numbered. Either way, the `HTMLFilename` option is used as a prefix. See section 7.3.1 for examples of naming and numbering HTML files.

Ctr `FootnoteDepth`  
Default: `5`

**FootnoteDepth:** Determines where to place pending footnotes. 5 places footnotes before each break down to the `\subparagraph` level. 1 places footnotes before each `\section` break. Any pending footnotes are also placed at the bottom of each page before each file break.

`\abstractname`  
Default: `Abstract`

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

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

`\CSSFilename`  
Default: `lwarp.css`


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

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

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

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

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

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

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

**Placed in the document wherever necessary:**

Env <code>warpprint</code>	<code>warpprint:</code> 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	warppHTML	<b>warppHTML:</b> 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.
	<code>\warpprintonly</code>	<b>\warpprintonly:</b> <code>{\langle contents \rangle}</code> A macro version of the warpprint environment.
	<code>\warppHTMLonly</code>	<b>\warppHTMLonly:</b> <code>{\langle contents \rangle}</code> A macro version of the warppHTML environment.

### 7.3.1 Example HTML file naming

Examples of ways to name or number HTML files:

#### Numbered HTML nodes:

Example: Homepage `index.html`, and `node-1`, `node-2`.<sup>10</sup>

---

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

---

#### Named HTML sections, no prefix:

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

---

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

---

#### Named HTML sections, with prefix:

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

<sup>10</sup>See `\SetHTMLFileNumber` to number in groups by chapter, for example.

---

```

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

```

---

## 7.4 Customizing the CSS

`\CSSFilename`  
 Default: `lwarp.css`

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

```

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

```

The styles provided by lwarp include:

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

**lwarp\_formal.css:** A formal style with a serif fonts and a traditional look.

**lwarp\_sagebrush.css:** A style with muted colors, gradient backgrounds, additional borders, and rounded corners.

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

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



**Rename it!**

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

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

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

File `project.css`

File `sample_project.css`

---

```

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

/* 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 --- ) */

```

---

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

## 7.5 Selecting the operating system

Prog Unix lwarp tries to detect which operating system is being used. UNIX / MAC OS / LINUX is the default (collectively referred to as “UNIX” in the configuration files), and MS-WINDOWS is supported as well.

Prog Mac OS

Prog Linux

Prog MS-Windows If WINDOWS is not correctly detected, use the lwarp option `OSWindows`.

Prog Windows

Opt OSWindows When detected or specified, the operating-system path separator used by lwarp is modified, the boolean `usingOSWindows` is set true. This boolean may be tested by the user for later use.

## 7.6 Selecting actions for print or HTML output

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

For most of built-in  $\TeX$  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.

## 7.7 Commands to be placed into the warpprint environment

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

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

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

Also see section 12: [Troubleshooting](#).

## 7.8 Title page

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

---

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

---

The title is used in the meta tags in the HTML files, and the rest are used in `\maketitle`. To use a `\subtitle` or `\published` field, see section 53.7.

`\maketitle` Use `\maketitle` just after the `\begin{document}`, as this will establish the title of the homepage. Optionally, use a `titlepage` environment instead.

Env `titlepage` The `titlepage` environment may be used to hold a custom title page. The `titlepage` will be set in a `<div>` class `titlepage`, and `\printtitle`, etc. may be used inside this environment.

Env `titlingpage` Another form of custom title page, where `\maketitle` is allowed, and additional information may be included as well.

`\title` `{\title}`




Avoid newlines in the `\title`; these will interfere with the file break and CSS detection. Use a `\subtitle` command instead (section 53.7). 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`.

`\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.

## 7.9 HTML page meta descriptions

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

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

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

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

## 7.10 HTML page meta author

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

## 7.11 Modifying xindy index processing

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

To customize index processing:

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

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

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

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

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

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


## 8 Special cases and limitations


Also see section 12: [Troubleshooting](#).

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

### 8.1 Formatting

#### 8.1.1 Text formatting

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

 **HTML special chars** `&`, `<`, and `>` have special meanings in HTML. If `\&`, `\textless`, and `\textgreater` are used, the proper result should occur in HTML, but there may be HTML parsing problems if these special characters occur unescaped in program listings or other verbatim text.

### 8.1.2 Horizontal space

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

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

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

### 8.1.3 Text alignment


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

### 8.1.4 Accents

Native  $\TeX$  accents such as `\''` will work, but many more kinds of accents are available when using Unicode-aware  $\XeTeX$  and  $\LuaTeX$ .

### 8.1.5 Textcomp

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

 **Missing symbols** Many `textcomp` symbols are not supported by many fonts. Try using more complete fonts in the CSS, but expect to see gaps in coverage.

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

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

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

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

### 8.1.8 Filenames in lists

[filename underscore](#) Escape underscores in the filenames:


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

### 8.1.9 relsize package

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

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

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


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

## 8.2 Boxes and minipages

### 8.2.1 Save Boxes

$\LaTeX$  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.


### 8.2.2 Minipages

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

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

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

**in a span** There is limited support for minipages inside an HTML `<span>`. An HTML `<div>` cannot appear inside a `<span>`. While in a `<span>`, minipages, and parboxes, and any enclosed lists have limited HTML tags, resulting in an “inline” format, without markup except for HTML breaks. Use `\newline` or `\par` for an HTML break.

- size** When using `\linewidth`, `\textwidth`, and `\textheight`, widths and heights are scaled proportionally to a 6×9 inch text area.
- no-width minipages** A minipage of width exactly `\linewidth` is automatically given no HTML width.
- full-width minipages** A new macro `\minipagefullwidth` requests that the next minipage be generated without an HTML width attribute, allowing it to be the full width of the display rather than the fixed width given.
-  **text alignment** Nested minipages adopt their parent's text alignment in HTML, whereas in regular  $\TeX$  PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

### 8.2.3 Side-by-side minipages

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

### 8.2.4 Framed minipages and other environments

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

- `\fbox` For a framed object, options include:
- `\fboxBlock`
- Env `fminipage` **To remove the frame in HTML output:** Place the `\fbox` command and its closing brace inside `warpprint` environments. This will nullify the frame for HTML output.
- To frame the contents inline with some formatting losses in HTML:** This is the default action of `\fbox` when enclosing a minipage. During HTML output, `\fbox` nullifies the HTML tags for minipage, `\parbox`, and lists. The contents are included as inline text inside the `\fbox`'s `<span>` of class `framebox`. For lists, line breaks are converted to HTML breaks. The result is a plain-text inline version of the contents, framed inline with the surrounding text, but lacking any extra HTML markup.
- To frame the contents on their own line with improved formatting in HTML:** A new command `\fboxBlock` is included, intended to be a direct replacement for `\fbox` for cases where the `\fbox` surrounds a minipage, table, or list. For print output, this behaves as `\fbox`. For HTML output, the contents are placed inside an HTML `<div>` with the class `framed`, resulting in the contents being

For inline minipage and lists:

placed on their own line with a frame surrounding them. The contents preserve their HTML formatting, so lists and minipages look nicer, and valid HTML is created for a tabular. While an `\fbox` containing a tabular is valid  $\text{\LaTeX}$  code, the result in HTML is problematic since a table is a `<div>` not a `<span>`, so use `\fboxBlock` around a tabular, or else place the tabular inside a minipage, or use `fminipage`, described next. Also see below regarding the “Misplaced alignment tab character &” error.

For display tabular,  
minipages, and lists:

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

colored boxes and frames:

**To create colored frames and boxes:** See section 247 for `xcolor`’s `\colorbox` and `\fcolorbox`, and `lwarp`’s additional `\colorboxBlock` and `\fcolorboxBlock`.

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

**To frame equations:** See section 127 for the `fancybox` package.

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

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

---

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

---

### 8.2.5 fancybox package

framed equation example

`fancybox`’s documentation has an example `FramedEqn` environment which combines `math`, `\Sbox`, a `minipage`, and an `\fbox`. This combination requires that the entire environment be enclosed inside a `lateximage`, which is done by adding `\lateximage`

at the very start of FramedEqn's beginning code, and `\endlateximage` at the very end of the ending code. Unfortunately, the HTML `alt` attribute is not used here.

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

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

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

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

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


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

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

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

### 8.2.6 mdframed package

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

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

**font** For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

where `\textbf` must appear just before the comma and will receive the following text as its argument (since the text happens to be between braces in the mdframed source). Since lwarp does not support `\bfseries` and friends, only one font selection may be made at a time.


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

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


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

## 8.3 Cross-references


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

 **underscores** does not yet work when generating math for MathJax.

**\nameref** `\nameref` refers to the most recently-used section where the `\label` was defined. If

 **empty link** no section has been defined before the `\label`, the link will be empty. Index entries also use `\nameref` and have the same limitation.

### 8.3.1 Page references

 **L<sup>A</sup>T<sub>E</sub>X page numbers** The printed page does not translate to the HTML page, so `\pageref` references are converted to parentheses containing `\pagerefPageFor`, which defaults to “see”, followed by a hyperlink to the appropriate object.




Ex:

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

`\pagerefPageFor` may be redefined to “page for”, empty, etc. See page 337.

### 8.3.2 **cleveref and varioref packages**

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

Ex:

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

See `\cpagerefFor` at page 394 to redefine the message which is printed for page number references.


### 8.3.3 **Hyperlinks, hyperref, and url**

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

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

### 8.3.4 **Footnotes and page notes**

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

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

## 8.4 Front and back matter

### 8.4.1 Starred chapters and sections

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

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

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

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

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

⚠ inaccessible HTML page

`\ForceHTMLTOC` To automatically force the HTML version of the document to have a TOC entry for a starred section, use `\ForceHTMLTOC` just before the `\chapter*` or `\section*`. The TOC will only be assigned for HTML output, not for print output, and it will appear in the main TOC and also the sidetoc per page.

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

### 8.4.2 abstract package

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

### 8.4.3 titling and authblk

package support

⚠ load order

lwarp supports the native  $\text{\LaTeX}$  titling commands, and also supports the packages `authblk` and `titling`. If both are used, `authblk` should be loaded before `titling`.

`\published` and `\subtitle`

If using the `titling` package, additional `titlepage` fields for `\published` and `\subtitle` may be added by using `\AddSubtitlePublished` in the preamble. See section 53.7.

### 8.4.4 tocloft package

Opt tocloft titles

⚠ tocloft & other packages

If using `tocloft` with `tocbibind`, `anonchp`, `fncychap`, or other packages which change chapter title formatting, load `tocloft` with its `titles` option, which tells `tocloft` to use standard  $\text{\LaTeX}$  commands to create the titles, allowing other packages to work with it.

### 8.4.5 appendix package

⚠ incorrect TOC link

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

### 8.4.6 pagenote package

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

### 8.4.7 endnotes package

table of contents

To place the endnotes in the TOC, use:

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

HTML page

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

```
\ForceHTMLPage
\theendnotes
```

### 8.4.8 glossaries package

Pkg glossaries

`xindy` is required for `glossaries`.

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

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

#### placement and toc options

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

##### Numbered section, on its own HTML page:

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

##### Unnumbered section, inline with the current HTML page:

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

##### Unnumbered section, on its own HTML page:

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

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

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

Opt `lwarpmk htmlglossary`

### 8.4.9 Index and the `tocbibind` package

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



#### tocloft & other packages

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

#### placement and toc options

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

**Inline, with a manual TOC entry:**

A commonly-used method to introduce an index in a  $\text{\LaTeX}$  document:

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

**On its own HTML page, with a manual TOC entry:**

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

**Inline, with an automatic TOC entry:**

Pkg `tocbibind`

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

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

**On its own HTML page, with an automatic TOC entry:**

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

Opt `tocbibind` `numindex`  
[numbered index section](#)

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

See section [63](#) for `lwarp`'s core index and glossary code, and section [236](#) for `tocbibind`.

## 8.5 Math

### 8.5.1 Rendering tradeoffs

[Math rendering](#)

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

- SVG files** In its current implementation, rendering math as images creates a new svg file for each expression. 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 check-summing may be used to remove the need for duplicate files.
- SVG inline** Another approach could be to in-line the svg files directly into the HTML. This may reduce the number of files and potentially speed loading the images, but slows the display of the rest of the document before the images are loaded.
- PNG files** Others converters have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but svg files are the preferred approach for scalable graphics.
- MathML** Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than svg drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 9 regarding EPUB output with MathJax.

### 8.5.2 SVG option

- SVG math option** For svg math, math is rendered as usual by  $\text{\LaTeX}$  into the initial PDF file using the current font<sup>11</sup>, then is captured from the PDF and converted to svg graphics via a number of utility programs. The svg format is a scalable-vector web format, so math may be typeset by  $\text{\LaTeX}$  with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML alt attribute carries the  $\text{\LaTeX}$  code which generated the math, allowing copy/paste of the  $\text{\LaTeX}$  math expression into other documents.
- SVG image font size** The size of the math and text used in the svg image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, for ex:  
`\renewcommand{\LateximageFontSizeName}{large}`
- SVG math copy/paste** For svg math, text copy/paste from the HTML `<alt>` tags lists the equation number or tag for single equations, along with the  $\text{\LaTeX}$  code for the math expression. For  $\mathcal{AMS}$  environments with multiple numbers in the same environment, only the first and last is copy/pasted, as a range. No tags are listed inside a starred  $\mathcal{AMS}$  environment, although the `\tag` macro will still appear inside the  $\text{\LaTeX}$  math expression.

### 8.5.3 MathJax option

- MathJax math option** The popular MathJax alternative ([mathjax.org](http://mathjax.org)) may be used to display math.

<sup>11</sup>See section 248 regarding fonts and fractions.

When MathJax is enabled, math is rendered twice:

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

### MathJax limitations

Prog MathJax

Limitations when using MathJax include:

#### 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/burnpanck/MathJax-siunitx>

Also see section 8.5.5.

⚠ `siunitx` inside an equation

⚠ other macros and packages

- Other math-related macros and packages are not supported by MathJax, including `\ensuremath`, `bigdelim`, `units`, and `nicefrac`, along with occasionally-used macros such as `\footnote` and `\relax`.

#### custom MathJax macros

- MathJax does not automatically support custom  $\text{\LaTeX}$  macros, but they may be created by the user inside a math expression:


```

\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}

```

#### 8.5.4 ntheorem package


 **Font control** This conversion is not total. Font control is via css, and the custom  $\LaTeX$  font settings are ignored.

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

#### 8.5.5 siunitx package

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

 **per-mode**

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

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

#### 8.5.6 units and nicefrac packages

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



### 8.5.7 newtxmath package

Pkg newtxmath The proper load order is:

⚠ loading sequence

```

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

## 8.6 Graphics

⚠ .pdf image files For `\includegraphics` with .pdf files, the user should provide a .pdf image file, and also a .svg, .png, or .jpg version of the same image. **These should be referred to without a file extension:**

⚠ no file extension

```
\includegraphics{filename} % print:.pdf, HTML:.svg or other
```

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

If a .pdf file is explicitly referred to with its file extension, a link to the .pdf file will appear in the HTML output.

```
\includegraphics{filename.pdf} % creates a link in HTML
```

other image files For .png, .jpg, or .gif image files, the same file may be used in both print or HTML versions, and may be used with a file extension, but will also be used without the file extension if it is the only file of its base name.

⚠ graphics vs. graphicx If using the older graphics syntax, use both optional arguments for `\includegraphics`. A single optional parameter is interpreted as the newer graphicx syntax. Note that viewports are not supported by warp; the entire image will be shown.

⚠ viewports

units For `\includegraphics`, avoid px and % units for width and height, or enclose them inside warpHTML environments. For font-proportional image sizes, use ex or em. For fixed-sized images, use cm, mm, in, pt, or pc. Use the keys `width=.5\linewidth`, or similar for `\textwidth` or `\textheight` to give fixed-sized images proportional to a 6 by 9 inch text area.

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 com-

bination, and if this is given then the HTML output will include that class for the image. The class is ignored for print 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.

### 8.6.1 grffile package

⚠ **matching PDF and SVG** `grffile` is supported as-is. File types known to the browser are displayed, and unknown file types are given a link. Each PDF image for print mode should be accompanied by an SVG, PNG, or JPG version for HTML.

### 8.6.2 color package

`color` is superseded by `xcolor`, and `lwarp` requires several of the features of `xcolor`.

⚠ **missing colors** It should be sufficient for the user's document to load `color` then load `xcolor` as well.

### 8.6.3 xcolor package

`\colorboxBlock` and `\fcolorboxBlock` `\colorboxBlock` and `\fcolorboxBlock` are provided for increased HTML compatibility, and they are identical to `\colorbox` and `\fcolorbox` in print mode. In HTML mode they place their contents into a `<div>` instead of a `<span>`. These `<div>`s are set to `display: inline-block` so adjacent `\colorboxBlock`s appear side-by-side in HTML, although text is placed before or after each.

Print-mode definitions for `\colorboxBlock` and `\fcolorboxBlock` are created by `lwarp`'s core if `xcolor` is loaded.

**background: none** `\fcolorbox` and `\fcolorboxBlock` allow a background color of `none`, in which case only the frame is drawn, which can be useful for HTML.

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

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

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

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

### 8.6.4 overpic package

- ⚠ **scaling** The macros `\overpicfontsize` and `\overpicfontskip` are used during HTML generation. These are sent to `\fontsize` to adjust the font size for scaling differences between the print and HTML versions of the document. Renew these macros before using the `overpic` and `Overpic` environments.

## 8.7 Tabular

Tabular mostly works as expected, but pay special attention to the following, especially if working with environments, macros inside tabulars, multirows, \* column specifiers, siunitx S columns, or the packages multirow, longtable, supertabular, or xtab.

### Defining environments:

⚠ **misplaced alignment  
alignment tab character &**

- When defining environments or macros which include tabular and instances of the `&` character, it may be necessary to make `&` active before the environment or macro is defined, then restore `&` to its default catcode after, using the following commands. These are ignored in print mode.

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

⚠ **tabular inside another  
environment**

- When creating a new environment which contains a tabular environment, lwarp's emulation of the tabular does not automatically resume when the containing environment ends, resulting in corrupted HTML rows. To fix this, use `\ResumeTabular` as follows. This is ignored in print mode.

```
\StartDefiningTabulars % because & is used in a
definition
\newenvironment{outerenvironment}
{
\tabular{cc}
left & right \\
}
{
\TabularMacro\ResumeTabular
left & right \\
\endtabular
}
\EndDefiningTabulars
```

### Cell contents:

### ⚠ paragraphs

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

### ⚠ `\multirow`

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

```
... & \multirow{2}{.5in}{text} & ...
... & \mrowcell & ...
```

### `vposn`

Note that recent versions of `\multirow` include a new optional `vposn` argument.

- The `\multirow` documentation regarding colored cells recommends using a negative number of rows. This will not work with `lwarp`, so `\warpprintonly` and `\warphTMLonly` must be used to make versions for print and HTML.
- See section 180.2 for `\multicolumnrow`.

### ⚠ `\multicolumn` & `\multirow`

`lwarp` does not support directly combining `\multicolumn` and `\multirow`. Use `\multicolumnrow` instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[Opt]{Text}
```

The two arguments for `\multicolumn` come first, followed by the five arguments for `\multirow`, many of which are optional, followed by the contents.

### ⚠ skipped cells

As per `\multirow`, skipped cells to the right of the `\multicolumnrow` statement are not included in the source code on the same line. On the following lines, `\mcolrowcell` must be used for each cell of each column and each row to be skipped:

```
... & \multicolumnrow{2}{c}[c]{3}[0]{1in}[Opt]{Text} & ...
... & \mcolrowcell & \mcolrowcell & ...
... & \mcolrowcell & \mcolrowcell & ...
```

### ⚠ empty cells

### `vposn`

Note that recent versions of `\multirow` include a new optional `vposn` argument.

### ⚠ macro in a table custom macros

- Using a custom macro inside a tabular data cell may result in an extra HTML data cell tag, corrupting the HTML table. To avoid this, use `\TabularMacro` just before the macro. This is ignored in print mode.

```
\TabularMacro\somemacro & more row contents \\
```

## Column specifiers:

### ⚠ \* column specification

- \* in a column specification is not used (so far). Repeat the column type the correct number of times.

### @ and !

- Only one each of @ and ! is used at each column, and they are used in that order.

`\multirow`

- In `\multirow` cells, the print version may have extra instances of `<`, `>`, `@`, and `!` cells on the second and later rows in the `\multirow` which do not appear in the HTML version.

⚠ `\newcolumnntype`

- `\newcolumnntype` is ignored; unknown column types are set to 1.

**Rules:**

vertical rules

- Vertical rules next to either side of an `@` or `!` column are displayed on both sides of the column.

width and trim

- Width options are honored. Trim options are converted to rounded top corners. Trim corners are not rounded with `@` or `!` columns, and full-width rules ignore trim.

full-width rules

- `\toprule`, `\midrule`, `\bottomrule`, and `\hline` ignore trim. When given an optional width, each cell is styled to create the custom border. Without an optional width, the entire row is given a class to assign the standard border.

combined rules

- If you wish to use `\cmidrule` followed by `\bottomrule`, it may be necessary to use:

```
\cmidrule{2-3} \[-2ex]
\bottomrule
```

The optional `-2ex` is ignored in HTML but improves the visual formatting in the print output.

⚠ `\warpprintonly`  
misplaced `\noalign`

- 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.

**Other:**

longtable headings

- `tabularx` ignores the width, but `X` columns do produce paragraph columns or multicolumns.

- For `longtable`, place headings and footings which do not apply to HTML inside `\warpprintonly{}`.

⚠ `S` columns

- For `S` columns (from the `siunitx` package), while producing print output, anything non-numeric must be placed inside `{ }` braces, including commands such as `\multirow`. While producing HTML output, though, anything placed inside braces is not seen by `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 \\\}
```

### 8.7.1 longtable package

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

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

instead of

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

Doing so helps avoid “Misplaced `\noalign`.” when using `\begin{warpprint}`.

Keep the `\endfirsthead` row, which is still relevant to HTML output.

⚠ `\kill` is ignored, place a `\kill` line inside

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

or place it inside `\warpingprintonly`.

### 8.7.2 supertabular and xtab packages

⚠ misplaced alignment  
alignment tab character &

For `\tablefirsthead`, etc., enclose them as follows:

```
\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars
```

See section 8.7.

### 8.7.3 bigdelim package

⚠ use `\mrowcell` `\ldelim` and `\rdelim` use `\multirow`, so `\mrowcell` must be used in the proper number of empty cells in the same column below `\ldelim` or `\rdelim`, but not in cells which are above or below the delimiter:

---

```

\begin{tabular}{lll}
<empty> & a & b \\
\ldelim{\}{2}{.25in}[left ] & c & d \\
\mrowcell & e & f \\
<empty> & g & h \\
\end{tabular}

```

---

```


      a   b
left { c   d
      e   f
      g   h

```

---

## 8.8 Floats

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

 **package conflicts** If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section [240.1](#).

### 8.8.2 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`.

### 8.8.3 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.

### 8.8.4 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.

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

⚠ `\FBwidth`, `\FBheight`

- 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.

### 8.8.5 keyfloat package

⚠ `keywrap`

If placing a `\keyfig[H]` inside a `keywrap`, use an absolute width for `\keyfig`, instead of `lw`-proportional widths. (The `[H]` option forces the use of a minipage, which internally adjusts for a virtual 6-inch wide minipage, which then corrupts the `lw` option.)

## 8.9 Miscellaneous

### 8.9.1 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}
```

---

`\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

`\leftmargini`

`\TMLvleftskip`

`\TMLleftmargini`



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.

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

### 8.9.2 newclude package

Pkg `newclude` `newclude` modifies `\label` in a non-adaptive way, so `newclude` must be loaded before `lwarp` is loaded.



loading

*Ex:*

---

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

---

### 8.9.3 babel package

Pkg `babel`

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

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


punctuation spaces Also when French is used, `lwarp` creates fixed-width space around punctuation by patching `\FBcolonspace`, `\FBthinspace`, `\FBguillspace`, `\FBmedkern`, `\FBthickkern`, `\FBtextellipsis`, and the tilde. If the user’s document also changes these parameters, the user’s changes should be placed inside a `warpprint` environment so that



customized spacing

the user’s changes do not affect the HTML output.

#### 8.9.4 **todonotes** package

- Pkg **todonotes** **todonotes** and **luatodonotes** are only emulated as if they were loaded with the `disable` option. All functions are disabled, but source compatibility is provided with the print version.
-  **xkeyvalue** **todonotes** and **luatodonotes** use `xkeyval`, so if the user's document uses `\presetkeys` then `xkeyval` will have to be loaded explicitly by the user's document.

## 9 EPUB conversion

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

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

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

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

To help convert lwarp HTML output to EPUB, add

```
\booltrue{FormatEPUB}
```

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

```
Enter ⇒ lwarpmk cleanall
```

```
Enter ⇒ lwarpmk html
```

```
Enter ⇒ lwarpmk limages
```

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

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

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

 **search order**

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

Preferences → Plugins → File type plugins → HTML to Zip

Check the box Add linked files in breadth first order.

### ⚠ section breaks

The EPUB-conversion program must also know where the section breaks are located. For a list of lwarp's section headings, see table 7. For example, an article class document would break at \section, which is mapped to HTML heading level <h4>, whereas a book class document would break at \chapter, which is HTML heading level <h3>. For Calibre, this option is found in

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

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

Match HTML tags with tag name:

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

//h:h4      — or —      //h:h3

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

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

### *MathJax support*

---

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

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

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

---

## 10 Word-processor conversion

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

### 10.1 Activating word-processor conversion

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

	<i>FormatWP</i>
Bool	FormatWP
	Default: false
	Changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments. Additionally, honors the booleans WPMarkFloats, WPMarkMinipages, WPMarkTOC, and WPMarkLOFT.

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

```
\booltrue{FormatWP}
```

formatting adjustments

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

- If using a class without chapters, \section and lower are shifted up in level for the HTML heading tags. The CSS has not been changed, so the section heading formats will not match the normal HTML output, but when imported to LibreOffice Writer the higher section headings will import as **Heading 1** for the title, **Heading 2** for \section, etc.
- Headers, footers, and navigation are removed at file splits.
- Any accumulated footnotes are printed at the bottom of each section.
- Forces single-file output.
- Turns off HTML debugging comments. These are comments appearing inside the HTML code, marking the opening/closing of sections and <div>s, but they are no longer useful when the document has been imported into a word processor.

- An additional `<div>` with an `id` encapsulates each float and minipage, which on import into LibreOffice Writer causes a thin frame to appear around the text block for each.
- Float captions are given an explicit italic formatting.
- Tabular rule borders are made explicit for LibreOffice Writer. LibreOffice displays a light border around each cell while editing, even those which have no border when printed, and lwarp also uses a light border for thin rules, so it will be best to judge the results using the print preview instead of while editing in LibreOffice.
- `\includegraphics` and `svg` math width and height are made explicit for LibreOffice.
- `\hspace` is approximated by a number of `\quads`, and rules are approximated by a number of underscores.
- Explicit HTML styles are given to:
  - `\textsc`, etc.
  - `\underline`, `soul` and `ulem` markup.
  - `center`, `flushleft`, `flushright`.
  - `\marginpar`, `keyfloat`, `sidenotes`, `floatflt`, and `wrapfig`.
  - `fancybox` `\shadowbox`, etc.
  - The  $\LaTeX$  and  $\TeX$  logos.

- Honors several booleans:

**WPMarkFloats:** Marks the begin and end of floats.

**WPMarkMinipages:** Marks the begin and end of minipages.

**WPMarkTOC:** Marks the location of the Table of Contents.

**WPMarkLOFT:** Marks the locations of the List of Figures/Tables.

**WPMarkMath:** Prints  $\LaTeX$  math instead of using images.

**WPTitleHeading:** Adjusts title and section headings.

Several of these may be used to add markers to the HTML text which help determine where to adjust the word processor document after import.

## 10.2 Additional modifications

---

### *WPMarkFloats*

---

Adds

```
=== begin table ===
...
=== end ===
```

or

```
=== begin figure ===
...
=== end ===
```

Bool WPMarkFloats  
Default: false

around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames and captions.

---



---

### *WPMarkMinipages*

---

Adds

```
=== begin minipage ===
...
=== end minipage ===
```

Bool WPMarkMinipages  
Default: false

around minipages while formatting for word processors. This helps identify boundaries of minipages to be manually converted to word-processor frames.

---



---

### *WPMarkTOC*

---

While formatting for word processors, adds

```
=== table of contents ===
```

Bool WPMarkTOC  
Default: true

where the Table of Contents would have been. This helps identify where to insert the actual toc.

*If set false, the actual toc is printed instead.*

---

---

*WPMarkLOFT*

---

While formatting for word processors, adds

```
=== list of figures ===    and/or
=== list of tables ===
```

Bool WPMarkLOFT  
Default: false

where each of these lists would have been. This helps identify where to insert the actual lists.

*If set false, the actual lists are printed instead.*

---



---

*WPMarkMath*

---

While formatting for word processors, prints math as  $\TeX$  code instead of creating svg images or MathJax. This is useful for cut/paste into the LibreOffice Writer TeXMaths extension.

Bool WPMarkMath  
Default: false  
Prog TeXMaths  
siunitx

When using the siunitx package, enter

```
\usepackage{siunitx}
```

in the TeXMaths preamble. Equation numbering is problematic for  $\mathcal{AMS}$  math environments.

---



---

*WPTitleHeading*

---

While formatting for word processors, true sets the document title to `<h1>`, which is expected for HTML documents, but also causes the lower-level section headings to start at **Heading 2** when imported into LibreOffice. Set to false to cause the title to be plain text, and the section headings to begin at **Heading 1**.

Bool WPTitleHeading  
Default: false  
section headings

See table 6 on page 113.

---

### 10.3 Recommendations

TOC, LOF, LOT For use with LibreOffice Writer, it is recommended to:

1. Set `\booltrue{FormatWP}`.
2. Set `\booltrue{WPMarkTOC}` and `\boolfalse{WPMarkLOFT}`.
3. Use lwarp to generate the HTML document.
4. Copy/paste from the HTML document into an empty LibreOffice Writer document.
5. Manually insert a LibreOffice TOC in the LibreOffice document.



Table 6: Section HTML headings for word-processor conversion

Section	HTML headings*			
	With <code>\chapter</code>		Without <code>\chapter</code>	
	WPTitleHeading		WPTitleHeading	
	true	false	true	false
Title	<code>&lt;h1&gt;</code>	plain	<code>&lt;h1&gt;</code>	plain
<code>\part</code>	<code>&lt;h2&gt;</code>	<code>&lt;h1&gt;</code>	<code>&lt;h2&gt;</code>	<code>&lt;h1&gt;</code>
<code>\chapter</code>	<code>&lt;h3&gt;</code>	<code>&lt;h2&gt;</code>	—	—
<code>\section</code>	<code>&lt;h4&gt;</code>	<code>&lt;h3&gt;</code>	<code>&lt;h3&gt;</code>	<code>&lt;h2&gt;</code>
<code>\subsection</code>	<code>&lt;h5&gt;</code>	<code>&lt;h4&gt;</code>	<code>&lt;h4&gt;</code>	<code>&lt;h3&gt;</code>
<code>\paragraph</code>	<code>&lt;h6&gt;</code>	<code>&lt;h5&gt;</code>	<code>&lt;h5&gt;</code>	<code>&lt;h4&gt;</code>
<code>\subparagraph</code>	span	<code>&lt;h6&gt;</code>	<code>&lt;h6&gt;</code>	<code>&lt;h5&gt;</code>

\* For default depths when not FormatWP, see table 7 on page 122.

6. Manually add frames around each float, adding a caption which is cut/pasted from each float's simulated caption.
7. Manually create cross references.

This process yields a document with an actual LibreOffice Table of Contents, but a simulated List of Figures and List of Tables.

[siunitx](#) For siunitx, remember to adjust the preamble as mentioned above.

[LO view border options](#) LibreOffice has options in the View menu to turn on/off the display of thin borders around table cells and text objects.

## 10.4 Limitations

Floats and captions are not explicitly converted to LibreOffice floats with their own captions. Floats are surrounded by a thin frame in the LibreOffice editor, and may be marked with `WPMarkFloats`, but are not given a proper LibreOffice object frame. Captions are given an explicit italic formatting, but not a proper LibreOffice paragraph style.

Cross references are not actual LibreOffice linked cross references.

The List of Figures and List of Tables are not linked. The pasted pseudo LOF and LOT match the numbering of the  $\text{\LaTeX}$  and HTML versions.

Equation numbering is not automatic, but the equation numbers in svg math will match the  $\text{\LaTeX}$  and HTML output. SVG math is recommended when using the  $\text{\LaTeX}$  environments, which may have multiple numbered equations per object.

As of when last checked, LibreOffice ignores the following:

- Minipage alignment.
- Tabular cell vertical alignment.
- Image rotation and scaling.
- Rounded border corners, which are also used by:
  - `\textcircled`
  - `booktabs trim`
- `\hspace` and rules, also used by algorithmic.
- Coloring of text decorations, used by `soul` and `ulem`.
- Overline text decoration, used by `romanbar`.

Libreoffice also has limitations with frames and backgrounds:

- Multiple lines in an object are framed individually instead of as a whole.
- Nested frames are not handled correctly.
- Images inside boxes are not framed correctly.
- Spans with background colors and frames are not displayed correctly.

## 11 Modifying lwarp

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

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

An environment may be converted to a `lateximage` then displayed with an image of the resulting  $\TeX$  output. See section 73 for an example of the `picture` environment.

To create a custom HTML block or inline CSS class, see section 40.8.

### 11.1 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 `mktexlsr` 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 189 (nttheorem) 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.

## 11.2 Adding a package to the `lwarp.dtx` file

When adding a package to `lwarp.dtx` for permanent including in `lwarp`, provide the `lwarp-<packagename>` code in `lwarp.dtx`, add its entry into `lwarp.ins`, and also remember to add

```
\LWR@loadafter{<packagename>}
```

to `lwarp.dtx` in section 21. This causes `lwarp` to stop with an error if `packagename` is loaded before `lwarp`.

## 11.3 Testing `lwarp`

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

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

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

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

## 11.4 Modifying `lwarpmk`

Prog `lwarpmk`  
File `lwarpmk.lua`

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

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

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

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

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

## 12 Troubleshooting

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

Also see:

Section 7.7: [Commands to be placed into the `warpprint` environment](#)

Section 8: [Special cases and limitations](#)

#### Text is not converting:

- Font-related UTF-8 information must be embedded in the PDF file. See section 7.1 regarding vector fonts.

#### Undefined HTML settings:

- See the warning regarding the placement of the HTML settings at section 7.3.

**Tabular problems:** See section 8.7.

#### Obscure error messages:

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

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

**“Improper `\prevdepth`”:** Something tried to use `\ensuremath` where `lwarp` then tries to create a `lateximage`. If you can locate what used `\ensuremath`, have it temporarily set:

```
\LetLtxMacro\@ensuredmath\LWR@origensuredmath
inside a group first.
```

#### Missing sections:

- See section 7.3 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 7.3.

#### Missing / incorrect cross-references:

- Use `lwarpmk` again followed by `lwarpmk html` or `lwarpmk print` to compile the document one more time.

## labels

⚠ underscores

\nameref

⚠ empty link

- Labels with special characters may be a problem. It is best to stick with alpha-numeric, hyphen, and perhaps the colon (if not French). The underscore currently does not yet work when generating math for MathJax.

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

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

Ex:

```
\cpageref{tab:first,tab:second}
```

in HTML becomes:

“pages **for** table 4.1 and **for** table 4.2”

See \cpagerefFor at page 394 to redefine the message which is printed for page number references.

⚠ cleveref page numbers

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

Use Xe<sub>La</sub>TeX or Lua<sub>La</sub>TeX.

**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 [warppHTML].

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

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

**Images are appearing in strange places:**

- lwarpmk limages to refresh the lateximage images.

**SVG images:**

⚠ Adding/removing

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

⚠ HTML instead of images

If HTML appears where an svg image should be, recompile the document one

more time to get the page numbers back in sync, then remake the images one more time.

⚠ Lots of files!

Expressing math as SVG images has the advantage of representing the math exactly as  $\TeX$  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 6.5 to use MathJax instead.

#### Plain-looking document:

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

#### Broken fragments of HTML:

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

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

- Be sure to `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 7.3.
- 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.

### 12.1.1 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.



## 12.2 Compiling the `lwarp.dtx` file

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

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

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

### Un-nested environments:

Be sure to properly nest:

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

File 1 **lwarp.sty**

## 13 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 7: Section depths and HTML headings

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

\* If FormatWP is true, section headings may be adjusted, depending on WPTitleHeading. See table 6 on page 113.

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

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

## 14 Section depths and HTML headings

Stacks are created to track depth inside the  $\LaTeX$  document structure. This depth is translated to HTML headings as shown in table 7. “Depth” here is not depth in the traditional computer-science stack-usage sense, but rather a representation of the nesting depth inside the  $\LaTeX$  document structure.

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

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

It is possible to use HTML5 `<section>` and `<h1>` for all levels, but this may not be well-recognized by older browsers.

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

## 15 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

**index entries** command macros. Each of these also appears in the index as individual entries, and are also listed together under “files”, “packages”, “environments”, “booleans”, and “counters”.

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

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

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

```

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

```

## 17 Unicode Input Characters

for HTML & PRINT:

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

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

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

```

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

```

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

```

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

```

## 18 Early package requirements

Pkg etoolbox Provides \ifbool and other functions.

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

Pkg ifplatform Provides \ifwindows to try to automatically detect Windows OS.

```
24 \RequirePackage{ifplatform}% sense op-system platform
```

Pkg letltxmacro Used to redefine \textbf and friends.

```
25 \RequirePackage{letltxmacro}
```

## 19 Operating-System portability

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

### 19.1 Common portability code

Bool usingOSWindows Set if the OSWindows option is used.

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

### 19.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

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

### 19.3 MS-Windows

For MS-Windows:

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

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

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

```
34 \ifwindows
35 \LWR@setOSWindows
36 \fi
```

## 20 Package options

Pkg `kvoptions` Allows key/value package options.

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

Bool `warpingprint`

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

Bool `mathjax`

```
39 \newbool{warpingprint}
40 \newbool{warpingHTML}
41 \newbool{mathjax}
```

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

```
42 \booltrue{warpingprint}%
43 \boolfalse{warpingHTML}%
44 \boolfalse{mathjax}%
```

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

```
45 \DeclareVoidOption{warpprint}{%
46 \PackageInfo{lwarp}{Using option 'warpprint'}
47 \booltrue{warpingprint}%
48 \boolfalse{warpingHTML}%
49 }
```

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

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

```
50 \DeclareVoidOption{warpHTML}{%
51 \PackageInfo{lwarp}{Using option 'warpHTML'}%
```



```
52 \booltrue{warpingHTML}%
53 \boolfalse{warpingprint}%
54 }
```

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

```
55 \DeclareVoidOption{mathsvg}{%
56 \PackageInfo{lwarp}{Using option 'mathsvg'}
57 \boolfalse{mathjax}%
58 }
```

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

```
59 \DeclareVoidOption{mathjax}{%
60 \PackageInfo{lwarp}{Using option 'mathjax'}
61 \booltrue{mathjax}%
62 }
```

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

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

```
63 \DeclareStringOption[\jobname]{BaseJobname}
```

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

```
64 \DeclareStringOption[english]{IndexLanguage}
```

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

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

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

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

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

```

66 \newbool{LWR@creatinglwarpmk}
67 \boolfalse{LWR@creatinglwarpmk}
68
69 \DeclareVoidOption{lwarpmk}{
70 \PackageInfo{lwarp}{Using option 'lwarpmk'}}
71 \booltrue{LWR@creatinglwarpmk}
72 }

```

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

```

73 \DeclareVoidOption{OSWindows}{
74 \PackageInfo{lwarp}{Using option 'OSWindows'}}
75 \LWR@setOSWindows
76 }

```

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

Default: \lwarp

```

77 \DeclareStringOption[] {HomeHTMLFilename}

```

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

Default: <empty>

```

78 \DeclareStringOption[] {HTMLFilename}

```

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

```

79 \DeclareBoolOption[false]{latexmk}

```

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

```

80 \ProcessKeyvalOptions*\relax

```

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

```

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

```

Defaults unless already over-ridden by the user:

```

82 \ifcempty{LWR@HomeHTMLFilename}{
83 \newcommand*{\HomeHTMLFilename}{\BaseJobname}

```

```

84 }{
85 \csedef{HomeHTMLFilename}{\LWR@HomeHTMLFilename}
86 }
87
88 \csedef{HTMLFilename}{\LWR@HTMLFilename}

```

## 20.1 Conditional compilation

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

Only process the contents if producing printed output.

```
89 \newcommand{\warpprintonly}[1]{\ifbool{warpingprint}{#1}{}}
```

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

Only process the contents if producing HTML output.

```
90 \newcommand{\warpHTMLonly}[1]{\ifbool{warpingHTML}{#1}{}}
```

`Pkg` `comment` Provides conditional code blocks.

```
91 \RequirePackage{comment}
```

Use `comment_print.cut` for print mode, and `comment_html.cut` for HTML mode. This helps `latexmk` to more reliably know whether to recompile.

```

92 \ifbool{warpingHTML}{
93 \def\DefaultCutFileName{\def\CommentCutFile{comment_html.cut}}
94 }{}
95
96 \ifbool{warpingprint}{
97 \def\DefaultCutFileName{\def\CommentCutFile{comment_print.cut}}
98 }{}

```

```
99 \excludecomment{testing}
```

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

```
100 \includecomment{warpall}
```

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

`Env` `warpHTML`

For HTML output:

```
101 \ifbool{warpingHTML}{%
102 \includecomment{warpHTML}
103 }
104 {\excludecomment{warpHTML}}%
```

```
105 \ifbool{warpingprint}
106 {\includecomment{warpprint}}
107 {\excludecomment{warpprint}}
```

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

```
108 \ifbool{LWR@creatinglwarpmk}
109 {\includecomment{LWR@createlwarpmk}}
110 {\excludecomment{LWR@createlwarpmk}}
```

## 21 Misplaced packages

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

Packages which should only be loaded before lwarp have their own

```
lwarp-<packagename>.sty
```

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

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

```
111 \newcommand*\LWR@loadafter}[1]{%
112 \@ifpackageloaded{#1}
113 {
114 \PackageError{lwarp}
115 {Package #1, or one which uses #1, must be loaded after lwarp}
116 {Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{lwarp}.
117 Package #1 may also be loaded by something else, which must also be moved
118 after lwarp.}
119 }
120 {}
121 }
```

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

```

122 \newcommand*{\LWR@loadbefore}[1]{%
123 \@ifpackageloaded{#1}
124 {}
125 {
126 \PackageError{lwarp}
127 {Package #1 must be loaded before lwarp}
128 {Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{lwarp}.}
129 }
130 }

```

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

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

```

131 \newcommand*{\LWR@loadnever}[2]{%
132 \PackageError{lwarp}
133 {Package #1 is not supported by lwarp's HTML conversion.
134 Package(s) #2 may be useful instead}
135 {Package #1 might conflict with lwarp in some way,
136 or is superceded by another package.
137 For a possible alternative, see package(s) #2.}
138 }

```

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:

```

139 \LWR@loadafter{a4}
140 \LWR@loadafter{a4wide}
141 \LWR@loadafter{a5comb}
142 \LWR@loadafter{abstract}
143 \LWR@loadafter{adjmulticol}
144 \LWR@loadafter{addlines}
145 \LWR@loadafter{afterpage}
146 \LWR@loadafter{algorithmicx}
147 \LWR@loadafter{alltt}
148 \LWR@loadafter{amsthm}
149 \LWR@loadafter{anonchap}
150 \LWR@loadafter{anysize}
151 \LWR@loadafter{appendix}
152 \LWR@loadafter{arabicfront}
153 \LWR@loadafter{array}
154 \LWR@loadafter{authblk}
155 \LWR@loadafter{balance}
156 \LWR@loadafter{bigdelim}
157 \LWR@loadafter{bigstrut}
158 \LWR@loadafter{bookmark}

```

```
159 \LWR@loadafter{booktabs}
160 \LWR@loadafter{boxedminipage}
161 \LWR@loadafter{boxedminipage2e}
162 \LWR@loadafter{breakurl}
163 \LWR@loadafter{ccaption}
164 \LWR@loadafter{changepage}
165 \LWR@loadafter{chnngpage}
166 \LWR@loadafter{chappg}
167 \LWR@loadafter{color}
168 \LWR@loadafter{crop}
169 \LWR@loadafter{cuted}
170 \LWR@loadafter{cutwin}
171 \LWR@loadafter{dblfnote}
172 \LWR@loadafter{dcolumn}
173 \LWR@loadafter{draftwatermark}
174 \LWR@loadafter{ebook}
175 \LWR@loadafter{ellipsis}
176 \LWR@loadafter{emptypage}
177 \LWR@loadafter{enumerate}
178 \LWR@loadafter{enumitem}
179 \LWR@loadafter{epigraph}
180 \LWR@loadafter{eso-pic}
181 \LWR@loadafter{everypage}
182 \LWR@loadafter{everyshi}
183 \LWR@loadafter{extramarks}
184 \LWR@loadafter{fancybox}
185 \LWR@loadafter{fancyhdr}
186 \LWR@loadafter{fancyvrb}
187 \LWR@loadafter{figcaps}
188 \LWR@loadafter{fix2col}
189 \LWR@loadafter{float}
190 \LWR@loadafter{floatflt}
191 \LWR@loadafter{floatpag}
192 \LWR@loadafter{floatrow}
193 \LWR@loadafter{flushend}
194 \LWR@loadafter{fncychap}
195 \LWR@loadafter{fnpos}
196 % fontenc must be loaded before lwarp
197 % fontspec must be loaded before lwarp
198 \LWR@loadafter{ftnright}
199 \LWR@loadafter{fullpage}
200 \LWR@loadafter{fullwidth}
201 \LWR@loadafter{geometry}
202 \LWR@loadafter{glossaries}
203 % \LWR@loadafter{graphics}% pre-loaded by xunicode
204 % \LWR@loadafter{graphicx}% pre-loaded by xunicode
205 \LWR@loadafter{grffile}
206 \LWR@loadafter{hyperref}
207 \LWR@loadafter{hyperxmp}
208 \LWR@loadafter{idxlayout}
```

```
209 \LWR@loadafter{indentfirst}
210 % inputenc must be loaded before lwarp
211 \LWR@loadafter{keyfloat}
212 \LWR@loadafter{layout}
213 \LWR@loadafter{letterspace}
214 \LWR@loadafter{lettrine}
215 \LWR@loadafter{lips}
216 \LWR@loadafter{listings}
217 \LWR@loadafter{longtable}
218 \LWR@loadafter{lscapex}
219 \LWR@loadafter{ltcaption}
220 \LWR@loadafter{luatodonotes}
221 \LWR@loadafter{marginfit}
222 \LWR@loadafter{marginfix}
223 \LWR@loadafter{marginnote}
224 \LWR@loadafter{mcaption}
225 \LWR@loadafter{mdframed}
226 \LWR@loadafter{metalogo}
227 \LWR@loadafter{microtype}
228 \LWR@loadafter{midfloat}
229 \LWR@loadafter{moreverb}
230 % morewrites must be loaded before lwarp
231 \LWR@loadafter{mparhack}
232 % \LWR@loadafter{multicol}% loaded by ltxdoc
233 \LWR@loadafter{multirow}
234 \LWR@loadafter{nameref}
235 \LWR@loadafter{needspace}
236 % newclude must be loaded before lwarp
237 \LWR@loadafter{newtxmath}
238 % newunicodechar must be loaded before lwarp
239 \LWR@loadafter{nextpage}
240 \LWR@loadafter{nonumonpart}
241 \LWR@loadafter{nopageno}
242 \LWR@loadafter{nowidow}
243 \LWR@loadafter{ntheorem}
244 \LWR@loadafter{overpic}
245 \LWR@loadafter{pagenote}
246 \LWR@loadafter{paralist}
247 \LWR@loadafter{parskip}
248 \LWR@loadafter{pdfscape}
249 \LWR@loadafter{pdfsync}
250 \LWR@loadafter{pfnote}
251 \LWR@loadafter{placeins}
252 \LWR@loadafter{prelim2e}
253 \LWR@loadafter{quotchap}
254 \LWR@loadafter{ragged2e}
255 \LWR@loadafter{realscripts}
256 \LWR@loadafter{relsize}
257 \LWR@loadafter{romanbar}
258 \LWR@loadafter{romanbarpagenumber}
```

```
259 \LWR@loadafter{rotating}
260 \LWR@loadafter{rotfloat}
261 \LWR@loadafter{savetrees}
262 % \LWR@loadafter{scalegnt}% loaded by babel-french
263 \LWR@loadafter{sectsty}
264 \LWR@loadafter{setspace}
265 \LWR@loadafter{shadow}
266 \LWR@loadafter{showidx}
267 \LWR@loadafter{showkeys}
268 \LWR@loadafter{sidecap}
269 \LWR@loadafter{sidenotes}
270 \LWR@loadafter{siunitx}
271 \LWR@loadafter{soul}
272 \LWR@loadafter{stabular}
273 \LWR@loadafter{subfig}
274 \LWR@loadafter{supertabular}
275 \LWR@loadafter{tabls}
276 \LWR@loadafter{tabularx}
277 \LWR@loadafter{tabulary}
278 \LWR@loadafter{textarea}
279 % \LWR@loadafter{textcomp}% maybe before lwarp with font packages
280 \LWR@loadafter{textpos}
281 \LWR@loadafter{theorem}
282 \LWR@loadafter{threeparttable}
283 \LWR@loadafter{tikz}
284 \LWR@loadafter{titleps}
285 \LWR@loadafter{titlesec}
286 \LWR@loadafter{titletoc}
287 \LWR@loadafter{titling}
288 \LWR@loadafter{tocbibind}
289 \LWR@loadafter{tocloft}
290 \LWR@loadafter{todonotes}
291 \LWR@loadafter{transparent}
292 \LWR@loadafter{trivfloat}
293 \LWR@loadafter{typearea}
294 \LWR@loadafter{ulem}
295 \LWR@loadafter{upref}
296 \LWR@loadafter{varioref}
297 \LWR@loadafter{verse}
298 \LWR@loadafter{wallpaper}
299 \LWR@loadafter{wrapfig}
300 \LWR@loadafter{xcolor}
301 \LWR@loadafter{xfrac}
302 \LWR@loadafter{xltextra}
303 \LWR@loadafter{xmpinl}
304 \LWR@loadafter{xtab}
305 \LWR@loadafter{zwpagelayout}
```



## 22 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:** 306 \begin{warppall}

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

Detects Xe<sub>La</sub>TeX and Lua<sub>La</sub>TeX:

```
307 \RequirePackage{iftex}
308 \newif\ifxetexorluatex
309 \ifXeTeX
310   \xetexorluatextrue
311 \else
312   \ifLuaTeX
313     \xetexorluatextrue
314   \else
315     \xetexorluatexfalse
316   \fi
317 \fi
```

```
318 \end{warppall}
```

**for HTML output:** 319 \begin{warpHTML}

```
320 \ifxetexorluatex
321 % ^^A \usepackage[no-math]{fontspec}
```

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

```
322 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
323 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
324 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
325 \else
```

**pdf<sub>La</sub>TeX only:** Only pre-loaded if pdf<sub>La</sub>TeX is being used.

Pkg microtype

**ligatures** Older browsers don't display ligatures. Turn off letter ligatures, keeping  $\TeX$  dash and quote ligatures, which may fail on older browsers but at least won't corrupt written words.

```

326 \RequirePackage {microtype}
327
328 \microtypesetup{
329   protrusion=false,
330   expansion=false,
331   tracking=false,
332   kerning=false,
333   spacing=false}
334
335 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}

336 \fi

337 \end{warpHTML}

```

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

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

**for HTML output:**

```

338 \begin{warpHTML}
339 \RequirePackage[paperheight=190in,paperwidth=20in,%
340 left=2in,right=12in,%
341 top=1in,bottom=1in,%
342 ]{geometry}
343 \@twosidefalse
344 \@mparswitchfalse
345 \end{warpHTML}

```

**for HTML & PRINT:**

```

346 \begin{warpall}

```

**Pkg xparse**

$\TeX$ 3 command argument parsing

```

347 \RequirePackage{xparse}

```

Pkg afterpackage Used to patch titling to add \AddSubtitlePublished.

```
348 \RequirePackage{afterpackage}
```

```
349 \end{warpall}
```

for HTML output: 350 \begin{warpHTML}

Pkg expl3

ℳℒX3 programming

```
351 \RequirePackage{expl3}
```

Pkg gettitlestring

Used to emulate \nameref.

```
352 \RequirePackage{gettitlestring}
```

Pkg everyhook

everyhook is used to patch paragraph handling.

```
353 \RequirePackage{everyhook}
```

```
354 \end{warpHTML}
```

for HTML & PRINT: 355 \begin{warpall}

Pkg filecontents

Used to write helper files, done in print mode.

Patched to work with morewrites, per <https://tex.stackexchange.com/questions/312830/does-morewrites-not-support-filecontents-and-can-i-write-body-of-environment-us/312910>

```
356 \RequirePackage{filecontents}
```

```
357
```

```
358 \@ifpackagelater{filecontents}{2011/10/09}%
```

```
359 {}
```

```
360 {
```

```
361 \newwrite\fcwrite
```

```
362 \let\LWR@origfilec@ntents\filec@ntents
```

```
363 \def\filec@ntents{\def\chardef##1\write{\let\reserved@c\fcwrite}\LWR@origfilec@ntents}
```

```
364 }
```

```
365 \end{warpall}
```

**for HTML output:** 366 \begin{warpHTML}

Pkg xifthen

367 \RequirePackage{xifthen}

Pkg xstring

368 \RequirePackage{xstring}

Pkg xstring

369 \RequirePackage{verbatim}

Pkg makeidx

370 \RequirePackage{makeidx}

371 \makeindex

Pkg calc

372 \RequirePackage{calc}

Pkg refcount

373 \RequirePackage{refcount}

Pkg newfloat

374 \RequirePackage{newfloat}

Pkg caption

375 \RequirePackage{caption}

376 \end{warpHTML}

**for HTML & PRINT:** 377 \begin{warpall}

Pkg environ

Used to encapsulate math environments for re-use in HTML <alt> text.

378 \RequirePackage{environ}

379 \end{warpall}

**for HTML output:** 380 `\begin{warpHTML}`

Pkg    `zref`

Used for cross-references.

381 `\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.

382 `\PassOptionsToPackage{leqno}{amsmath}`

383 `\RequirePackage{amsmath}`

Used to convert lengths for image width/height options.

384 `\RequirePackage{printlen}`

385 `\end{warpHTML}`

## 23 Loading packages

**for HTML output:** 386 `\begin{warpHTML}`

Remember the original `\RequirePackage`:

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

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

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

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

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

389 `\newcommand*\LWR@findword[3][,]{%`

390     `\StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%`

391 }

`\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.

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

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

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

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

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

See if the package name was found:

```
395 \IfStrEq{\LWR@strresulttwo}{}%
396 {}% no filename
397 {}% yes filename
```

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

```
398 \IfFileExists{lwarp-\LWR@strresulttwo.sty}%
399 {}% lwarp-* file found
400     \StrSubstitute%
401         {\LWR@requirepackagenames}%
402         {\LWR@strresult}%
403         {lwarp-\LWR@strresulttwo}[\LWR@requirepackagenames]%
404 }%
405 {}% no lwarp-* file
406 }% yes filename
407 }
```

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

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

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

Redirect up to nine names:

```
409 \renewcommand*{\LWR@requirepackagenames}{#2}
410 \LWR@lookforpackagename{1}
411 \LWR@lookforpackagename{2}
412 \LWR@lookforpackagename{3}
413 \LWR@lookforpackagename{4}
```

```

414 \LWR@lookforpackagename{5}
415 \LWR@lookforpackagename{6}
416 \LWR@lookforpackagename{7}
417 \LWR@lookforpackagename{8}
418 \LWR@lookforpackagename{9}

```

\RequirePackage depending on the options and version:

```

419 \IfValueTF{#1}
420 {% options given
421   \IfValueTF{#3}% version given?
422   {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}[#3]}
423   {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}
424 }
425 {% no options given
426   \IfValueTF{#3}% version given?
427   {\LWR@origRequirePackage{\LWR@requirepackagenames}[#3]}
428   {\LWR@origRequirePackage{\LWR@requirepackagenames}}
429 }
430 }
431 \LetLtxMacro{\usepackage}{\RequirePackage}

```

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

Uses the original package, including options.

```

432 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
433 \PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
434 \IfValueTF{#2}
435 {\ProvidesPackage{lwarp-#1}[#2]}
436 {\ProvidesPackage{lwarp-#1}}
437 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
438 \ProcessOptions\relax
439
440 \IfValueTF{#2}
441 {\LWR@origRequirePackage{#1}[#2]}
442 {\LWR@origRequirePackage{#1}}
443 }

```

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

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

```

444 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
445 \PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
446 \IfValueTF{#2}
447 {\ProvidesPackage{lwarp-#1}[#2]}

```

```

448 {\ProvidesPackage{lwarp-#1}}
449 \DeclareOption*{}
450 \ProcessOptions\relax
451 }

452 \end{warpHTML}

```

## 24 File handles

Defines file handles for writes.

**for HTML & PRINT:** 453 \begin{warpall}

\LWR@quickfile For quick temporary use only. This is reused in several places.

```

454 \newwrite\LWR@quickfile%

455 \end{warpall}

```

**for HTML output:** 456 \begin{warpHTML}

\LWR@lateximagesfile For lateximages.txt.

```

457 \newwrite\LWR@lateximagesfile

458 \end{warpHTML}

```

## 25 Include a file

During HTML output, \include{<filename>} causes the following to occur:

1. lwarp creates <filename>\_html\_inc.tex whose contents are:
 

```
\input <filename>.tex
```
2. <filename>\_html\_inc.tex is then \included instead of <filename>.tex.
3. <filename>\_html\_inc.aux is automatically generated and used by  $\LaTeX$ .

**for HTML output:** 459 \begin{warpHTML}



`\include {<filename>}`

`\@include {<filename>}` Modified to load `_html_inc` files.

```

460 \def\@include#1 {%
461 \immediate\openout\LWR@quickfile #1_html_inc.tex% new
462 \immediate\write\LWR@quickfile{\string\input{#1.tex}}% new
463 \immediate\closeout\LWR@quickfile% new
464 \LWR@origclearpage% \changed
465 \if@files
466     \immediate\write\@mainaux{\string\@input{#1_html_inc.aux}}% changed
467 \fi
468 \@tempwtrue
469 \if@partsw
470     \@tempwfalse
471     \edef\reserved@b{#1}%
472     \@for\reserved@a:=\@partlist\do
473     {\ifx\reserved@a\reserved@b\@tempwtrue\fi}%
474 \fi
475 \if@tempsw
476     \let\@auxout\@partaux
477     \if@files
478         \immediate\openout\@partaux #1_html_inc.aux % changed
479         \immediate\write\@partaux{\relax}%
480     \fi
481     \@input{#1_html_inc.tex}% changed
482     \LWR@origclearpage% changed
483     \@writeckpt{#1}%
484     \if@files
485         \immediate\closeout\@partaux
486     \fi
487 \else
488     \deadcycles\z@
489     \@nameuse{cp@#1}%
490 \fi
491 \let\@auxout\@mainaux%
492 }

493 \end{warpHTML}

```

## 26 Copying a file

for HTML output: `494 \begin{warpHTML}`

`\LWR@copyfile {<source filename>} {<destination filename>}`

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

```

495 \newwrite\LWR@copyoutfile % open the file to write to
496 \newread\LWR@copyinfile % open the file to read from
497
498 \newcommand*{\LWR@copyfile}[2]{%
499 \immediate\openout\LWR@copyoutfile=#2
500 \openin\LWR@copyinfile=#1
501 \begingroup\endlinechar=-1
502 \makeatletter
503 \loop\unless\ifeof\LWR@copyinfile
504 \read\LWR@copyinfile to\LWR@fileline % Read one line and store it into \LWR@fileline
505 % \LWR@fileline\par % print the content into the pdf
506 % print the content:
507 \immediate\write\LWR@copyoutfile{\unexpanded\expandafter{\LWR@fileline}}%
508 \repeat
509 \closeout\LWR@copyoutfile
510 \endgroup
511 }

512 \end{warpHTML}

```

## 27 Debugging messages

**for HTML & PRINT:** 513 \begin{warpall}

Bool LWR@tracinglwarp True if tracing is turned on.

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

\tracinglwarp Turns on the debug tracing messages.

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

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

```

516 \newcommand{\LWR@traceinfo}[1]{%
517 \ifbool{LWR@tracinglwarp}%
518 {%
519 \typeout{*** lwarp: #1}%
520 % \PackageInfo{lwarp}{#1 : }%
521 }%
522 {}%
523 }

```

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

Default: false

```
524 \newbool{HTMLDebugComments}
525 \boolfalse{HTMLDebugComments}
```

If \tracinglwarp, show where preamble hooks occur:

```
526 \AfterEndPreamble{
527 \LWR@traceinfo{AfterEndPreamble}
528 }
529
530 \AtBeginDocument{
531 \LWR@traceinfo{AtBeginDocument}
532 }

533 \end{warpall}
```

## 28 HTML-conversion output modifications

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

for HTML & PRINT: 534 \begin{warpall}

### 28.1 User-level controls

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

Default: false

```
535 \newbool{FormatEPUB}
536 \boolfalse{FormatEPUB}
```

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

Default: false

```
537 \newbool{FormatWP}
538 \boolfalse{FormatWP}
```

Bool WPMarkFloats Adds

Default: false

```
=== begin table ===
...
=== end ===
```

or

```
=== begin figure ===
...
=== end ===
```

around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames and captions.<sup>12</sup>

```
539 \newbool{WPMarkFloats}
540 \boolfalse{WPMarkFloats}
```

Bool WPMarkMinipages Adds

Default: false

```
=== begin minipage ===
...
=== end minipage ===
```

around minipages while formatting for word processors. This helps identify boundaries of minipages to be manually converted to word-processor frames.

```
541 \newbool{WPMarkMinipages}
542 \boolfalse{WPMarkMinipages}
```

Bool WPMarkTOC While formatting for word processors, adds

Default: true

```
=== table of contents ===
```

where the Table of Contents would have been. This helps identify where to insert the actual TOC.

*If set false, the actual TOC is printed instead.*

```
543 \newbool{WPMarkTOC}
544 \booltrue{WPMarkTOC}
```

Bool WPMarkLOFT While formatting for word processors, adds

Default: false

```
=== list of figures === and/or
=== list of tables ===
```

where each of these lists would have been. This helps identify where to insert the actual lists.

*If set false, the actual lists are printed instead.*

```
545 \newbool{WPMarkLOFT}
```

---

<sup>12</sup>Perhaps some day word processors will have HTML import options for identifying <figure> and <figcaption> tags for figures and tables.

```
546 \boolfalse{WPMarkLOFT}
```

Bool WPMarkMath While formatting for word processors, prints math as  $\LaTeX$  code instead of creating SVG images or MathJax. This is useful for cut/paste into the LibreOffice Writer TeXMaths extension.

Default: false

```
547 \newbool{WPMarkMath}
548 \boolfalse{WPMarkMath}
```

Bool WPTitleHeading While formatting for word processors, true sets the document title to `<h1>`, which is expected for HTML documents, but also causes the lower-level section headings to start at **Heading 2** when imported into LibreOffice. Set to false to cause the title to be plain text, and the section headings to begin at **Heading 1**.

Default: false

See table 6 on page 113.

```
549 \newbool{WPTitleHeading}
550 \boolfalse{WPTitleHeading}
```

```
551 \end{warpall}
```

## 28.2 Heading adjustments

If formatting the HTML for a word processor, adjust heading levels.

If WPTitleHeading is true, adjust so that part is **Heading 1**.

If WPTitleHeading is false, use `<h1>` for the title, and set part to **Heading 2**.

for HTML output: 552 \begin{warpHTML}

```
553 \AtBeginDocument{
554 \ifbool{FormatWP}{
555 \@ifundefined{chapter}{
556 \ifbool{WPTitleHeading}{% part and section starting at h2
557 \renewcommand*{\LWR@tagtitle}{h1}
558 \renewcommand*{\LWR@tagtitleend}{/h1}
559 \renewcommand*{\LWR@tagpart}{h2}
560 \renewcommand*{\LWR@tagpartend}{/h2}
561 \renewcommand*{\LWR@tagsection}{h3}
562 \renewcommand*{\LWR@tagsectionend}{/h3}
563 \renewcommand*{\LWR@tagsubsection}{h4}
564 \renewcommand*{\LWR@tagsubsectionend}{/h4}
565 \renewcommand*{\LWR@tagsubsubsection}{h5}
566 \renewcommand*{\LWR@tagsubsubsectionend}{/h5}
567 \renewcommand*{\LWR@tagparagraph}{h6}
568 \renewcommand*{\LWR@tagparagraphend}{/h6}
```

```

569 \renewcommand*{\LWR@tagsubparagraph}{span class="subparagraph">{}}
570 \renewcommand*{\LWR@tagsubparagraphend}{/span}
571 }% WPTitleHeading
572 {% not WPTitleHeading, part and section starting at h1
573 \renewcommand*{\LWR@tagtitle}{div class="title"}
574 \renewcommand*{\LWR@tagtitleend}{/div}
575 \renewcommand*{\LWR@tagpart}{h1}
576 \renewcommand*{\LWR@tagpartend}{/h1}
577 \renewcommand*{\LWR@tagsection}{h2}
578 \renewcommand*{\LWR@tagsectionend}{/h2}
579 \renewcommand*{\LWR@tagsubsection}{h3}
580 \renewcommand*{\LWR@tagsubsectionend}{/h3}
581 \renewcommand*{\LWR@tagsubsubsection}{h4}
582 \renewcommand*{\LWR@tagsubsubsectionend}{/h4}
583 \renewcommand*{\LWR@tagparagraph}{h5}
584 \renewcommand*{\LWR@tagparagraphend}{/h5}
585 \renewcommand*{\LWR@tagsubparagraph}{h6}
586 \renewcommand*{\LWR@tagsubparagraphend}{/h6}
587 }% not WPTitleHeading
588 }% chapter undefined
589 {% chapter defined
590 \ifbool{WPTitleHeading}{
591 {% not WPTitleHeading, part and chapter starting at h1
592 \renewcommand*{\LWR@tagtitle}{div class="title"}
593 \renewcommand*{\LWR@tagtitleend}{/div}
594 \renewcommand*{\LWR@tagpart}{h1}
595 \renewcommand*{\LWR@tagpartend}{/h1}
596 \renewcommand*{\LWR@tagchapter}{h2}
597 \renewcommand*{\LWR@tagchapterend}{/h2}
598 \renewcommand*{\LWR@tagsection}{h3}
599 \renewcommand*{\LWR@tagsectionend}{/h3}
600 \renewcommand*{\LWR@tagsubsection}{h4}
601 \renewcommand*{\LWR@tagsubsectionend}{/h4}
602 \renewcommand*{\LWR@tagsubsubsection}{h5}
603 \renewcommand*{\LWR@tagsubsubsectionend}{/h5}
604 \renewcommand*{\LWR@tagparagraph}{h6}
605 \renewcommand*{\LWR@tagparagraphend}{/h6}
606 \renewcommand*{\LWR@tagsubparagraph}{span class="subparagraph">{}}
607 \renewcommand*{\LWR@tagsubparagraphend}{/span}
608 }% not WPTitleHeading
609 }% chapter defined
610 }{}% FormatWP
611 }% AtBeginDocument

612 \end{warpHTML}

```

## 29 Remembering original formatting macros

for HTML output: 613 \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.

```

614 \LetLtxMacro{\LWR@origtextrm}{\textrm}
615 \LetLtxMacro{\LWR@origtextsf}{\textsf}
616 \LetLtxMacro{\LWR@origtexttt}{\texttt}
617 \LetLtxMacro{\LWR@origtextnormal}{\textnormal}
618 \LetLtxMacro{\LWR@origtextbf}{\textbf}
619 \LetLtxMacro{\LWR@origtextmd}{\textmd}
620 \LetLtxMacro{\LWR@origtextit}{\textit}
621 \LetLtxMacro{\LWR@origtextsl}{\textsl}
622 \LetLtxMacro{\LWR@origtextsc}{\textsc}
623 \LetLtxMacro{\LWR@origtextup}{\textup}
624 \LetLtxMacro{\LWR@origemph}{\emph}
625
626 \LetLtxMacro{\LWR@origrmfamily}{\rmfamily}
627 \LetLtxMacro{\LWR@origsfamily}{\sffamily}
628 \LetLtxMacro{\LWR@origttfamily}{\ttfamily}
629 \LetLtxMacro{\LWR@origbfseries}{\bfseries}
630 \LetLtxMacro{\LWR@origmdseries}{\mdseries}
631 \LetLtxMacro{\LWR@origupshape}{\upshape}
632 \LetLtxMacro{\LWR@origslshape}{\slshape}
633 \LetLtxMacro{\LWR@origscshape}{\scshape}
634 \LetLtxMacro{\LWR@origitshape}{\itshape}
635 \LetLtxMacro{\LWR@origem}{\em}
636 \LetLtxMacro{\LWR@orignormalfont}{\normalfont}
637
638 \let\LWR@origraggedright\raggedright
639 \let\LWR@origonecolumn\onecolumn
640
641 \let\LWR@origsp\sp
642 \let\LWR@origsb\sb
643 \LetLtxMacro{\LWR@origtextsuperscript}{\textsuperscript}
644 \LetLtxMacro{\LWR@orig@textsuperscript}{\@textsuperscript}
645 \AtBeginDocument{
646 \LetLtxMacro{\LWR@origtextsubscript}{\textsubscript}
647 \LetLtxMacro{\LWR@orig@textsubscript}{\@textsubscript}
648 }
649 \LetLtxMacro{\LWR@origunderline}{\underline}
650
651 \let\LWR@origscriptsize\scriptsize
652
653 \let\LWR@orignewpage\newpage

```

```

654
655 \let\LWR@origpagestyle\pagestyle
656 \let\LWR@origthispagestyle\thispagestyle
657 \let\LWR@origpagenumbering\pagenumbering
658
659 \LetLtxMacro{\LWR@origminipage}{\minipage}
660 \let\LWR@origendminipage\endminipage
661 \LetLtxMacro{\LWR@origparbox}{\parbox}
662
663 \let\LWR@orignewline\newline
664
665 \LetLtxMacro{\LWR@origitem}{\item}
666
667 \LetLtxMacro{\LWR@origitemize}{\itemize}
668 \LetLtxMacro{\LWR@endorigitemize}{\enditemize}
669 \LetLtxMacro{\LWR@origenumerate}{\enumerate}
670 \LetLtxMacro{\LWR@endorigenumerate}{\endenumerate}
671 \LetLtxMacro{\LWR@origdescription}{\description}
672 \LetLtxMacro{\LWR@endorigdescription}{\enddescription}
673
674 \let\LWR@origpar\par
675
676 \LetLtxMacro{\LWR@origfootnote}{\footnote}
677 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
678
679 \let\LWR@origclearpage\clearpage
680 \let\clearpage\relax
681 \let\cleardoublepage\relax
682 \end{warpHTML}

```

## 30 Accents

Native  $\TeX$  accents such as `\` will work, but many more kinds of accents are available when using Unicode-aware  $\XeTeX$  and  $\text{Lua}\TeX$ .

for HTML output: 683 \begin{warpHTML}

Without `\AtBeginDocument`, `\t` was being re-defined somewhere.

```
684 \AtBeginDocument{
```

The following are restored for print when inside a `lateximage`.

For Unicode engines, only `\t` needs to be redefined:



```
685 \LetLtxMacro{\LWR@origt}{\t}
```

For pdf<sub>La</sub>TeX, additional work is required:

```
686 \ifPDFTeX
687 \LetLtxMacro{\LWR@origequalaccent}{\=}
688 \LetLtxMacro{\LWR@origdotaccent}{\cdot}
689 \LetLtxMacro{\LWR@origu}{\u}
690 \LetLtxMacro{\LWR@origv}{\v}
691 \LetLtxMacro{\LWR@origc}{\c}
692 \LetLtxMacro{\LWR@origd}{\d}
693 \LetLtxMacro{\LWR@origb}{\b}
```

The HTML redefinitions follow.

For pdf<sub>La</sub>TeX, Unicode diacritical marks are used:

```
694 \renewcommand*{\=}[1]{\#1\HTMLUnicode{0305}}
695 \renewcommand*{\cdot}[1]{\#1\HTMLUnicode{0307}}
696 \renewcommand*{\u}[1]{\#1\HTMLUnicode{0306}}
697 \renewcommand*{\v}[1]{\#1\HTMLUnicode{030C}}
698 \renewcommand*{\c}[1]{\#1\HTMLUnicode{0327}}
699 \renewcommand*{\d}[1]{\#1\HTMLUnicode{0323}}
700 \renewcommand*{\b}[1]{\#1\HTMLUnicode{0331}}
701 \fi
```

For all engines, a Unicode diacritical tie is used:

```
702 \def\LWR@t#1#2{\#1\HTMLUnicode{0361}\#2}
703 \renewcommand*{\t}[1]{\LWR@t#1}
```

`\LWR@restoreorigaccents` Called from `\restoreoriginalformatting` when a lateximage is begun.

```
704 \ifPDFTeX
705 \newcommand*{\LWR@restoreorigaccents}{%
706 \LetLtxMacro{\=}{\LWR@origequalaccent}%
707 \LetLtxMacro{\cdot}{\LWR@origdotaccent}%
708 \LetLtxMacro{\u}{\LWR@origu}%
709 \LetLtxMacro{\v}{\LWR@origv}%
710 \LetLtxMacro{\t}{\LWR@origt}%
711 \LetLtxMacro{\c}{\LWR@origc}%
712 \LetLtxMacro{\d}{\LWR@origd}%
713 \LetLtxMacro{\b}{\LWR@origb}%
714 }
715 \else% XeLaTeX, LuaLaTeX:
716 \newcommand*{\LWR@restoreorigaccents}{%
717 \LetLtxMacro{\t}{\LWR@origt}%
718 }
719 \fi
```

```
720 }% AtBeginDocument
```

```
721 \end{warpHTML}
```

## 31 Configuration Files

```
722 \begin{warpprint}
723 \typeout{lwarp: generating configuration files}
724 \end{warpprint}
```

### 31.1 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.

```
725 \begin{warpprint}
726 \immediate\openout\LWR@quickfile=\jobname_html.tex
727 \immediate\write\LWR@quickfile{%
728 \detokenize{\PassOptionsToPackage}%
729 {warpHTML,BaseJobname=\jobname}{lwarp}%
730 }
731 \immediate\write\LWR@quickfile{%
732 \detokenize{\input}\string{\jobname.tex}\string }%
733 }
734 \immediate\closeout\LWR@quickfile
735 \end{warpprint}
```

### 31.2 lwarpmk.conf

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

An example `lwarpmk.conf`:

---

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

---

```

for PRINT output: 736 \begin{warpprint}
737 \ifcsdef{LWR@quickfile}{-}{\newwrite{LWR@quickfile}}
738 \immediate\openout\LWR@quickfile=lwarpmk.conf
739 \ifbool{usingOSWindows}{
740 \immediate\write\LWR@quickfile{opsystem = "Windows"}
741 }{
742 \immediate\write\LWR@quickfile{opsystem = "Unix"}
743 }
744 \ifPDFTeX
745 \immediate\write\LWR@quickfile{latexname = "pdflatex"}
746 \fi
747 \ifXeTeX
748 \immediate\write\LWR@quickfile{latexname = "xelatex"}
749 \fi
750 \ifLuaTeX
751 \immediate\write\LWR@quickfile{latexname = "lualatex"}
752 \fi
753 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
754 \immediate\write\LWR@quickfile{%
755 homehtmlfilename = "\HomeHTMLFilename"%
756 }
757 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
758 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}}
759 \immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
760 \immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
761 \immediate\closeout\LWR@quickfile
762 \end{warpprint}

```

### 31.3 project.lwarpmkconf

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

```

763 \begin{warpprint}
764 \ifcsdef{LWR@quickfile}{-}{\newwrite{LWR@quickfile}}
765 \immediate\openout\LWR@quickfile=\jobname.lwarpmkconf
766 \ifbool{usingOSWindows}{
767 \immediate\write\LWR@quickfile{opsystem = "Windows"}
768 }{
769 \immediate\write\LWR@quickfile{opsystem = "Unix"}
770 }
771 \ifPDFTeX
772 \immediate\write\LWR@quickfile{latexname = "pdflatex"}
773 \fi
774 \ifXeTeX
775 \immediate\write\LWR@quickfile{latexname = "xelatex"}

```

```

776 \fi
777 \ifLuaTeX
778 \immediate\write\LWR@quickfile{latexname = "lua\latex"}
779 \fi
780 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
781 \immediate\write\LWR@quickfile{%
782 homehtmlfilename = "\HomeHTMLFilename"%
783 }
784 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
785 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
786 \immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
787 \immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
788 \immediate\closeout\LWR@quickfile
789 \end{warpprint}

```

### 31.4 lwarp.css

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

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

```

790 \begin{warpprint}
791 \begin{filecontents*}{lwarp.css}
792 /*
793  CSS stylesheet for the LaTeX lwarp package
794  Copyright 2016-2017 Brian Dunn -- BD Tech Concepts LLC
795 */
796
797
798 /* a fix for older browsers: */
799 header, section, footer, aside, nav, main,
800     article, figure { display: block; }
801
802
803 A:link {color:#000080 ; text-decoration: none ; }
804 A:visited {color:#800000 ; }
805 A:hover {color:#000080 ; text-decoration: underline ;}
806 A:active {color:#800000 ; }
807
808 a.tocpart {display: inline-block ; margin-left: 0em ;
809     font-weight: bold ;}
810 a.tocchapter {display: inline-block ; margin-left: 0em ;
811     font-weight: bold ;}
812 a.tocsection {display: inline-block ; margin-left: 1em ;
813     text-indent: -.5em ; font-weight: bold ; }
814 a.tocsubsection {display: inline-block ; margin-left: 2em ;

```

```
815     text-indent: -.5em ; }
816 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
817     text-indent: -.5em ; }
818 a.tocparagraph {display: inline-block ; margin-left: 4em ;
819     text-indent: -.5em ; }
820 a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
821     text-indent: -.5em ; }
822 a.tocfigure {margin-left: 0em}
823 a.tocsubfigure {margin-left: 2em}
824 a.toc table {margin-left: 0em}
825 a.tocsubtable {margin-left: 2em}
826 a.toctheorem {margin-left: 0em}
827 a.toclstlisting {margin-left: 0em}
828
829
830 body {
831     font-family: "DejaVu Serif", "Bitstream Vera Serif",
832         "Lucida Bright", Georgia, serif;
833     background: #FAF7F4 ;
834     color: black ;
835     margin: 0em ;
836     padding: 0em ;
837     font-size: 100% ;
838     line-height: 1.2 ;
839 }
840
841 p {margin: 1.5ex 0em 1.5ex 0em ;}
842
843 /* Holds a section number to add space between it and the name */
844 span.sectionnumber { margin-right: 0em }
845
846 /* Inserted in front of index lines */
847 span.indexitem {margin-left: 0em}
848 span.indexsubitem {margin-left: 2em}
849 span.indexsubsubitem {margin-left: 4em}
850
851 div.hidden, span.hidden { display: none ; }
852
853 kbd {
854     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
855         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
856         "Courier New", monospace;
857     font-size: 100% ;
858 }
859
860 pre { padding: 3pt ; }
861
862 span.strong { font-weight: bold; }
863
864 span.textmd { font-weight: normal; }
```

```
865
866 span.textsc { font-variant: small-caps; }
867
868 span.textsl { font-style: oblique; }
869
870 span.textup { font-variant: normal; }
871
872 span.textrm {
873     font-family: "DejaVu Serif", "Bitstream Vera Serif",
874     "Lucida Bright", Georgia, serif;
875 }
876
877 span.textsf {
878     font-family: "DejaVu Sans", "Bitstream Vera Sans",
879     Geneva, Verdana, sans-serif ;
880 }
881
882 span.textcircled { border: 1px solid black ; border-radius: 1ex ; }
883
884 span.underline {
885     text-decoration: underline ;
886     text-decoration-skip ;
887 }
888
889
890 /* For realscripts */
891 .supsubscript {
892     display: inline-block;
893     text-align: left ;
894 }
895
896 .supsubscript sup,
897 .supsubscript sub {
898     position: relative;
899     display: block;
900     font-size: .5em;
901     line-height: 1;
902 }
903
904 .supsubscript sup {
905     top: .5em;
906 }
907
908 .supsubscript sub {
909     top: .5em;
910 }
911
912 span.attribution {
913     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
914 }
```

```
915
916 span.citetitle {
917   margin-left: 1em ; font-size: 80% ; font-style: oblique;
918 }
919
920 span.poemtitle {
921   font-size: 120% ; font-weight: bold;
922 }
923
924 pre.tabbing {
925   font-family: "Linux Libertine Mono O", "Lucida Console",
926               "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
927               "Liberation Mono", "FreeMono", "Andale Mono",
928               "Nimbus Mono L", "Courier New", monospace;
929 }
930
931 blockquote {
932   margin-left: 0px ;
933   margin-right: 0px ;
934 }
935
936 /* quotchap is for the quotchap package */
937 div.quotchap {
938   font-style: oblique ;
939   overflow-x: auto ;
940   margin-left: 2em ;
941   margin-right: 2em ;
942 }
943
944 blockquote p, div.quotchap p {
945   line-height: 1.5;
946   text-align: left ;
947   font-size: .85em ;
948   margin-left: 3em ;
949   margin-right: 3em ;
950 }
951
952 /* qauthor is for the quotchap package */
953 div.qauthor {
954   display: block ;
955   text-align: right ;
956   margin-left: auto ;
957   margin-right: 2em ;
958   font-size: 80% ;
959   font-variant: small-caps;
960 }
961
962 div.qauthor p {
963   text-align: right ;
964 }
```

```
965
966 blockquotation {
967     margin-left: 0px ;
968     margin-right: 0px ;
969 }
970
971 blockquotation p {
972     line-height: 1.5;
973     text-align: left ;
974     font-size: .85em ;
975     margin-left: 3em ;
976     margin-right: 3em ;
977 }
978
979 div.epigraph {
980     line-height: 1.2;
981     text-align: left ;
982     padding: 3ex 1em 0ex 1em ;
983 /*     margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
984     margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
985 /*     margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
986     font-size: .85em ;
987     max-width: 27em ;
988 }
989
990
991
992 div.epigraphsource{
993     text-align:right ;
994     margin-left:auto ;
995 /*     max-width: 50% ; */
996     border-top: 1px solid #A0A0A0 ;
997     padding-bottom: 3ex ;
998     line-height: 1.2;
999 }
1000
1001 div.epigraph p { padding: .5ex ; margin: 0ex ;}
1002 div.epigraphsource p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
1003
1004
1005 /* lettrine package: */
1006 span.lettrine { font-size: 3ex ; float: left ; }
1007 span.lettrinetext { font-variant: small-caps ; }
1008
1009 /* ulem and soul packages: */
1010 span.underline {
1011     text-decoration: underline ;
1012     text-decoration-skip ;
1013 }
1014
```



```
1015 span.uuline {
1016     text-decoration: underline ;
1017     text-decoration-skip ;
1018     text-decoration-style: double ;
1019 }
1020
1021 span.uwave {
1022     text-decoration: underline ;
1023     text-decoration-skip ;
1024     text-decoration-style: wavy ;
1025 }
1026
1027 span.sout {
1028     text-decoration: line-through ;
1029 }
1030
1031 span.xout {
1032     text-decoration: line-through ;
1033 }
1034
1035 span.dashuline {
1036     text-decoration: underline ;
1037     text-decoration-skip ;
1038     text-decoration-style: dashed ;
1039 }
1040
1041 span.dotuline {
1042     text-decoration: underline ;
1043     text-decoration-skip ;
1044     text-decoration-style: dotted ;
1045 }
1046
1047 span.letterspacing { letter-spacing: .2ex ; }
1048
1049 span.capsspacing {
1050     font-variant: small-caps ;
1051     letter-spacing: .1ex ;
1052 }
1053
1054 span.highlight { background: #F8E800 ; }
1055
1056
1057
1058
1059 html body {
1060     margin: 0 ;
1061     line-height: 1.2;
1062 }
1063
1064
```

```
1065 body div {
1066     margin: 0ex;
1067 }
1068
1069
1070 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1071 {
1072     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1073         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1074         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1075         "Times New Roman", serif;
1076     font-style: normal ;
1077     font-weight: bold ;
1078     text-align: left ;
1079 }
1080
1081 h1 { /* title of the entire website, used on each page */
1082     text-align: center ;
1083     font-size: 2.5em ;
1084     padding: .4ex 0em 0ex 0em ;
1085 }
1086 h2 { font-size: 2.25em }
1087 h3 { font-size: 2em }
1088 h4 { font-size: 1.75em }
1089 h5 { font-size: 1.5em }
1090 h6 { font-size: 1.25em }
1091 span.paragraph {font-size: 1em ; font-variant: normal ;
1092     margin-right: 1em ; }
1093 span.subparagraph {font-size: 1em ; font-variant: normal ;
1094     margin-right: 1em ; }
1095
1096
1097
1098 /* Title of the file */
1099 h1 {
1100     margin: 0ex 0em 0ex 0em ;
1101     line-height: 1.3;
1102     text-align: center ;
1103 }
1104
1105 /* Part */
1106 h2 {
1107     margin: 1ex 0em 1ex 0em ;
1108     line-height: 1.3;
1109     text-align: center ;
1110 }
1111
1112 /* Chapter */
1113 h3 {
1114     margin: 3ex 0em 1ex 0em ;
```

```
1115 line-height: 1.3;
1116 }
1117
1118 /* Section */
1119 h4 {
1120   margin: 3ex 0em 1ex 0em ;
1121   line-height: 1.3;
1122 }
1123
1124 /* Sub-Section */
1125 h5 {
1126   margin: 3ex 0em 1ex 0em ;
1127   line-height: 1.3;
1128 }
1129
1130 /* Sub-Sub-Section */
1131 h6 {
1132   margin: 3ex 0em 1ex 0em ;
1133   line-height: 1.3;
1134 }
1135
1136
1137 div.titlepage {
1138   text-align: center ;
1139 }
1140
1141 .footnotes {
1142   font-size: .85em ;
1143   margin: 3ex 1em 0ex 1em ;
1144   padding-bottom: 1ex ;
1145   border-top: 1px solid silver ;
1146 }
1147
1148 .marginpar {
1149   max-width:50%;
1150   float:right;
1151   text-align:left;
1152   margin: 1ex 0.5em 1ex 1em ;
1153   padding: 1ex 0.5em 1ex 0.5em ;
1154   font-size: 85% ;
1155   border-top: 1px solid silver ;
1156   border-bottom: 1px solid silver ;
1157   overflow-x: auto;
1158 }
1159
1160 .marginpar br { margin-bottom: 2ex ; }
1161
1162 div.marginblock {
1163   max-width:50%;
1164   float:right;
```

```
1165     text-align:left;
1166     margin: 1ex 0.5em 1ex 1em ;
1167     padding: 1ex 0.5em 1ex 0.5em ;
1168     overflow-x: auto;
1169 }
1170
1171 div.marginblock div.minipage {
1172     display: block ;
1173     margin: 0pt auto 0pt auto ;
1174 }
1175
1176 div.marginblock div.minipage p { font-size: 85%}
1177
1178 div.marginblock br { margin-bottom: 2ex ; }
1179
1180
1181 section.textbody div.footnotes{
1182     margin: 3ex 0em 0ex 0em ;
1183     border-bottom: 2px solid silver ;
1184 }
1185
1186 .footnoteheader {
1187     border-top: 2px solid silver ;
1188     margin-top: 3ex ;
1189     padding-top: 1ex ;
1190     font-weight: bold ;
1191 }
1192
1193 .mpfootnotes {
1194     text-align: left ;
1195     font-size: .85em ;
1196     margin-left: 1em ;
1197     border-top: 1px solid silver ;
1198 }
1199
1200 /* Remove footnote top border in the title page. */
1201 div.titlepage div.mpfootnotes {
1202     border-top: none ;
1203 }
1204
1205
1206
1207 ol {
1208     margin: 1ex 1em 1ex 0em;
1209     line-height: 1.2;
1210 }
1211
1212 ul, body dir, body menu {
1213     margin: 1ex 1em 1ex 0em;
1214     line-height: 1.2;
```

```
1215 }
1216
1217 li { margin: 0ex 0em 1ex 0em; }
1218
1219 html {
1220     margin: 0;
1221     padding: 0;
1222 }
1223
1224 .programlisting {
1225     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1226                 "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1227                 "Courier New", monospace;
1228     margin: 1ex 0ex 1ex 0ex ;
1229     padding: .5ex 0pt .5ex 0pt ;
1230     overflow-x: auto;
1231 }
1232
1233 section.textbody>pre.programlisting {
1234 border-top: 1px solid silver ;
1235 border-bottom: 1px solid silver ;
1236 }
1237
1238
1239 .inlineprogramlisting {
1240     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1241                 "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1242                 "Courier New", monospace;
1243     overflow-x: auto;
1244 }
1245
1246 span.listinglabel {
1247     display: inline-block ;
1248     font-size: 70% ;
1249     width: 4em ;
1250     text-align: right ;
1251     margin-right: 2em ;
1252 }
1253
1254 div.abstract {
1255     margin: 2em 5% 2em 5% ;
1256     padding: 1ex 1em 1ex 1em ;
1257     /* font-weight: bold ; */
1258     font-size: 90% ;
1259     text-align: left ;
1260 }
1261
1262 div.abstract dl {line-height:1.5;}
1263 div.abstract dt {color:#304070;}
1264
```

```
1265 div.abstracttitle{
1266     font-family: "URW Classico", Optima, "Linux Biolinum O",
1267         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1268         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1269     font-weight:bold;
1270     font-size:1.25em;
1271     text-align: center ;
1272 }
1273
1274 span.abstracrunintitle{
1275     font-family: "URW Classico", Optima, "Linux Biolinum O",
1276         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1277         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1278     font-weight:bold;
1279 }
1280
1281
1282 .verbatim {
1283     overflow-x: auto ;
1284 }
1285
1286 .alltt {
1287     overflow-x: auto ;
1288 }
1289
1290
1291 .bverbatim {
1292     margin: 1ex 0pt 1ex 0pt ;
1293     padding: .5ex 0pt .5ex 0pt ;
1294     overflow-x: auto ;
1295 }
1296
1297 .lverbatim {
1298     margin: 1ex 0pt 1ex 0pt ;
1299     padding: .5ex 0pt .5ex 0pt ;
1300     overflow-x: auto ;
1301 }
1302
1303 .fancyvrb {
1304     font-size:.85em ;
1305     margin: 3ex 0pt 3ex 0pt
1306 }
1307
1308 .fancyvrblabel {
1309     font-weight:bold;
1310     text-align: center ;
1311 }
1312
1313
1314 .verse {
```

```
1315     font-family: "Linux Libertine Mono O", "Lucida Console",
1316               "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1317               "Liberation Mono", "FreeMono", "Andale Mono",
1318               "Nimbus Mono L", "Courier New", monospace;
1319     margin-left: 1em ;
1320 }
1321
1322
1323 div.singlespace { line-height: 1.2 ; }
1324 div.onehalfspace { line-height: 1.5 ; }
1325 div.doublespace { line-height: 2 ; }
1326
1327
1328 /* Word processor format output: */
1329 div.wpfigure { border: 1px solid red ; margin: .5ex ; padding: .5ex ; }
1330 div.wptable { border: 1px solid blue ; margin: .5ex ; padding: .5ex ; }
1331 div.wpminipage { border: 1px solid green ; margin: .5ex ; padding: .5ex ; }
1332
1333
1334
1335
1336 /* Minipage environments, vertically aligned to top, center, bottom: */
1337 .minipage, .fminipage, .fcolorminipage {
1338     /* display: inline-block ; */
1339     /* Mini pages which follow each other will be tiled. */
1340     margin: .25em .25em .25em .25em;
1341     padding: .25em .25em .25em .25em;
1342     display: inline-flex;
1343     flex-direction: column ;
1344     overflow: auto;
1345 }
1346
1347 /* Paragraphs in the flexbox did not collapse their margins. */
1348 /* Have not yet researched this. */
1349 .minipage p {margin: .75ex 0em .75ex 0em ;}
1350
1351 .fbboxBlock .minipage, .colorbox .minipage, .colorboxBlock .minipage,
1352 .fcolorbox .minipage, .fcolorboxBlock .minipage
1353     {border: none ; background: none;}
1354
1355 .fbbox, .fbboxBlock { border: 1px solid black ; }
1356
1357 .fbbox, .fbboxBlock, .fcolorbox, .fcolorboxBlock, .colorbox, .colorboxBlock,
1358 .fminipage, .fcolorminipage
1359     {display: inline-block}
1360
1361 .shadowbox, .shabox {
1362     border: 1px solid black;
1363     box-shadow: 3px 3px 3px #808080 ;
1364     border-radius: 0px ;
```

```
1365     padding: .4ex .3em .4ex .3em ;
1366     margin: 0pt .3ex 0pt .3ex ;
1367     display: inline-block ;
1368 }
1369
1370 .doublebox {
1371     border: 3px double black;
1372     border-radius: 0px ;
1373     padding: .4ex .3em .4ex .3em ;
1374     margin: 0pt .3ex 0pt .3ex ;
1375     display: inline-block ;
1376 }
1377
1378 .ovalbox, .Ovalbox {
1379     border: 1px solid black;
1380     border-radius: 1ex ;
1381     padding: .4ex .3em .4ex .3em ;
1382     margin: 0pt .3ex 0pt .3ex ;
1383     display: inline-block ;
1384 }
1385
1386 .Ovalbox { border-width: 2px ; }
1387
1388 .framebox {
1389     border: 1px solid black;
1390     border-radius: 0px ;
1391     padding: .3ex .2em 0ex .2em ;
1392     margin: 0pt .1ex 0pt .1ex ;
1393     display: inline-block ;
1394 }
1395
1396
1397 .mdframed {
1398 /*     padding: 0ex ; */
1399 /*     border: 1px solid black; */
1400 /*     border-radius: 0px ; */
1401     padding: 0ex ;
1402     margin: 3ex 5% 3ex 5% ;
1403 /*     display: inline-block ; */
1404 }
1405
1406 .mdframed p { padding: 0ex .5em 0ex .5em ; }
1407
1408 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
1409
1410 .mdframedtitle {
1411     padding: .5em ;
1412     display: block ;
1413     font-size: 130%
1414 }
```



```
1415
1416 .mdframedsubtitle {
1417     padding: 0ex .5em 0ex .5em ;
1418     display: block ;
1419     font-size: 115% ;
1420 }
1421
1422 .mdframedsubsubtitle {
1423     padding: 0ex .5em 0ex .5em ;
1424     display: block ;
1425 }
1426
1427 .mdtheorem {
1428     padding: 0ex .5em 0ex .5em ;
1429     margin: 3ex 5% 3ex 5% ;
1430 /*    display: inline-block ; */
1431 }
1432
1433
1434 /* framed package */
1435 .framed, pre.boxedverbatim, fcolorbox {
1436     margin: 3ex 0em 3ex 0em ;
1437     border: 1px solid black;
1438     border-radius: 0px ;
1439     padding: .3ex 1em 0ex 1em ;
1440     display: block ;
1441 }
1442
1443 .snugframed {
1444     margin: 3ex 0em 3ex 0em ;
1445     border: 1px solid black;
1446     border-radius: 0px ;
1447     display: block ;
1448 }
1449
1450 .framedleftbar {
1451     margin: 3ex 0em 3ex 0em ;
1452     border-left: 3pt solid black;
1453     border-radius: 0px ;
1454     padding: .3ex .2em .3ex 1em ;
1455     display: block ;
1456 }
1457
1458 .framedtitle {
1459     margin: 0em ;
1460     padding: 0em ;
1461     font-size: 130%
1462 }
1463
1464 .framedtitle p { padding: .3em }
```

```
1465
1466
1467
1468 dl {
1469     margin: 1ex 2em 1ex 0em;
1470     line-height: 1.3;
1471 }
1472
1473 dl dt {
1474     margin-top: 1ex;
1475     margin-left: 1em ;
1476     font-weight: bold;
1477 }
1478
1479 dl dd p { margin-top: 0em; }
1480
1481
1482 nav.toc, nav.lof, nav.lot, nav.lol, nav.lothm {
1483     font-family: "URW Classico", Optima, "Linux Biolinum O",
1484         "DejaVu Sans", "Bitstream Vera Sans",
1485         Geneva, Verdana, sans-serif ;
1486     margin-bottom: 4ex ;
1487 }
1488
1489 nav.toc p, nav.lof p, nav.lot p, nav.lol p, nav.lothm p {
1490     line-height: 1.2 ;
1491     margin-top: .5ex ;
1492     margin-bottom: .5ex;
1493     font-size: .9em ;
1494 }
1495
1496
1497
1498 img, img.hyperimage, img.borderimage {
1499     max-width: 600px;
1500     border: 1px solid silver;
1501     box-shadow: 3px 3px 3px #808080 ;
1502     padding: .5% ;
1503     margin: .5% ;
1504     background: none ;
1505 }
1506
1507 img.inlineimage{
1508     padding: 0px ;
1509     box-shadow: none ;
1510     border: none ;
1511     background: none ;
1512     margin: 0px ;
1513     display: inline-block ;
1514     border-radius: 0px ;
```

```
1515 }
1516
1517 img.logoimage{
1518     max-width: 300px ;
1519     box-shadow: 3px 3px 3px #808080 ;
1520     border: 1px solid black ;
1521     background:none ;
1522     padding:0 ;
1523     margin:.5ex ;
1524     border-radius: 10px ;
1525 }
1526
1527
1528 .section {
1529 /*
1530     To have each section float relative to each other:
1531 */
1532 /*
1533     display: block ;
1534     float: left ;
1535     position: relative ;
1536     background: white ;
1537     border: 1px solid silver ;
1538     padding: .5em ;
1539 */
1540     margin: 0ex .5em 0ex .5em ;
1541     padding: 0 ;
1542 }
1543
1544
1545 figure {
1546     margin: 3ex auto 3ex auto ;
1547     padding: 1ex 1em 1ex 1em ;
1548     overflow-x: auto ;
1549 }
1550
1551
1552 /* To automatically center images in figures: */
1553 /*
1554 figure img.inlineimage {
1555     margin: 0ex auto 0ex auto ;
1556     display: block ;
1557 }
1558 */
1559
1560 /* To automatically center minipages in figures: */
1561 /*
1562 figure div.minipage, figure div.minipage div.minipage {
1563     margin: 1ex auto 1ex auto ;
1564     display: block ;
```

```
1565 }
1566 */
1567
1568 figure div.minipage p { font-size: 85% ; }
1569
1570 figure.subfigure, figure.subtable {
1571     display: inline-block ; margin: 3ex 1em 3ex 1em ;
1572 }
1573
1574 figcaption .minipage { margin:0 ; padding: 0 }
1575
1576 div.minipage figure { border: none ; box-shadow: none ; }
1577
1578 div.floatrow { text-align: center; }
1579
1580 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
1581
1582 div.floatfoot { font-size: .85em ;
1583     border-top: 1px solid silver ; line-height: 1.2 ; }
1584
1585 figcaption , .lstlistingtitle {
1586     font-size: .85em ;
1587     text-align: center ;
1588     font-weight: bold ;
1589     margin-top: 1ex ;
1590     margin-bottom: 1ex ;
1591 }
1592
1593 figure.subfigure figcaption, figure.subtable figcaption {
1594     border-bottom: none ; background: none ;
1595 }
1596
1597 div.nonfloatcaption {
1598     margin: 1ex auto 1ex auto ;
1599     font-size: .85em ;
1600     text-align: center ;
1601     font-weight: bold ;
1602 }
1603
1604 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
1605 figure div.floatrow div.minipage figcaption {
1606     border: none ;
1607     background: none ;
1608 }
1609
1610
1611 table {
1612     margin: 1ex auto 1ex auto ;
1613     border-collapse: separate ;
1614     border-spacing: 0px ;
```

```
1615     line-height: 1.3 ;
1616   }
1617
1618tr.hline td {border-top: 1px solid #808080 ; margin-top: 0ex ;
1619    margin-bottom: 0ex ; } /* for \hline */
1620
1621tr.tbrule td {border-top: 1px solid black ; margin-top: 0ex ;
1622    margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
1623
1624td {padding: 1ex .5em 1ex .5em ;}
1625
1626table td.tdl { text-align: left ; vertical-align: middle ; }
1627table td.tdc { text-align: center ; vertical-align: middle ; }
1628table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
1629table td.tdbang { text-align: center ; vertical-align: middle ; }
1630table td.tdr { text-align: right ; vertical-align: middle ; }
1631table td.tdp { text-align: left ; vertical-align: bottom ; }
1632table td.tdm { text-align: left ; vertical-align: middle ; }
1633table td.tdb { text-align: left ; vertical-align: top ; }
1634table td.tdP { text-align: center ; vertical-align: bottom ; }
1635table td.tdM { text-align: center ; vertical-align: middle ; }
1636table td.tdB { text-align: center ; vertical-align: top ; }
1637
1638table td.tvertbarl { border-left: 1px solid black }
1639table td.tvertbarr { border-right: 1px solid black }
1640
1641
1642/* for cmidrules: */
1643table td.tdrule {
1644    border-top: 1px solid #A0A0A0 ;
1645}
1646
1647table td.tdrulel {
1648    border-top-left-radius:.5em ;
1649    border-top: 1px solid #A0A0A0 ;
1650}
1651
1652table td.tdruler {
1653    border-top-right-radius:.5em ;
1654    border-top: 1px solid #A0A0A0 ;
1655}
1656
1657table td.tdrulelr {
1658    border-top-left-radius:.5em ;
1659    border-top-right-radius:.5em ;
1660    border-top: 1px solid #A0A0A0 ;
1661}
1662
1663
1664/* Margins of paragraphs inside table cells: */
```

```
1665 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
1666     padding-bottom: 1ex ; margin: 0ex ; }
1667 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
1668     padding-bottom: 1ex ; margin: 0ex ; }
1669 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
1670     padding-bottom: 1ex ; margin: 0ex ; }
1671
1672 td.tdp , td.tdprule , td.tdP , td.tdPrule
1673     { padding: 0ex .5em 0ex .5em ; }
1674 td.tdm , td.tdmrule , td.tdM , td.tdMrule
1675     { padding: 0ex .5em 0ex .5em ; }
1676 td.tdb , td.tdbrule , td.tdB , td.tdBrule
1677     { padding: 0ex .5em 0ex .5em ; }
1678
1679
1680 /* table notes: */
1681 .tnotes {
1682     margin: 0ex 5% 1ex 5% ;
1683     padding: 0.5ex 1em 0.5ex 1em;
1684     font-size:.85em;
1685     text-align: left ;
1686 }
1687
1688 .tnotes dl dt p {margin-bottom:0px;}
1689
1690 .tnoteitemheader {margin-right: 1em;}
1691
1692
1693 /* for bigdelim */
1694 .ldelim, .rdelim { font-size: 200% }
1695
1696
1697 /* center, flushleft, flushright environments */
1698 div.center{text-align:center;}
1699 div.center table {margin-left:auto;margin-right:auto;}
1700 div.flushleft{text-align:left;}
1701 div.flushleft table {margin-left:0em ; margin-right:auto;}
1702 div.flushright{text-align:right;}
1703 div.flushright table {margin-left:auto ; margin-right: 0em ;}
1704
1705
1706 /* Fancybox */
1707 div.Btrivlist table tr td { padding: .2ex 0em ; }
1708
1709
1710 /* program listing callouts: */
1711 span.callout {
1712     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1713         Geneva, Verdana, sans-serif ;
1714     border-radius: .5em;
```

```
1715     background-color:black;
1716     color:white;
1717     padding:0px .25em 0px .25em;
1718     margin: 0 ;
1719     font-weight: bold;
1720     font-size:.72em ;
1721 }
1722
1723 div.programlisting pre.verbatim span.callout{
1724     font-size: .85em ;
1725 }
1726
1727
1728
1729
1730
1731 div.published
1732 {
1733     text-align: center ;
1734     font-variant: normal ;
1735     font-style: italic ;
1736     font-size: 1em ;
1737     margin: 3ex 0em 3ex 0em ;
1738 }
1739
1740 div.subtitle
1741 {
1742     text-align: center ;
1743     font-variant: normal ;
1744     font-style: italic ;
1745     font-size: 1.25em ;
1746     margin: 3ex 0em 3ex 0em ;
1747 }
1748
1749 div.subtitle p { margin: 1ex ; }
1750
1751 div.author
1752 {
1753     font-variant: normal ;
1754     font-style: normal ;
1755     font-size: 1em ;
1756     margin: 3ex 0em 3ex 0em ;
1757 }
1758
1759 div.oneauthor {
1760     display: inline-block ;
1761     margin: 3ex 1em 0ex 1em ;
1762 }
1763
1764 /*
```

```
1765 div.author table {
1766     margin: 3ex auto 0ex auto ;
1767     background: none ;
1768 }
1769
1770 div.author table tbody tr td { padding: .25ex ; }
1771 */
1772
1773 span.affiliation {font-size: .85em ; font-variant: small-caps; }
1774
1775 div.titledate {
1776     text-align: center ;
1777     font-size: .85em ;
1778     font-style: italic;
1779     margin: 6ex 0em 6ex 0em ;
1780 }
1781
1782
1783 nav.topnavigation{
1784     text-align: left ;
1785     padding: 0.5ex 1em 0.5ex 1em ;
1786 /*     margin: 2ex 0em 3ex 0em ; */
1787     margin: 0 ;
1788     border-bottom: 1px solid silver ;
1789     border-top: 1px solid silver ;
1790     clear:right ;
1791 }
1792
1793 nav.botnavigation{
1794     text-align: left ;
1795     padding: 0.5ex 1em 0.5ex 1em ;
1796 /*     margin: 3ex 0em 2ex 0em ; */
1797     margin: 0 ;
1798     border-top: 1px solid silver ;
1799     border-bottom: 1px solid silver ;
1800     clear:right ;
1801 }
1802
1803
1804 header{
1805     line-height: 1.2 ;
1806     font-size: 1em ;
1807 /*     border-bottom: 2px solid silver ; */
1808     margin: 0px ;
1809     padding: 0ex 1em 0ex 1em ;
1810     text-align:center ;
1811 }
1812
1813 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
1814
```



```
1815
1816 footer{
1817     font-size: .85em ;
1818     line-height: 1.2 ;
1819     margin-top: 1ex ;
1820     border-top: 2px solid silver ;
1821     padding: 2ex 1em 2ex 1em ;
1822     clear:right ;
1823     text-align:left ;
1824 }
1825
1826
1827 a.linkhome { font-weight:bold ; font-size: 1em ;}
1828
1829
1830 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
1831
1832 img.lateximage{
1833     padding: 0px 0px 0px 0px ;
1834     box-shadow: none ;
1835     border: none ;
1836     background: none ;
1837     margin: 0px 0px -.15ex 0px ;
1838     /* pdfcrop leaves a slight margin, adjust to baseline */
1839     max-width: 100% ;
1840     border-radius: 0ex ;
1841     border: none ;
1842 }
1843
1844
1845
1846 nav.sidetoc {
1847     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1848         "Lucida Bright", Georgia, serif;
1849     float:right ;
1850     width: 20%;
1851     border-left: 1px solid silver;
1852     border-top: 1px solid silver;
1853     border-bottom: 1px solid silver;
1854 /*     border-top: 2px solid #808080 ; */
1855     background: #FAF7F4 ;
1856     padding: 2ex 0em 2ex 1em ;
1857     margin: 0ex 0em 2ex 1em ;
1858     font-size:.9em ;
1859     border-radius: 20px 0px 0px 20px ;
1860 }
1861
1862 div.sidetoccontents {
1863 /*     border-top: 1px solid silver ; */
1864     overflow-y: auto ;
```

```
1865     width: 100% ;
1866     text-align: left ;
1867 }
1868
1869
1870 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
1871     text-indent: 0 ; }
1872
1873 nav.sidetoc p a {color:black ; font-size: .7em ;}
1874
1875 div.sidetoc title {font-size: 1.2em; font-weight:bold; text-align:center;
1876     border-bottom: 1px solid silver ;    }
1877
1878 nav.sidetoc a:hover {text-decoration: underline ; }
1879
1880
1881
1882 section.textbody { margin: 0ex 1em 0ex 1em ;}
1883
1884
1885 div.multicolsheading { -webkit-column-span: all;
1886     -moz-column-span: all; column-span: all; }
1887 div.multicols { -webkit-columns: 3 380px ;
1888     -moz-columns: 3 380px ; columns: 3 380px ; }
1889 div.multicols p {margin-top: 0ex}
1890
1891
1892
1893 /* Used to support algorithmicx: */
1894 span.floatright { float: right ; }
1895
1896
1897
1898
1899 /* Native LaTeX theorems: */
1900
1901 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
1902 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
1903
1904
1905 /* theorem, amsthm, and ntheorem packages */
1906
1907 span.theoremheader,
1908 span.theoremheaderplain,
1909 span.theoremheaderdefinition,
1910 span.theoremheaderbreak,
1911 span.theoremheadermarginbreak,
1912 span.theoremheaderchangebreak,
1913 span.theoremheaderchange,
1914 span.theoremheadermargin
```

```

1915 {
1916     font-style:normal ; font-weight: bold ; margin-right: 1em ;
1917 }
1918
1919 span.amsthmnameplain,
1920 span.amsthmnamedefinition,
1921 span.amsthmnumberplain,
1922 span.amsthmnumberdefinition
1923 {
1924     font-style:normal ; font-weight: bold ;
1925 }
1926
1927
1928 span.amsthmnameremark,
1929 span.amsthmnumberremark
1930 {font-style:italic ; font-weight: normal ; }
1931
1932
1933 span.amsthmnoteplain,
1934 span.amsthmnotedefinition
1935 {font-style:normal ;}
1936
1937
1938 span.theoremheaderremark,
1939 span.theoremheaderproof,
1940 span.amsthmproofname
1941 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
1942
1943 span.theoremheadersc
1944 {
1945     font-style:normal ;
1946     font-variant: small-caps ;
1947     font-weight: normal ;
1948     margin-right: 1em ;
1949 }
1950
1951 .theoremdemark {float:right}
1952
1953 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
1954 div.theorembodybreak, div.theorembodynonumberbreak,
1955 div.theorembodymarginbreak,
1956 div.theorembodychangebreak,
1957 div.theorembodychange,
1958 div.theorembodymargin
1959 {
1960     font-style:italic;
1961     margin-top: 3ex ; margin-bottom: 3ex ;
1962 }
1963
1964 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,

```

```
1965 div.theorembodyplainupright, nonumberplainuprightsc,
1966 div.amsthmbodydefinition, div.amsthmbodyremark,
1967 div.amsthmproof
1968 {
1969     font-style: normal ;
1970     margin-top: 3ex ; margin-bottom: 3ex ;
1971 }
1972
1973 span.amsthmnoteremark {}
1974
1975
1976
1977 /*
1978 For CSS LaTeX and related logos:
1979 Based on:
1980 http://edward.oconnor.cx/2007/08/tex-poshlet
1981 http://nitens.org/taraborelli/texlogo
1982 */
1983
1984 .latexlogofont {
1985     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1986         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1987     font-variant: normal ;
1988 }
1989
1990 .latexlogo {
1991     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1992         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1993     letter-spacing: .03em ;
1994     font-size: 1.1em;
1995 }
1996
1997 .latexlogo sup {
1998     text-transform: uppercase;
1999     letter-spacing: .03em ;
2000     font-size: 0.85em;
2001     vertical-align: 0.15em;
2002     margin-left: -0.36em;
2003     margin-right: -0.15em;
2004 }
2005
2006 .latexlogo sub {
2007     text-transform: uppercase;
2008     vertical-align: -0.5ex;
2009     margin-left: -0.1667em;
2010     margin-right: -0.125em;
2011     font-size: 1em;
2012 }
2013
2014 .xetexlogo {
```

```
2015     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2016           "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2017     letter-spacing: .03em ;
2018     font-size: 1.1em;
2019 }
2020
2021 /* A smaller gap between Xe and Tex v.s. LaTeX: */
2022 .xetexlogo sub {
2023     text-transform: uppercase;
2024     vertical-align: -0.5ex;
2025     margin-left: -0.0667em;
2026     margin-right: -0.2em;
2027     font-size: 1em;
2028     letter-spacing: .03em ;
2029 }
2030
2031 /* A large gap between Xe and LaTeX v.s. TeX: */
2032 .xelatexlogo sub {
2033     text-transform: uppercase;
2034     vertical-align: -0.5ex;
2035     margin-left: -0.0667em;
2036     margin-right: -.05em;
2037     font-size: 1em;
2038     letter-spacing: .03em ;
2039 }
2040
2041 .amslogo {
2042     font-family: "TeXGyreChorus","URW Chancery L",
2043           "Apple Chancery","ITC Zapf Chancery","Monotype Corsiva",
2044           "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
2045           "Hoefler Text", Times, "Times New Roman", serif;
2046     font-style: italic;
2047 }
2048
2049 .lyxlogo {
2050     font-family: "URW Classico", Optima, "Linux Biolinum O",
2051           "DejaVu Sans", "Bitstream Vera Sans", Geneva,
2052           Verdana, sans-serif ;
2053 }
2054
2055
2056
2057
2058 /* Only display top and bottom navigation if a small screen: */
2059 /* Hide the sidetoc if a small screen: */
2060 nav.topnavigation { display:none; }
2061 nav.botnavigation { display:none; }
2062
2063 @media screen and (max-width: 45em) {
2064 /*     nav.sidetoc {display:none;} */
```

```
2065     nav.sidetoc {
2066         float: none ;
2067         width: 100% ;
2068         margin: 5ex 0px 5ex 0px ;
2069         padding: 0 ;
2070         border-radius: 0 ;
2071         border-bottom: 1px solid black ;
2072         border-top: 1px solid black ;
2073         box-shadow: none ;
2074     }
2075 /*     nav.topnavigation { display:block } */
2076     nav.botnavigation { display:block }
2077     .marginpar {
2078         max-width: 100%;
2079         float: none;
2080         display:block ;
2081         margin: 1ex 1em 1ex 1em ;
2082     }
2083 }
2084
2085 @media print {
2086     body {
2087         font-family: "Linux Libertine O",
2088         "DejaVu Serif", "Bitstream Vera Serif",
2089         "Liberation Serif", "Nimbus Roman No 9 L",
2090         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2091     }
2092     nav.sidetoc { display:none; }
2093     nav.topnavigation { display: none; }
2094     nav.botnavigation { display: none; }
2095 }
2096
2097 @media handheld {
2098     nav.sidetoc { display:none; }
2099     nav.topnavigation { display:block }
2100     nav.botnavigation { display:block }
2101 }
2102
2103 @media projection {
2104     nav.sidetoc { display:none; }
2105     nav.topnavigation { display:block }
2106     nav.botnavigation { display:block }
2107 }
2108 \end{filecontents*}
2109 % \end{Verbatim}% for syntax highlighting
2110 \end{warpprint}
```

### 31.5 lwarp\_sagebrush.css

File lwarp\_sagebrush.css An optional css which may be used for a semi-modern appearance.

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

```

2111 \begin{warpprint}
2112 \begin{filecontents*}{lwarp_sagebrush.css}
2113 @import url("lwarp.css") ;
2114
2115
2116 A:link {color:#105030 ; text-decoration: none ; }
2117 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
2118 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
2119 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
2120
2121
2122
2123 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2124 {
2125     font-family: "URW Classico", Optima, "Linux Biolinum O",
2126         "Linux Libertine O", "Liberation Serif",
2127         "Nimbus Roman No 9 L", "FreeSerif",
2128         "Hoefler Text", Times, "Times New Roman", serif;
2129     font-variant: small-caps ;
2130     font-weight: normal ;
2131     color: #304070 ;
2132     text-shadow: 2px 2px 3px #808080;
2133 }
2134
2135 h1 { /* title of the entire website, used on each page */
2136     font-variant: small-caps ;
2137     color: #304070 ;
2138     text-shadow: 2px 2px 3px #808080;
2139     background-color: #F7F7F0 ;
2140     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
2141 }
2142
2143 h1 {
2144     border-bottom: 1px solid #304070;
2145     border-top: 2px solid #304070;
2146 }
2147
2148 h2 {
2149     border-bottom: 1px solid #304070;
2150     border-top: 2px solid #304070;
2151     background-color: #F7F7F0 ;
2152     background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);

```

```
2153 }
2154
2155
2156
2157 div.abstract {
2158     background: #f5f5eb ;
2159     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2160
2161     border: 1px solid silver;
2162     border-radius: 1em ;
2163 }
2164
2165 div.abstract dl {line-height:1.5;}
2166 div.abstract dt {color:#304070;}
2167
2168 div.abstracttitle{
2169     font-family: "URW Classico", Optima, "Linux Biolinum O",
2170         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2171         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2172     font-weight:bold;
2173     font-variant: small-caps ;
2174     font-size:1.5em;
2175     border-bottom: 1px solid silver ;
2176     color: #304070 ;
2177     text-align: center ;
2178     text-shadow: 1px 1px 2px #808080;
2179 }
2180
2181 span.abstracrunintitle{
2182     font-family: "URW Classico", Optima, "Linux Biolinum O",
2183         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2184         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2185     font-weight:bold;
2186 }
2187
2188
2189 div.epigraph {
2190     background: #f5f5eb ;
2191     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2192
2193     border: 1px solid silver ;
2194     border-radius: 1ex ;
2195     box-shadow: 3px 3px 3px #808080 ;
2196 }
2197
2198
2199 .example {
2200     background-color: #f5f5eb ;
2201     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2202
```



```
2203 }
2204
2205 div.exampletitle{
2206     font-family: "URW Classico", Optima, "Linux Biolinum O",
2207         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2208         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2209     font-weight:bold;
2210     font-variant: small-caps ;
2211     border-bottom: 1px solid silver ;
2212     color: #304070 ;
2213     text-align: center ;
2214     text-shadow: 1px 1px 2px #808080;
2215 }
2216
2217
2218 .sidebar {
2219     background-color: #f5f5eb ;
2220     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2221
2222 }
2223
2224 div.sidebartitle{
2225     font-family: "URW Classico", Optima, "Linux Biolinum O",
2226         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2227         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2228     font-weight:bold;
2229     font-variant: small-caps ;
2230     border-bottom: 1px solid silver ;
2231     color: #304070 ;
2232     text-align: center ;
2233     text-shadow: 1px 1px 2px #808080;
2234 }
2235
2236
2237 .fancyvrblabel {
2238     font-family: "URW Classico", Optima, "Linux Biolinum O",
2239         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2240         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2241     font-weight:bold;
2242     font-variant: small-caps ;
2243     font-size: 1.5em ;
2244     color: #304070 ;
2245     text-align: center ;
2246     text-shadow: 1px 1px 2px #808080;
2247 }
2248
2249
2250
2251 div.minipage {
2252     background-color: #eeeeee7 ;
```

```
2253     border: 1px solid silver ;
2254     border-radius: 1ex ;
2255 }
2256
2257 section.textbody > div.minipage {
2258     box-shadow: 3px 3px 3px #808080 ;
2259 }
2260
2261 div.fboxBlock div.minipage { box-shadow: none ; }
2262
2263 .framed .minipage , .framedleftbar .minipage {
2264     border: none ;
2265     background: none ;
2266     padding: 0ex ;
2267     margin: 0ex ;
2268 }
2269
2270 figure.figure .minipage, figcaption .minipage { border: none; }
2271
2272 div.marginblock div.minipage { border: none; }
2273
2274 figure , div.marginblock {
2275     background-color: #eeeeee7 ;
2276     border: 1px solid silver ;
2277     border-radius: 1ex ;
2278     box-shadow: 3px 3px 3px #808080 ;
2279 }
2280
2281 figure figure {
2282     border: 1px solid silver ;
2283     margin: 0em ;
2284     box-shadow: none ;
2285 }
2286
2287 /*
2288 figcaption {
2289     border-top: 1px solid silver ;
2290     border-bottom: 1px solid silver ;
2291     background-color: #e8e8e8 ;
2292 }
2293 */
2294
2295
2296 div.table {
2297     box-shadow: 3px 3px 3px #808080 ;
2298 }
2299
2300 /*
2301 .tnotes {
2302     background: #e8e8e8;
```

```
2303     border: 1px solid silver;
2304 }
2305 */
2306
2307
2308 nav.topnavigation{
2309     background-color: #b0b8b0 ;
2310     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
2311 }
2312
2313 nav.botnavigation{
2314     background-color: #b0b8b0 ;
2315     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
2316 }
2317
2318
2319
2320 header{
2321     background-color: #F7F7F0 ;
2322     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
2323 }
2324
2325 footer{
2326     background-color: #F7F7F0 ;
2327     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
2328 }
2329
2330
2331
2332 nav.sidetoc {
2333     background-color: #F7F7F0 ;
2334     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
2335     box-shadow: 3px 3px 3px #808080 ;
2336     border-radius: 0px 0px 0px 20px ;
2337 }
2338
2339 div.sidetoc title {color: #304070 ; }
2340
2341 nav.sidetoc a: hover {
2342     color: #006000 ;
2343     text-decoration: none ;
2344     text-shadow: 0px 0px 2px #a0a0a0;
2345 }
2346
2347
2348 @media screen and (max-width: 45em) {
2349     nav.sidetoc { border-radius: 0 ; }
2350 }
2351
2352
```

```

2353 \end{filecontents*}
2354 % \end{Verbatim}% for syntax highlighting
2355 \end{warpprint}

```

### 31.6 lwarp\_formal.css

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

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

```

2356 \begin{warpprint}
2357 \begin{filecontents*}{lwarp_formal.css}
2358 @import url("lwarp.css") ;
2359
2360
2361
2362 A:link {color:#802020 ; text-decoration:none; }
2363 A:visited {color:#802020 ; text-shadow:none ;}
2364 A:hover {color:#400000 ; text-shadow:none ;}
2365 A:active {color:#C00000 ; text-shadow:none ;}
2366
2367
2368 body {
2369     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2370         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2371         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2372         "Times New Roman", serif;
2373     background: #fffcf5;
2374 }
2375
2376 span.textrm {
2377     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2378         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2379         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2380         "Times New Roman", serif;
2381 }
2382
2383 span.textsf {
2384     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2385         Geneva, Verdana, sans-serif ;
2386 }
2387
2388
2389
2390 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2391 {

```

```
2392     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2393         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2394         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2395         "Times New Roman", serif;
2396     color: #800000 ;
2397     text-shadow: none ;
2398 }
2399
2400 h1, h2 {
2401     background-color: #fffcf5 ;
2402     background-image: none ;
2403     border-bottom: 1px solid #808080;
2404     border-top: 2px solid #808080;
2405 }
2406
2407 div.abstracttitle {
2408     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2409         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2410         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2411         "Times New Roman", serif;
2412     color: black ;
2413     text-shadow: none ;
2414 }
2415
2416 span.abstracrunintitle {
2417     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2418         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2419         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2420         "Times New Roman", serif;
2421     color: black ;
2422     text-shadow: none ;
2423 }
2424
2425 div.abstract { font-size: 100% }
2426
2427 .sidebar {
2428     background: #fffcf5;
2429     background-image: none ;
2430     margin: 2em 5% 2em 5%;
2431     padding: 0.5em 1em;
2432     border: none ;
2433     border-top : 1px solid silver;
2434     border-bottom : 1px solid silver;
2435     font-size: 90% ;
2436 }
2437
2438 div.sidebartitle{
2439     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2440         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2441         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
```

```
2442         "Times New Roman", serif;
2443     color: #800000 ;
2444     text-shadow: none ;
2445     border: none ;
2446 }
2447
2448 .example {
2449     background: #fffcf5;
2450     background-image: none ;
2451     margin: 2em 5% 2em 5%;
2452     padding: 0.5em 1em;
2453     border: none ;
2454     border-top : 1px solid silver;
2455     border-bottom : 1px solid silver;
2456 }
2457
2458 div.exampletitle{
2459     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2460         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2461         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2462         "Times New Roman", serif;
2463     color: #800000 ;
2464     text-shadow: none ;
2465     border: none ;
2466 }
2467
2468 div.fancyvrblabel{
2469     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2470         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2471         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2472         "Times New Roman", serif;
2473     color: #800000 ;
2474     text-shadow: none ;
2475     border: none ;
2476 }
2477
2478
2479
2480 .verse {
2481     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2482         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2483         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2484         "Times New Roman", serif;
2485 }
2486
2487
2488 figure {
2489     margin: 3ex 5% 3ex 5% ;
2490     padding: 1ex 1em 1ex 1em ;
2491     background-color: #fffcf5 ;
```

```
2492     overflow-x: auto ;
2493     border: none ;
2494 /*     border-top: 1px solid silver; */
2495 /*     border-bottom: 1px solid silver; */
2496 }
2497
2498
2499 figcaption , .lstlisting {
2500     border: none ;
2501 /*     border-top: 1px solid silver ; */
2502 /*     border-bottom: 1px solid silver ; */
2503     background-color: #ffffcf5 ;
2504 }
2505
2506 .tnotes {
2507     background: #ffffcf5 ;
2508 }
2509
2510 .theorem {
2511     background: none ;
2512 }
2513
2514 .minipage {
2515     background-color: #ffffcf5 ;
2516     border: none ;
2517 }
2518
2519 div.floatrow figure { border: none ; }
2520
2521 figure figure { border: none ; }
2522
2523
2524 nav.toc, nav.lof, nav.lot, nav.lol {
2525     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2526         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2527         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2528         "Times New Roman", serif;
2529 }
2530
2531 nav.sidetoc {
2532     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2533         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2534         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2535         "Times New Roman", serif;
2536     background-image: linear-gradient(to bottom, #ffffcf5, #C0C0C0);
2537     border-radius: 0px 0px 0px 20px ;
2538 }
2539
2540 div.sidetoc{
2541     color: #800000 ;
```

```

2542 }
2543
2544 header{
2545     background-color: #e0e0e0 ;
2546     background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
2547     text-align:center ;
2548 }
2549
2550 footer{
2551     background-color: #e0e0e0 ;
2552     background-image: linear-gradient(to bottom, #fffcf5, #b0b0b0);
2553     padding: 2ex 1em 2ex 1em ;
2554     clear:right ;
2555     text-align:left ;
2556 }
2557
2558 nav.botnavigation {
2559     background: #dedcd5 ;
2560     border-top: 1px solid black ;
2561 }
2562 \end{filecontents*}
2563 % \end{Verbatim}% for syntax highlighting
2564 \end{warpprint}

```

### 31.7 sample\_project.css

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

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

```

2565 \begin{warpprint}
2566 \begin{filecontents*}{sample_project.css}
2567 /* ( --- Start of project.css --- ) */
2568 /* ( --- A sample project-specific CSS file for lwarp --- ) */
2569
2570 /* Load default lwarp settings: */
2571 @import url("lwarp.css") ;
2572 /* or lwarp_formal.css, lwarp_sagebrush.css */
2573
2574 /* Project-specific CSS setting follow here. */
2575 /* . . . */
2576
2577 /* ( --- End of project.css --- ) */
2578 \end{filecontents*}
2579 % \end{Verbatim}% for syntax highlighting
2580 \end{warpprint}

```



### 31.8 lwarp.xdy

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

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

```

2581 \begin{warpprint}
2582 \begin{filecontents*}{lwarp.xdy}
2583 (require "tex/inputenc/latin.xdy")
2584 (merge-rule "\\PS *" "Postscript")
2585 (require "texindy.xdy")
2586 (require "page-ranges.xdy")
2587 (require "book-order.xdy")
2588 (markup-locref :open "\hyperindexref{" :close "}")
2589 \end{filecontents*}
2590 % \end{Verbatim}% for syntax highlighting
2591 \end{warpprint}

```

### 31.9 lwarp\_mathjax.txt

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

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

```

2592 \begin{warpprint}
2593 \begin{filecontents*}{lwarp_mathjax.txt}
2594 <!-- https://groups.google.com/forum/#!topic/
2595                               mathjax-users/jUtewUcE2bY -->
2596 <script type="text/x-mathjax-config">
2597 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
2598     var seteqsectionDefault = {name: "", num: 0};
2599     var seteqsections = {}, seteqsection = seteqsectionDefault;
2600     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
2601     var AMS = MathJax.Extension["TeX/AMSmath"];
2602     TEX.Definitions.Add({
2603     macros: {
2604         seteqsection: "mySection",
2605         seteqnumber: "mySetEqNumber"
2606     }
2607     });
2608
2609     PARSE.Augment({
2610     mySection: function (name) {
2611         seteqsection.num = AMS.number;

```

```

2612     var n = this.GetArgument(name);
2613     if (n === "") {
2614         seteqsection = seteqsectionDefault;
2615     } else {
2616         if (!seteqsections["_"+n])
2617             seteqsections["_"+n] = {name:n, num:0};
2618         seteqsection = seteqsections["_"+n];
2619     }
2620     AMS.number = seteqsection.num;
2621 },
2622 mySetEqNumber: function (name) {
2623     var n = this.GetArgument(name);
2624     if (!n || !n.match(/^ *[0-9]+ *$/))
2625         n = ""; else n = parseInt(n)-1;
2626     <!-- $ syntax highlighting -->
2627     if (n === "" || n < 1)
2628         TEX.Error
2629             ("Argument to "+name+" should be a positive integer");
2630     AMS.number = n;
2631 }
2632 });
2633 MathJax.Hub.Config({
2634   TeX: {
2635     equationNumbers: {
2636       formatTag: function (n)
2637         {return "("+(seteqsection.name+"."+n).replace(/\./,"")+")"},
2638       formatID: function (n) {
2639         n = (seteqsection.name+'.'+n).replace
2640           (/[<>']/g,"").replace(/\./,"");
2641         return 'mjax-eqn-' + n;
2642       }
2643     }
2644   }
2645 });
2646 });
2647 </script>
2648
2649 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
2650 <script type="text/x-mathjax-config">
2651   MathJax.Ajax.config.path["Contrib"] =
2652     "https://cdn.mathjax.org/mathjax/contrib";
2653 </script>
2654
2655 <!-- https://github.com/burnpanck/MathJax-siunitx -->
2656
2657 <script type="text/x-mathjax-config">
2658   MathJax.Hub.Config({
2659     extensions: ["tex2jax.js","siunitx/siunitx.js"],
2660     jax: ["input/TeX","output/HTML-CSS"],
2661     tex2jax: {inlineMath: [["$","$"],["\\(","\\)"]]},

```

```

2662   TeX: {extensions: ["AMSmath.js", "AMSsymbols.js", "sinuitx.js"]}
2663   });
2664   MathJax.Ajax.config.path['siunitx'] = 'http://rawgit.com/burnpanck/MathJax-siunitx/master/';
2665 </script>
2666
2667 <script type="text/x-mathjax-config">
2668 MathJax.Hub.Config({
2669   TeX: {
2670     equationNumbers: {
2671       autoNumber: "AMS"
2672     }
2673   }
2674 });
2675 </script>
2676
2677 <!-- Alternative CDN provider: -->
2678 <script type="text/javascript" async
2679 src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?config=TeX-AMS_HTML-full">
2680 </script>
2681
2682 <!-- No longer supported after April 30, 2017: -->
2683 <!--
2684 <script
2685   src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
2686 </script>
2687 -->
2688
2689 \end{filecontents*}
2690 % \end{Verbatim}% for syntax highlighting
2691 \end{warpprint}

```

### 31.10 lwarpmk option

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

```

2692 \begin{LWR@create!lwarpmk}

```

Prog lwarpmk Creates a local copy of lwarpmk:

```

2693 \begin{filecontents*}{lwarpmk.lua}
2694 #!/usr/bin/env texlua
2695
2696 -- Copyright 2016-2017 Brian Dunn
2697
2698 -- Print the usage of the lwarpmk command:
2699
2700 printversion = "v0.43"

```

```
2701
2702 function printhelp ()
2703 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
2704 end
2705
2706 function printusage ()
2707 print ( [[
2708
2709 lwarpmk print [project]: Compile the print version if necessary.
2710 lwarpmk print1 [project]: Forced single compile of the print version.
2711 lwarpmk printindex [project]: Process the index for the print version.
2712 lwarpmk printglossary [project]: Process the glossary for the print version.
2713 lwarpmk html [project]: Compile the HTML version if necessary.
2714 lwarpmk html1 [project]: Forced single compile of the HTML version.
2715 lwarpmk htmlindex [project]: Process the index for the html version.
2716 lwarpmk htmlglossary [project]: Process the glossary for the html version.
2717 lwarpmk again [project]: Touch the source code to trigger recompiles.
2718 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
2719 lwarpmk pdftohtml [project]:
2720     For use with latexmk or a Makefile:
2721     Convert project_html.pdf to project_html.html and
2722     individual HTML files.
2723 lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, *_html_inc.*, .gl*
2724 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
2725 lwarpmk -h: Print this help message.
2726 lwarpmk --help: Print this help message.
2727
2728 ]] )
2729 printconf ()
2730 end
2731
2732 -- Print the format of the configuration file lwarpmk.conf:
2733
2734 function printconf ()
2735 print ( [[
2736 An example lwarpmk.conf or <project>.lwarpmkconf project file:
2737 --
2738 opsystem = "Unix"      (or "Windows")
2739 latexname = "pdflatex" (or "lualatex", or "xelatex")
2740 sourcename = "projectname" (the source-code filename w/o .tex)
2741 homehtmlfilename = "index" (or perhaps the project name)
2742 htmlfilename = "" (or "projectname" - filename prefix)
2743 latexmk = "false" (or "true" to use latexmk to build PDFs)
2744 language = "english" (use a language supported by xindy)
2745 xdyfile = "lwarp.xdy" (or a custom file based on lwarp.xdy)
2746 --
2747 Filenames must contain only letters, numbers, underscore, or dash.
2748 Values must be in "quotes".
2749
2750 ]] ) ;
```

```
2751 end
2752
2753
2754 -- Split one large sourcefile into a number of files,
2755 -- starting with destfile.
2756 -- The file is split at each occurrence of <!--|Start file|newfilename|*
2757
2758 function splitfile (destfile,sourcefile)
2759 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
2760 local sfile = io.open(sourcefile)
2761 io.output(destfile)
2762 for line in sfile:lines() do
2763 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
2764 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
2765 io.output(newfilename) ;
2766 else -- not a splitpoint
2767 io.write (line .. "\n") ;
2768 end
2769 end -- do
2770 io.close(sfile)
2771 end -- function
2772
2773 -- Incorrect value, so print an error and exit.
2774
2775 function cvalueerror ( line, linenum , cvalue )
2776     print ( linenum .. " : " .. line ) ;
2777     print ("lwarpmk: incorrect variable value \"" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
2778     printconf () ;
2779     os.exit(1) ;
2780 end
2781
2782 -- Load settings from the project's "lwarpmk.conf" file:
2783
2784 function loadconf ()
2785 -- Default configuration filename:
2786 local conffile = "lwarpmk.conf"
2787 -- Optional configuration filename:
2788 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
2789 -- Default language:
2790 language = "english"
2791 -- Default xdyfile:
2792 xdyfile = "lwarp.xdy"
2793 -- Verify the file exists:
2794 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
2795 print("lwarpmk: " .. conffile .. " does not exist.")
2796 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
2797 printhelp () ;
2798 os.exit(1) -- exit the entire lwarpmk script
2799 else -- file exists
2800 -- Read the file:
```

```

2801 print ("lwarpmk: Reading " .. conffile .. ".")
2802 local cfile = io.open(conffile)
2803 -- Scan each line:
2804 local linenum = 0
2805 for line in cfile:lines() do -- scan lines
2806 linenum = linenum + 1
2807 i,j,cvarname,cvalue = string.find (line,"([%w-_]*)%s*=%s*\"([%w%-_.]*)\"") ;
2808 -- Error if incorrect enclosing characters:
2809 if ( i == nil ) then
2810 print ( linenum .. " : " .. line ) ;
2811 print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
2812 printconf () ;
2813 os.exit(1) ;
2814 end
2815 if ( cvarname == "opssystem" ) then
2816     -- Verify choice of opssystem:
2817     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
2818         opssystem = cvalue
2819     else
2820         cvalueerror ( line, linenum , cvalue )
2821     end
2822 elseif ( cvarname == "latexname" ) then
2823     -- Verify choice of LaTeX compiler:
2824     if (
2825         (cvalue == "pdflatex") or
2826         (cvalue == "xelatex") or
2827         (cvalue == "lualatex")
2828     ) then
2829         latexname = cvalue
2830     else
2831         cvalueerror ( line, linenum , cvalue )
2832     end
2833 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
2834 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
2835 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
2836 elseif ( cvarname == "latexmk" ) then latexmk = cvalue
2837 elseif ( cvarname == "language" ) then language = cvalue
2838 elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue
2839 else
2840 print ( linenum .. " : " .. line ) ;
2841 print ("lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. conffile .. ".\n" ) ;
2842 printconf () ;
2843 os.exit(1) ;
2844 end
2845 end -- do scan lines
2846 io.close(cfile)
2847 end -- file exists
2848 -- Select some operating-system commands:
2849 if opssystem=="Unix" then -- For Unix / Linux / Mac OS:
2850 rmname = "rm"

```

```
2851 mvname = "mv"
2852 touchnamepre = "touch"
2853 touchnamepost = ""
2854 dirslash = "/"
2855 opquote= "\"'"
2856 elseif opsystem=="Windows" then -- For Windows
2857 rmname = "DEL"
2858 mvname = "MOVE"
2859 touchnamepre = "COPY /b"
2860 touchnamepost = "+,,"
2861 dirslash = "\\"
2862 opquote= "\""
2863 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
2864 end --- for Windows
2865
2866 -- set xindycmd according to pdflatex vs xelatex/lualatex:
2867 if ( latexname == "pdflatex" ) then
2868 xindycmd = "texindy -C utf8"
2869 glossarycmd = "xindy -C utf8"
2870 else
2871 xindycmd = "xindy -M texindy -C utf8"
2872 glossarycmd = "xindy -C utf8"
2873 end
2874
2875 end -- loadconf
2876
2877
2878 function refreshdate ()
2879 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
2880 end
2881
2882
2883 -- Scan the LaTeX log file for the phrase "Rerun to get",
2884 -- indicating that the file should be compiled again.
2885 -- Return true if found.
2886
2887 function reruntoget (filesorce)
2888 local fsource = io.open(filesorce)
2889 for line in fsource:lines() do
2890 if ( string.find(line,"Rerun to get") ~= nil ) then
2891     io.close(fsource)
2892     return true
2893 end
2894 end
2895 io.close(fsource)
2896 return false
2897 end
2898
2899
2900 -- Compile one time, return true if should compile again.
```

```
2901-- fsuffix is "" for print, "_html" for HTML output.
2902
2903function onetime (fsuffix)
2904print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
2905err = os.execute(
2906--      "echo " ..
2907      latexname .. " " .. sourcename..fsuffix )
2908if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2909return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
2910end
2911
2912
2913-- Compile up to five times.
2914-- fsuffix is "" for print, "_html" for HTML output
2915
2916function manytimes (fsuffix)
2917if onetime(fsuffix) == true then
2918if onetime(fsuffix) == true then
2919if onetime(fsuffix) == true then
2920if onetime(fsuffix) == true then
2921if onetime(fsuffix) == true then
2922end end end end end
2923end
2924
2925-- Exit if the given file does not exist.
2926
2927function verifyfileexists (filename)
2928if (lfs.attributes ( filename , "modification" ) == nil ) then
2929print ( "lwarpmk: " .. filename .. " not found." ) ;
2930os.exit (1) ;
2931end
2932end
2933
2934
2935-- Convert <project>_html.pdf into HTML files:
2936
2937function pdftohtml ()
2938    -- Convert to text:
2939    print ("lwarpmk: Converting " .. sourcename
2940        .. "_html.pdf to " .. sourcename .. "_html.html")
2941    os.execute("pdftotext -enc UTF-8 -npgbrk -layout "
2942        .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
2943    -- Split the result into individual HTML files:
2944    splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
2945end
2946
2947
2948-- Remove auxiliary files:
2949
2950function removeaux ()
```



```

2951     os.execute ( rmname .. " " ..
2952         sourcename .. ".aux " .. sourcename .. "_html.aux " ..
2953         sourcename .. ".toc " .. sourcename .. "_html.toc " ..
2954         sourcename .. ".lof " .. sourcename .. "_html.lof " ..
2955         sourcename .. ".lot " .. sourcename .. "_html.lot " ..
2956         sourcename .. ".idx " .. sourcename .. "_html.idx " ..
2957         sourcename .. ".ind " .. sourcename .. "_html.ind " ..
2958         sourcename .. ".log " .. sourcename .. "_html.log " ..
2959         sourcename .. ".gl* " .. sourcename .. "_html.gl* " ..
2960         "*_html_inc.*"
2961     )
2962 end
2963
2964
2965
2966 -- Create lateximages based on lateximages.txt:
2967 function createlateximages ()
2968 print ("lwarpmk: Creating lateximages.")
2969 local limagesfile = io.open("lateximages.txt")
2970 -- Create the lateximages directory, ignore error if already exists
2971 err = os.execute("mkdir lateximages")
2972 -- Scan lateximages.txt
2973 for line in limagesfile:lines() do
2974 -- lwimgpage is the page number in the PDF which has the image
2975 -- lwimgnum is the sequential lateximage number to assign for the image
2976 i,j,lwimgpage,lwimgnum = string.find (line,"|(.*)|(.*)|")
2977 -- For each entry:
2978 if ( i~=nil ) then
2979 -- Separate out the image into its own single-page pdf:
2980 err = os.execute(
2981 "pdfseparate -f " .. lwimgpage .. " -l " ..
2982 lwimgpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
2983 -- Crop the image:
2984 err = os.execute(
2985 "pdfcrop -- hires lateximagetemp-" .. lwimgpage .. ".pdf lateximage-" .. lwimgnum .. ".pdf")
2986 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2987 -- Convert the image to svg:
2988 err = os.execute(
2989 "pdftocairo -svg lateximage-" .. lwimgnum .. ".pdf lateximage-" .. lwimgnum .. ".svg")
2990 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2991 -- Move the result into lateximages/:
2992 err = os.execute(
2993 mvname .. " lateximage-" .. lwimgnum .. ".svg lateximages" .. dirslash )
2994 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2995 -- Remove the temporary files:
2996 err = os.execute(
2997 rmname .. " lateximage-" .. lwimgnum .. ".pdf lateximagetemp-" .. lwimgpage .. ".pdf")
2998 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2999 end
3000 end -- do

```

```

3001 io.close(limagesfile)
3002 end -- function
3003
3004
3005 -- Use latexmk to compile source and index:
3006 -- fsuffix is "" for print, or "_html" for HTML
3007 function compilelatexmk ( fsuffix )
3008     -- The recorder option is required to detect changes in <project>.tex
3009     -- while we are loading <project>_html.tex.
3010     err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
3011         .. "-e "
3012         .. opquote
3013         .. "$makeindex = q/" -- $
3014         .. xindycmd
3015         .. " -M " .. xdyfile
3016         .. " -L " .. language .. " /"
3017         .. opquote
3018         .. " -pdflatex=\"\" .. latexname .. " %O %S\" "
3019         .. sourcename..fsuffix ..".tex" ) ;
3020     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
3021 end
3022
3023
3024
3025 -- lwarpmk --version :
3026
3027 if (arg[1] == "--version") then
3028     print ( "lwarpmk: " .. printversion )
3029
3030 else -- not --version
3031
3032 -- print intro:
3033
3034 print ("lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package.")
3035
3036 -- lwarpmk print:
3037
3038 if arg[1] == "print" then
3039     loadconf ()
3040     if ( latexmk == "true" ) then
3041         compilelatexmk ("" )
3042         print ("lwarpmk: Done.")
3043     else -- not latexmk
3044         verifyfileexists (sourcename .. ".tex") ;
3045         -- See if up to date:
3046         if (
3047             ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
3048             (
3049                 lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3050                 lfs.attributes ( sourcename .. ".pdf" , "modification" )

```

```
3051     )
3052   ) then
3053     -- Recompile if not yet up to date:
3054     manytimes("")
3055     print ("lwarpmk: Done.") ;
3056   else
3057     print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
3058   end
3059 end -- not latexmk
3060
3061 elseif arg[1] == "print1" then
3062   loadconf ()
3063   verifyfileexists (sourcename .. ".tex") ;
3064   onetime("")
3065   print ("lwarpmk: Done.") ;
3066
3067 -- lwarp printindex:
3068 -- Compile the index then touch the source
3069 -- to trigger a recompile of the document:
3070
3071 elseif arg[1] == "printindex" then
3072   loadconf ()
3073   print ("lwarpmk: Processing the index.")
3074   os.execute(
3075     xindycmd
3076     .. " -M " .. xdyfile
3077     .. " -L " .. language
3078     .. " " .. sourcename .. ".idx")
3079   print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3080   refreshdate ()
3081   print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3082   print ("lwarpmk: Done.")
3083
3084 -- lwarp printglossary:
3085 -- Compile the glossary then touch the source
3086 -- to trigger a recompile of the document:
3087
3088 elseif arg[1] == "printglossary" then
3089   loadconf ()
3090   print ("lwarpmk: Processing the glossary.")
3091
3092   os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
3093     " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
3094     .. sourcename .. ".glo")
3095   print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3096   refreshdate ()
3097   print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3098   print ("lwarpmk: Done.")
3099
3100 -- lwarpmk html:
```

```
3101
3102 elseif arg[1] == "html" then
3103   loadconf ()
3104   if ( latexmk == "true" ) then
3105     compilelatexmk ("_html")
3106     pdftohtml ()
3107     print ("lwarpmk: Done.")
3108   else -- not latexmk
3109     verifyfileexists ( sourcename .. ".tex" ) ;
3110     -- See if exists and is up to date:
3111     if (
3112       ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
3113       (
3114         lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3115         lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
3116       )
3117     ) then
3118       -- Recompile if not yet up to date:
3119       manytimes("_html")
3120       pdftohtml ()
3121       print ("lwarpmk: Done.")
3122     else
3123       print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
3124     end
3125   end -- not latexmk
3126
3127 elseif arg[1] == "html1" then
3128   loadconf ()
3129   verifyfileexists ( sourcename .. ".tex" ) ;
3130   onetime("_html")
3131   pdftohtml ()
3132   print ("lwarpmk: Done.")
3133
3134 elseif arg[1] == "pdftohtml" then
3135   loadconf ()
3136   pdftohtml ()
3137
3138 -- lwarpmk htmlindex:
3139 -- Compile the index then touch the source
3140 -- to trigger a recompile of the document:
3141
3142 elseif arg[1] == "htmlindex" then
3143   loadconf ()
3144   print ("lwarpmk: Processing the index.")
3145   os.execute(
3146     xindycmd
3147     .. " -M " .. xdyfile
3148     .. " -L " .. language
3149     .. " " .. sourcename .. "_html.idx"
3150 )
```

```
3151 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3152 refreshdate ()
3153 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3154 print ("lwarpmk: Done.")
3155
3156 -- lwarpmk htmlglossary:
3157 -- Compile the glossary then touch the source
3158 -- to trigger a recompile of the document:
3159
3160 elseif arg[1] == "htmlglossary" then
3161 loadconf ()
3162 print ("lwarpmk: Processing the glossary.")
3163
3164 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
3165 " _html -t " .. sourcename .. " _html.glg -o " .. sourcename ..
3166 " _html.gls " .. sourcename .. " _html.glo")
3167
3168 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3169 refreshdate ()
3170 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3171 print ("lwarpmk: Done.")
3172
3173 -- lwarpmk limages:
3174 -- Scan the lateximages.txt file to create lateximages,
3175 -- then touch the source to trigger a recompile.
3176
3177 elseif arg[1] == "limages" then
3178 loadconf ()
3179 print ("lwarpmk: Processing images.")
3180 createlateximages ()
3181 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3182 refreshdate ()
3183 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3184 print ("lwarpmk: Done.")
3185
3186 -- lwarpmk again:
3187 -- Touch the source to trigger a recompile.
3188
3189 elseif arg[1] == "again" then
3190 loadconf ()
3191 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3192 refreshdate ()
3193 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3194 print ("lwarpmk: Done.")
3195
3196 -- lwarpmk clean:
3197 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
3198
3199 elseif arg[1] == "clean" then
3200 loadconf ()
```

```

3201 removeaux ()
3202 print ("lwarpmk: Done.")
3203
3204 -- lwarpmk cleanall
3205 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
3206 --    and also project.pdf, *.html
3207
3208 elseif arg[1] == "cleanall" then
3209 loadconf ()
3210 removeaux ()
3211 os.execute ( rmname .. " " ..
3212     sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
3213     "*.html"
3214     )
3215 print ("lwarpmk: Done.")
3216
3217 -- lwarpmk with no argument :
3218
3219 elseif (arg[1] == nil) then
3220 printhelp ()
3221
3222 -- lwarpmk -h or lwarpmk --help :
3223
3224 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
3225 printusage ()
3226
3227 else
3228 print ("lwarpmk: Unknown command \""..arg[1].."\".\n")
3229 printhelp ()
3230 end
3231
3232 end -- not --version
3233 \end{filecontents*}
3234 % \end{Verbatim}% for syntax highlighting
3235 \end{LWR@createlwarpmk}

```

## 32 Stacks

for HTML output: 3236 \begin{warpHTML}



Stacks are used to remember how to close sections and list items. Before a new section is started, previously nested sections and items must be closed out (un-nested) in proper order. Note that starting a new section may close several levels of previously nested items at the same time. For example, starting a new `\section` would close any currently open subsection, subsubsection, and paragraph. General environ-

ments are not nested on the stack since they have their own close mechanism. List environments are nested, and items inside those environments are nested one level deeper still. List environments may be nested inside other list environments, and list items are nested inside list environments as well. Thus, the stack may have items which are not necessarily in order, since a description may contain an enumerate, for example. Depths to be recorded in `\LWR@closedepthone`, etc.

### 32.1 Assigning depths

initial depths for empty stack entries:

```
3237 \newcommand*{\LWR@depthnone}{-5}
```

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

```
3238 \newcommand*{\LWR@depthfinished}{-4}
3239 \newcommand*{\LWR@depthpart}{-1}
3240 \newcommand*{\LWR@depthchapter}{0}
3241 \newcommand*{\LWR@depthsection}{1}
3242 \newcommand*{\LWR@depthsubsection}{2}
3243 \newcommand*{\LWR@depthsubsubsection}{3}
3244 \newcommand*{\LWR@depthparagraph}{4}
3245 \newcommand*{\LWR@depthsubparagraph}{5}
```

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

```
3246 \newcommand*{\LWR@depthlist}{6}
```

used by `\item`:

```
3247 \newcommand*{\LWR@depthlistitem}{7}
```

### 32.2 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.

```
3248 \newcommand*{\LWR@closeone}{}% top of the stack
3249 \newcommand*{\LWR@closetwo}{}
3250 \newcommand*{\LWR@closeethree}{}
3251 \newcommand*{\LWR@closefour}{}
3252 \newcommand*{\LWR@closefive}{}
3253 \newcommand*{\LWR@closesix}{}%
```

```

3254 \newcommand*{\LWR@closeseven}{\LWR@depthnone}
3255 \newcommand*{\LWR@closeeight}{\LWR@depthnone}
3256 \newcommand*{\LWR@closenine}{\LWR@depthnone}
3257 \newcommand*{\LWR@closeten}{\LWR@depthnone}
3258 \newcommand*{\LWR@closeeleven}{\LWR@depthnone}
3259 \newcommand*{\LWR@closetwelve}{\LWR@depthnone}

```

### 32.3 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}

```

---

```

3260 \newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
3261 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
3262 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
3263 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
3264 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
3265 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
3266 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
3267 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
3268 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
3269 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}
3270 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
3271 \newcommand*{\LWR@closedephtwelve}{\LWR@depthnone}

```

### 32.4 Pushing and popping the stack

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

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

```

3272 \NewDocumentCommand{\pushclose}{m m}

```



```

3273 {
3274 \let\LWR@closetwelve\LWR@closeeleven
3275 \let\LWR@closeeleven\LWR@closeten
3276 \let\LWR@closeten\LWR@closenine
3277 \let\LWR@closenine\LWR@closeeight
3278 \let\LWR@closeeight\LWR@closeseven
3279 \let\LWR@closeseven\LWR@closesix
3280 \let\LWR@closesix\LWR@closefive
3281 \let\LWR@closefive\LWR@closefour
3282 \let\LWR@closefour\LWR@closethree
3283 \let\LWR@closethree\LWR@closetwo
3284 \let\LWR@closetwo\LWR@closeone
3285 \let\LWR@closeone#1
3286 \let\LWR@closedepthtwelve\LWR@closedeptheleven
3287 \let\LWR@closedeptheleven\LWR@closedephten
3288 \let\LWR@closedephten\LWR@closedepthnine
3289 \let\LWR@closedepthnine\LWR@closedeptheight
3290 \let\LWR@closedeptheight\LWR@closedepthseven
3291 \let\LWR@closedepthseven\LWR@closedepthsix
3292 \let\LWR@closedepthsix\LWR@closedepthfive
3293 \let\LWR@closedepthfive\LWR@closedepthfour
3294 \let\LWR@closedepthfour\LWR@closedepththree
3295 \let\LWR@closedepththree\LWR@closedephtwo
3296 \let\LWR@closedephtwo\LWR@closedepthone
3297 \let\LWR@closedepthone#2
3298 }

```

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

```

3299 \newcommand*{\popclose}
3300 {
3301 \let\LWR@closeone\LWR@closetwo
3302 \let\LWR@closetwo\LWR@closethree
3303 \let\LWR@closethree\LWR@closefour
3304 \let\LWR@closefour\LWR@closefive
3305 \let\LWR@closefive\LWR@closesix
3306 \let\LWR@closesix\LWR@closeseven
3307 \let\LWR@closeseven\LWR@closeeight
3308 \let\LWR@closeeight\LWR@closenine
3309 \let\LWR@closenine\LWR@closeten
3310 \let\LWR@closeten\LWR@closeeleven
3311 \let\LWR@closeeleven\LWR@closetwelve
3312 \let\LWR@closedepthone\LWR@closedephtwo
3313 \let\LWR@closedephtwo\LWR@closedepththree
3314 \let\LWR@closedepththree\LWR@closedepthfour
3315 \let\LWR@closedepthfour\LWR@closedepthfive
3316 \let\LWR@closedepthfive\LWR@closedepthsix
3317 \let\LWR@closedepthsix\LWR@closedepthseven
3318 \let\LWR@closedepthseven\LWR@closedeptheight

```

```

3319 \let\LWR@closedeptheight\LWR@closedepthnine
3320 \let\LWR@closedepthnine\LWR@closedepthten
3321 \let\LWR@closedepthten\LWR@closedeptheleven
3322 \let\LWR@closedeptheleven\LWR@closedephtwelve
3323 }

3324 \end{warpHTML}

```

### 33 Data arrays

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

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

To assign an empty value:

```
\LWR@setexpararray{name}{index}{}
```

**for HTML output:** 3325 \begin{warpHTML}

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

```

3326 \NewDocumentCommand{\LWR@setexpararray}{m m m}{%
3327 \ifthenelse{\isempty{#3}}{%
3328 {\csdef{#1#2}{}}%
3329 {\expandafter\edef\csname #1#2\endcsname{\expandonce#3}}%
3330 }

```

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

```
3331 \newcommand*{\LWR@getexpararray}[2]{\csuse{#1#2}}
```

```
3332 \end{warpHTML}
```

### 34 HTML entities

**for HTML output:** 3333 \begin{warpHTML}

HTML entites and HTML Unicode entities:

```
3334 \let\LWR@origampersand\&
```

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

```

3335 \newcommand*\HTMLentity[1]{%
3336 % \LWR@traceinfo{HTMLentity \detokenize{#1}}%
3337 \begingroup%
3338 \LWR@FBcancel%
3339 \LWR@origampersand#1;%
3340 \endgroup
3341 % \LWR@traceinfo{HTMLentity done}%
3342 }
```

`\HTMLunicode`  $\{\langle hex\_unicode \rangle\}$

```

3343 \newcommand*\HTMLunicode[1]{\HTMLentity{\#x#1}}
```

`\&`

```

3344 \renewcommand*\&{\HTMLentity{amp}}
```

`\textless`

`\textgreater`

```

3345 \let\LWR@origtextless\textless
3346 \renewcommand*\textless{\HTMLentity{lt}}
3347
3348 \let\LWR@origtextgreater\textgreater
3349 \renewcommand*\textgreater{\HTMLentity{gt}}

3350 \end{warpHTML}
```

## 35 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:** 3351 `\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.

```

3352 \providecommand*\BaseJobname{\jobname}
```

`\HTMLFilename` The prefix for all generated HTML files other than the home page, defaulting to empty. See section 7.3.1.

```
3353 \providecommand*\HTMLFilename{}{}
```

`\HomeHTMLFilename` The filename of the home page, defaulting to the `\BaseJobname`. See section 7.3.1.

```
3354 \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.

```
3355 \newcommand*\SetHTMLFileNumber[1]{%
3356 \setcounter{LWR@htmlfilenumber}{#1}%
3357 }
```

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

Defaults to use section names in the filenames.

```
3358 \newbool{FileSectionNames}
3359 \booltrue{FileSectionNames}
```

```
3360 \end{warppall}
```

**for HTML output:** `3361 \begin{warppHTML}`

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

```
3362 \newcounter{LWR@htmlfilenumber}
3363 \setcounter{LWR@htmlfilenumber}{0}
```

`\LWR@htmlsectionfilename` `{\langle htmlfilenumber or name \rangle}`

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

```
3364 \newcommand*\LWR@htmlsectionfilename[1]{%
3365 \LWR@traceinfo{LWR@htmlsectionfilename A}%
```

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

```

3366 \LWR@traceinfo{about to assign temp}%
3367 \edef\LWR@tempone{#1}%
3368 \LWR@traceinfo{about to compare with ??}%
3369 \ifthenelse{\equal{\LWR@tempone}{??}}%
3370 {\LWR@traceinfo{found ??}}%
3371 {\LWR@traceinfo{not found ??}}%
3372 \LWR@traceinfo{about to compare with zero or empty}%
3373 \ifthenelse{%
3374   \equal{\LWR@tempone}{0}%
3375   \OR \equal{\LWR@tempone}{}%
3376   \OR \equal{\LWR@tempone}{??}%
3377 }%
3378 {%
3379   \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
3380   \HomeHTMLFilename.html%
3381 }%

```

For a  $\LaTeX$  section named “Index” or “index” without a prefix, create a filename with a leading underscore to avoid colliding with the HTML filename `index.html`:

```

3382 {%
3383   \LWR@traceinfo{LWR@htmlsectionfilename C \LWR@tempone}%
3384   \ifthenelse{%
3385     \equal{\HTMLFilename}{%} \AND
3386     \equal{\LWR@tempone}{Index} \OR
3387     \equal{\LWR@tempone}{index}%
3388   }%
3389   {%
3390     \LWR@traceinfo{prefixing the index name with an underscore.}%
3391     \_#1.html%
3392   }%

```

Otherwise, create a filename with the chosen prefix:

```

3393   {\HTMLFilename#1.html}%
3394 }%
3395 \LWR@traceinfo{LWR@htmlsectionfilename Z}%
3396 }

```

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

Prints the filename for the given label

```

3397 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
3398 \LWR@traceinfo{LWR@htmlrefsectionfilename A: !#1!}%
3399 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%

```

```

3400 \LWR@traceinfo{LWR@htmlrefsectionfilename B}%
3401 }

3402 \end{warpHTML}

```

## 36 Homepage link

**for HTML output:** 3403 \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.

```

3404 \newcommand*{\LinkHome}{%
3405 \LWR@subhyperrefclass{%
3406 \HomeHTMLFilename.html}%
3407 {Home}{linkhome}%
3408 }

```

`\LWR@topnavigation` Creates a link to the homepage at the top of the page for use when the window is too narrow for the sideroc.

```

3409 \newcommand*{\LWR@topnavigation}{
3410 \LWR@htmlclassline{nav}{topnavigation}{\LinkHome}
3411 }

```

`\LWR@botnavigation` Creates a link to the homepage at the bottom of the page for use when the window is too narrow for the sideroc.

```

3412 \newcommand*{\LWR@botnavigation}{
3413 \LWR@htmlclassline{nav}{botnavigation}{\LinkHome}
3414 }

3415 \end{warpHTML}

```

## 37 \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:** 3416 \begin{warpHTML}

`\PrintStack` Prints the closedepth stack.

```

3417 \newcommand*{\PrintStack}{
3418 \LWR@startpars
3419 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
3420 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
3421 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
3422 \LWR@closedephten{} \LWR@closedeptheleven{} \LWR@closedephtwelve{}
3423 }

3424 \end{warpHTML}

```

## 38 Closing stack levels

**for HTML output:** 3425 `\begin{warpHTML}`

Close one nested level:

```

3426 \newcommand*{\LWR@closeoneprevious}{%
3427
3428 \LWR@closeone{}
3429
3430 \popclose{}
3431 }

```

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

```

3432 \newcommand*{\LWR@closeprevious}[1]{

Close any pending paragraph:

3433 \LWR@stoppars

Close anything nested deeper than the desired depth:

3434 \whileof\not\(\LWR@closedepthone<#1\)){\LWR@closeoneprevious}
3435 }

3436 \end{warpHTML}

```

## 39 PDF pages and styles

**for HTML output:** 3437 `\begin{warpHTML}`

`\LWR@forcenewpage` New PDF page a before major environment.

This is used just before major environments, such as verse. Reduces the chance of an environment overflowing the HTML PDF output page.

```
3438 \newcommand{\LWR@forcenewpage}{%
3439 \LWR@stoppars\LWR@originewpage\LWR@startpars%
3440 }
```

`\pagestyle`, etc. are nullified for HTML output.

`\pagestyle` `{<style>}`

```
3441 \renewcommand*{\pagestyle}[1]{}
```

`\thispagestyle` `{<style>}`

```
3442 \renewcommand*{\thispagestyle}[1]{}
```

`\markboth` `{<left>}{<right>}`

```
3443 \renewcommand*{\markboth}[2]{}
```

`\markright` `{<right>}`

```
3444 \renewcommand*{\markright}[1]{}
```

`\raggedbottom`

```
3445 \renewcommand*{\raggedbottom}{}%
```

`\flushbottom`

```
3446 \renewcommand*{\flushbottom}{}%
```

`\sloppy`

```
3447 \renewcommand*{\sloppy}{}%
```

`\fussy`

```
3448 \renewcommand*{\fussy}{}%
```



```

\pagenumbering   {\langle commands\rangle}

3449 \renewcommand*{\pagenumbering}[1]{}

3450 \end{warpHTML}

```

## 40 HTML tags, spans, divs, elements

for HTML output: 3451 \begin{warpHTML}

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

```

3452 \newcommand*{\LWR@tagtitle}{h1}
3453 \newcommand*{\LWR@tagtitleend}{/h1}
3454 \newcommand*{\LWR@tagpart}{h2}
3455 \newcommand*{\LWR@tagpartend}{/h2}
3456 \newcommand*{\LWR@tagchapter}{h3}
3457 \newcommand*{\LWR@tagchapterend}{/h3}
3458 \newcommand*{\LWR@tagsection}{h4}
3459 \newcommand*{\LWR@tagsectionend}{/h4}
3460 \newcommand*{\LWR@tagsubsection}{h5}
3461 \newcommand*{\LWR@tagsubsectionend}{/h5}
3462 \newcommand*{\LWR@tagsubsubsection}{h6}
3463 \newcommand*{\LWR@tagsubsubsectionend}{/h6}
3464 \newcommand*{\LWR@tagparagraph}{span class="paragraph"{} }
3465 \newcommand*{\LWR@tagparagraphend}{/span}
3466 \newcommand*{\LWR@tagsubparagraph}{span class="subparagraph"{} }
3467 \newcommand*{\LWR@tagsubparagraphend}{/span}
3468
3469 \newcommand*{\LWR@tagregularparagraph}{p}

```

### 40.2 Babel-French

Adjust babel-french for HTML spaces. So far, this only works for pdf<sub>l</sub>at<sub>e</sub>x and xelat<sub>e</sub>x.

*(Based on original code by DANIEL FLIPO.)*

```

3470 \providecommand*{\LWR@FBcancel}{}
3471
3472 \AtBeginDocument{%
3473 \@ifundefined{frenchbsetup}%
3474 {}%

```

```

3475 {%
3476   \frenchbsetup{FrenchFootnotes=false}%

3477 %
3478   \LetLtxMacro\LWR@FBcancel\NoAutoSpacing%
3479   \renewcommand*{\FBcolonspace}{%
3480     \begingroup%
3481     \LWR@FBcancel%
3482     \LWR@origampersand{} \,
3483     \endgroup%
3484   }%
3485   \renewcommand*{\FBthinspace}{%
3486     \begingroup%
3487     \LWR@FBcancel%
3488     \LWR@origampersand\#x202f;% \,
3489     \endgroup%
3490   }%
3491   \renewcommand*{\FBguillspace}{%
3492     \begingroup%
3493     \LWR@FBcancel%
3494     \LWR@origampersand{} \, for \og xyz \fg{}
3495     \endgroup%
3496   }%
3497   \DeclareDocumentCommand{\FBmedkern}{-}{%
3498     \begingroup%
3499     \LWR@FBcancel%
3500     \LWR@origampersand\#x202f;% \,
3501     \endgroup%
3502   }%
3503   \DeclareDocumentCommand{\FBthickkern}{-}{%
3504     \begingroup%
3505     \LWR@FBcancel%
3506     \LWR@origampersand{} \, ~
3507     \endgroup%
3508   }%
3509   \renewcommand*{~}{\HTMLentity{nbsp}}% was overwritten by babel-french
3510   \ifFBunicode%
3511   \else%
3512     \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}%
3513     \DeclareTextCommandDefault{\FBtextellipsis}{\textellipsis\xspace}%
3514   \fi%
3515 }%
3516 }

```

### 40.3 HTML tags

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

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

```

3517 \newcommand*{\LWR@htmltagc}[1]{%
3518 {%
3519 % \LWR@traceinfo{\LWR@htmltagc \detokenize{#1}}%
3520 \begingroup%
3521 \LWR@FBcancel%
3522 \ifmmode\else\protect\LWR@origttfamily\fi%
3523 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
3524 \endgroup%
3525 }%
3526 }

```

Env `LWR@nestspan` Disable minipage, `\parbox`, and HTML `<div>`s inside a `<span>`.

⚠ `\begin{LWR@nestspan}` must follow the opening `<span>` tag to allow a paragraph to start if the span is at the beginning of a new paragraph.

⚠ `\end{LWR@nestspan}` must follow the `</span>` or a `<p>` may appear inside the span.

```

3527 \newcommand*{\LWR@nestspanitem}{%
3528 \ifnewlist\else{\LWR@htmltagc{br /}}\fi%
3529 \LWR@origitem%
3530 }
3531
3532 \newenvironment*{LWR@nestspan}
3533 {%
3534 \LWR@traceinfo{LWR@nestspan}%
3535 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
3536 {%
3537   \LWR@traceinfo{LWR@nestspan: inside a lateximage}%
3538 }%
3539 {% not in a lateximage
3540   \LWR@traceinfo{LWR@nestspan: NOT inside a lateximage}%
3541   \addtocounter{LWR@spandepth}{1}%
3542   \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{-}{-}%
3543   \RenewDocumentEnvironment{BlockClass}{o m}{-}{-}%
3544   \renewcommand{\BlockClassSingle}[2]{##2}%
3545   \renewcommand{\LWR@forcenewpage}{-}%
3546   \renewcommand{\LWR@itemizestart}{%
3547     \let\item\LWR@nestspanitem%
3548   }%
3549   \renewcommand{\LWR@itemizeend}{-}%
3550   \renewcommand{\LWR@enumeratestart}{%
3551     \let\item\LWR@nestspanitem%
3552   }%
3553   \renewcommand{\LWR@enumerateend}{-}%
3554   \renewcommand{\LWR@descriptionstart}{%
3555     \let\item\LWR@nestspanitem%

```

```

3556     }%
3557     \renewcommand{\LWR@descriptionend}{}%
3558 }% not in a lateximage
3559 }% starting env
3560 {% ending env
3561 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{%
3562 }{%
3563 {\addtocounter{LWR@spandepth}{-1}}%
3564 \LWR@traceinfo{LWR@nestspan: done}%
3565 }
3566
3567 \AfterEndEnvironment{LWR@nestspan}{\global\let\par\LWR@closeparagraph}

```

`\LWR@htmlspan`    `{\langle tag \rangle}{\langle text \rangle}`



`\LWR@spandepth` is used to ensure that paragraph tags are not generated inside a span. The exact sequence of when to add and subtract the counter is important to correctly handle the paragraph tags before and after the span.

```

3568 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
3569 \LWR@ensuredoingapar%
3570 \LWR@htmltagc{#1}%
3571 \begin{LWR@nestspan}%
3572 #2%
3573 \LWR@htmltagc{/#1}%
3574 \end{LWR@nestspan}%
3575 }

```

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

```

3576 \NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{%
3577 \LWR@traceinfo{LWR@htmlspanclass #2}%
3578 \LWR@ensuredoingapar%
3579 \LWR@subhtmlclass{span}[#1]{#2}%
3580 \begin{LWR@nestspan}%
3581 #3%
3582 \LWR@htmltagc{/span}%
3583 \LWR@traceinfo{LWR@htmlspanclass done}%
3584 \end{LWR@nestspan}%
3585 }

```

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

Print an HTML tag: `<tag>`

```

3586 \newcommand*{\LWR@htmltagb}[1]{%
3587 \LWR@htmltagc{#1}%

```

```

3588 \endgroup%
3589 }
3590
3591 \newcommand*{\LWR@htmltag}{%
3592 \begingroup\catcode'\_ =12
3593 \LWR@htmltagb%
3594 }

```

## 40.4 Block tags and comments

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

```

\LWR@htmlopencomment
\LWR@htmlclosecomment
3595 \newcommand*{\LWR@htmlopencomment}{%
3596 {%
3597 % \LWR@traceinfo{\LWR@htmlopencomment}%
3598 \begingroup%
3599 \LWR@FBcancel%
3600 \ifmmode\else\protect\LWR@origttfamily\fi%
3601 \LWR@origtextless{}!{-}{-}%
3602 \endgroup%
3603 }%
3604 }
3605
3606 \newcommand*{\LWR@htmlclosecomment}{%
3607 {%
3608 % \LWR@traceinfo{\LWR@htmlclosecomment}%
3609 \begingroup%
3610 \LWR@FBcancel%
3611 \ifmmode\else\protect\LWR@origttfamily\fi%
3612 {-}{-}\LWR@origtextgreater{}%
3613 \endgroup%
3614 }%
3615 }

```

```

\LWR@htmlcomment  {\langle comment \rangle}

3616 \newcommand{\LWR@htmlcomment}[1]{%
3617 \LWR@htmlopencomment}%
3618 {%
3619 \LWR@origttfamily% break ligatures
3620 #1%
3621 }%
3622 \LWR@htmlclosecomment{}}

```

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

```

3623 \newcommand{\LWR@htmlblockcommentb}[1]
3624 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
3625
3626 \newcommand{\LWR@htmlblockcomment}
3627 {%
3628 \begingroup\catcode'\_ =12%
3629 \LWR@htmlblockcommentb%
3630 }

```

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

```

3631 \newcommand*\LWR@htmlblocktag[1]{%
3632 \LWR@stoppars%
3633 \LWR@htmltag{#1}%
3634 \LWR@startpars%
3635 }

```

## 40.5 Div class and element class

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

Factored and reused in several places.

The trailing spaces allow more places for a line break.

```

3636 \NewDocumentCommand{\LWR@subhtmllementclass}{m o m}{%
3637 \LWR@traceinfo{\LWR@subhtmllementclass #1 #3}%
3638 \IfValueTF{#2}%
3639 {% option
3640   \ifthenelse{\equal{#2}{}}{%
3641     {\LWR@htmltag{#1 class="#3"}}% empty option
3642     {\LWR@htmltag{#1 class="#3" style="#2"}}% non-empty option
3643 }% option
3644 {\LWR@htmltag{#1 class="#3"}}% no option
3645 \LWR@traceinfo{\LWR@subhtmllementclass done}%
3646 }

```

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

```

3647 \NewDocumentCommand{\LWR@htmllementclass}{m o m}{%
3648 \LWR@stoppars%
3649 \LWR@subhtmllementclass{#1}[#2]{#3}%
3650 \LWR@startpars%
3651 }

```

```

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

3652 \newcommand*{\LWR@htmlelementclassend}[2]{%
3653 \LWR@stoppars%
3654 \LWR@htmltag{/#1}%
3655 \ifbool{HTMLDebugComments}{%
3656     \LWR@htmlcomment{End of #1 ‘‘#2’’}%
3657 }{}%
3658 \LWR@startpars%
3659 }

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

3660 \NewDocumentCommand{\LWR@htmldivclass}{o m}{%
3661 \LWR@htmlelementclass{div}[#1]{#2}%
3662 }

\LWR@htmldivclassend  {\langle class \rangle}

3663 \newcommand*{\LWR@htmldivclassend}[1]{%
3664 \LWR@htmlelementclassend{div}{#1}%
3665 }

```

## 40.6 Single-line elements

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

```

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

3666 \NewDocumentCommand{\LWR@htmlelementclassline}{m o m +m}{%
3667 \LWR@stoppars
3668 \LWR@subhtmlelementclass{#1}[#2]{#3}%
3669 #4%
3670 \LWR@htmltag{/#1}
3671 \LWR@startpars
3672 }

```

## 40.7 HTML5 semantic elements

```

\LWR@htmlmlelement  {\langle element \rangle}

```

```

3673 \newcommand*{\LWR@htmlelement}[1]{%
3674 \LWR@htmlblocktag{#1}
3675 }

```

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

```

3676 \newcommand*{\LWR@htmlelementend}[1]{%
3677 \LWR@stoppars
3678 \LWR@htmltag{/#1}
3679 \LWR@startpars
3680 }
3681
3682 \end{warpHTML}

```

## 40.8 High-level block and inline classes

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

Nullified versions are provided for print mode.

For other direct-formatting commands, see section [75](#).

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

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

**for HTML output:**

```

3683 \begin{warpHTML}
3684 \NewDocumentEnvironment{BlockClass}{o m}%
3685 {
3686 \LWR@htmldivclass[#1]{#2}
3687 }
3688 {
3689 \LWR@htmldivclassend{#2}
3690 }
3691 \end{warpHTML}

```

**for PRINT output:**

```

3692 \begin{warpprint}
3693 \NewDocumentEnvironment{BlockClass}{o m}{\}{}%
3694 \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:**

```

3695 \begin{warpHTML}
3696 \newcommand*{\BlockClassSingle}[2]{%
3697 \LWR@htmlelementclassline{div}{#1}{#2}%

```



```
3698 }
3699 \end{warpHTML}
```

**for PRINT output:** 3700 \begin{warpprint}  
3701 \newcommand{\BlockClassSingle}[2]{#2}  
3702 \end{warpprint}

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

**for HTML output:** 3703 \begin{warpHTML}  
3704 \NewDocumentCommand{\InlineClass}{o m +m}{%  
3705 \LWR@htmlspanclass[#1]{#2}{#3}%  
3706 }  
3707 \end{warpHTML}

**for PRINT output:** 3708 \begin{warpprint}  
3709 \NewDocumentCommand{\InlineClass}{o m +m}{#3}%  
3710 \end{warpprint}

Env `LWR@BlockClassWP` [`<WPstyle>`] [`<HTMLstyle>`] [`<class>`] Low-level interface for `<div>` classes with an automatic float ID. These are often used when `\ifbool{FormatWP}`.

**for HTML output:** 3711 \begin{warpHTML}  
3712 \NewDocumentEnvironment{LWR@BlockClassWP}{m m m}%  
3713 {%  
3714 \LWR@stoppars%  
3715 \ifbool{FormatWP}%  
3716 {%  
3717 \LWR@maybeinthisfloat%  
3718 \LWR@htmltag{%  
3719 div class="#3" %  
3720 id="autofloat-\arabic{LWR@thisfloat}"%  
3721 \ifblank{#1}{#{ style="#1"%  
3722 }%  
3723 }% FormatWP  
3724 {% not FormatWP  
3725 \LWR@htmltag{%  
3726 div class="#3"%  
3727 \ifblank{#2}{#{ style="#2"%  
3728 }%  
3729 }% not FormatWP  
3730 \LWR@startpars%  
3731 }  
3732 {\LWR@htmldivclassend{#3}}  
3733 \end{warpHTML}

**for PRINT output:** 3734 \begin{warpprint}  
3735 \NewDocumentEnvironment{LWR@BlockClassWP}{m m m}{-}{-}%  
3736 \end{warpprint}

## 40.9 Closing HTML tags

for HTML output: 3737 \begin{warpHTML}

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

```

3738 \newcommand*{\LWR@printclosepart}
3739     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
3740 \newcommand*{\LWR@printclosechapter}
3741     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
3742 \newcommand*{\LWR@printclosesection}
3743     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
3744 \newcommand*{\LWR@printclosesubsection}
3745     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
3746 \newcommand*{\LWR@printclosesubsubsection}
3747     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
3748 \newcommand*{\LWR@printcloseparagraph}
3749     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
3750 \newcommand*{\LWR@printclosesubparagraph}
3751     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}

```

Lists require closing HTML tags:

```

3752 \newcommand*{\LWR@printcloselistitem}
3753     {\LWR@htmltag{/li}}
3754 \newcommand*{\LWR@printclosedescitem}
3755     {\LWR@htmltag{/dd}}
3756 \newcommand*{\LWR@printcloseitemize}
3757     {\LWR@htmltag{/ul}}
3758 \newcommand*{\LWR@printcloseenumerate}
3759     {\LWR@htmltag{/ol}}
3760 \newcommand*{\LWR@printclosedescription}
3761     {\LWR@htmltag{/dl}}

3762 \end{warpHTML}

```

## 41 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 42 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:** 3763 `\begin{warpHTML}`

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

```
3764 \newcounter{LWR@spandepth}
3765 \setcounter{LWR@spandepth}{0}
```

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

```
3766 \newbool{LWR@doingstartpars}
3767 \boolfalse{LWR@doingstartpars}
```

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

```
3768 \newbool{LWR@doingapar}
3769 \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:

```
3770 \newcommand*{\LWR@ensuredoingapar}{%
3771 \ifbool{LWR@doingstartpars}%
3772 {\global\booltrue{LWR@doingapar}}%
3773 {}%
3774 }
```

`\LWR@openparagraph`

```
3775 \newcommand*{\LWR@openparagraph}
3776 {%
```

See if paragraph handling is enabled:

```
3777 \ifbool{LWR@doingstartpars}%
3778 {% handling pars
```

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

```
3779 \ifthenelse{%
3780 \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3781 \cnttest{\value{LWR@spandepth}}{>}{0}%
3782 }% 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>.

```
3783    {}% no nested par tags
```

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

```
3784    {% yes nest par tags
3785        \LWR@htmltagc{\LWR@tagregularparagraph}%
```

Manually indent item list labels to avoid left margin intrusion:

See if are nested inside an item list:

```
3786        \ifnumcomp{\@listdepth}{>}{0}%
3787        {%
```

If so, leave some horizontal room in the PDF output for list labels:

```
3788        \LWR@origspace{1in}%
3789        }%
3790        {}%
```

Now have started a paragraph.

```
3791        \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.)

```
3792        \let\par\LWR@closeparagraph%
3793    }% end of yes nest par tags
3794}% end of handling pars
3795{}% not handling pars
3796 }
```

\LWR@closeparagraph

```
3797 \newcommand*{\LWR@closeparagraph}
3798 {%
```

See if paragraph handling is enabled:

```
3799 \ifbool{\LWR@doingapar}%
```

If currently in paragraph mode:

```
3800 {% handling pars
```

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

```
3801     \ifthenelse{%
3802         \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3803         \cnttest{\value{LWR@spandepth}}{>}{0}%
3804     }%
```

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

```
3805     {% no nested par tags
3806         \ifthenelse{%
3807             \cnttest{\value{LWR@spandepth}}{>}{0}%
3808             \AND%
3809             \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3810         }%
3811         {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
3812         {}}%
3813     {% no nested par tags
```

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

```
3814     {% yes nest par tags
```

Print a closing tag and some extra vertical space:

```
3815         \unskip%
3816         \LWR@htmltagc{/LWR@tagregularparagraph}%
3817         \LWR@orignewline%
```

No longer doing a paragraph:

```
3818         \global\boolfalse{LWR@doingapar}%
3819 % Disable the special \env{minipage} & \cs{hspace} interaction
3820 % until a new minipage is found:
3821 %     \begin{macrocode}
3822         \global\boolfalse{LWR@minipagethispar}%
3823     }% end of yes nest par tags
3824}% end of handling pars
```

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

```
3825 {% not handling pars
3826     \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
```

```

3827     {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}%
3828     {}}%
3829 }% not handling pars

```

Finish with regular paragraph processing

```

3830 \LWR@origpar%
3831 }

```

```

3832 \end{warpHTML}

```

## 42 Paragraph start/stop handling

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

Section 41 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:** 3833 `\begin{warpHTML}`

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

```

3834 \newcommand*{\LWR@startpars}%
3835 {%

```

Ignore if inside a span:

```

3836 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
3837 {}%
3838 {%

```

See if currently handling HTML paragraphs:

```

3839     \ifbool{LWR@doingstartpars}%

```

If already in paragraph mode, do nothing.

```

3840     {}%

```

If not currently in paragraph mode:

```

3841     {%

```

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

```
3842      \PushPreHook{par}{\LWR@openparagraph}%
```

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

```
3843      \let\par\LWR@closeparagraph
```

```
3844
```

```
3845      }% an intentionally blank line
```

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

```
3846      \global\setbool{LWR@doingstartpars}{true}%
```

No <par> tag yet to undo:

```
3847      \global\boolfalse{LWR@doingapar}%
```

```
3848 }% nestspan
```

```
3849 }
```

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

```
3850 \newcommand*{\LWR@stoppars}%
```

```
3851 {%
```

Ignore if inside a span:

```
3852 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
```

```
3853 {}%
```

```
3854 {%
```

See if currently handling HTML paragraphs:

```
3855      \ifbool{LWR@doingapar}%
```

if currently in an HTML paragraph:

```
3856      {%
```

Print a closing tag:

```
3857      \unskip%
```

```
3858      \LWR@htmltagc{\LWR@tagregularparagraph}%
```

```
3859      \LWR@orignewline%
```

No longer have an open HTML paragraph:

```
3860      \global\boolfalse{LWR@doingapar}%
```

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

```
3861      \global\boolfalse{LWR@minipagethispar}
3862
3863    }% an intentionally blank line
```

If was not in an HTML paragraph:

```
3864    {}%
```

See if currently allowing HTML paragraphs:

```
3865    \ifbool{LWR@doingstartpars}%
```

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

```
3866    {\ClearPreHook{par}}%
```

Else: do nothing

```
3867    {}%
```

no longer in paragraph mode

```
3868    \global\setbool{LWR@doingstartpars}{false}%
```

no <p> tag to undo:

```
3869    \global\boolfalse{LWR@doingapar}%
3870 }% nestspan
3871 }
```

```
3872 \end{warpHTML}
```

## 43 Page headers and footers

**for HTML & PRINT:** 3873 \begin{warpall}

In the following, catcode is manually changed back and forth without groups, since new macros are being defined which must not be contained within the groups.



```

3874 \newcommand{\LWR@firstpagetop}{ } % for the home page alone
3875 \newcommand{\LWR@pagetop}{ } % for all other pages
3876 \newcommand{\LWR@pagebottom}{ }
3877
3878 \newcommand{\LWR@setfirstpagetopb}[1]{%
3879 \renewcommand{\LWR@firstpagetop}{#1}
3880 \catcode'\_ =8
3881 }

```

\HTMLFirstPageTop {*(text and logos)*}

```

3882 \newcommand{\HTMLFirstPageTop}{%
3883 \catcode'\_ =12
3884 \LWR@setfirstpagetopb
3885 }

3886 \newcommand{\LWR@setpagetopb}[1]{%
3887 \renewcommand{\LWR@pagetop}{#1}
3888 \catcode'\_ =8
3889 }

```

\HTMLPageTop {*(text and logos)*}

```

3890 \newcommand{\HTMLPageTop}{%
3891 \catcode'\_ =12
3892 \LWR@setpagetopb
3893 }

3894 \newcommand{\LWR@setpagebottomb}[1]{%
3895 \renewcommand{\LWR@pagebottom}{#1}
3896 \catcode'\_ =8
3897 }

```

\HTMLPageBottom {*(text and logos)*}

```

3898 \newcommand{\HTMLPageBottom}{%
3899 \catcode'\_ =12
3900 \LWR@setpagebottomb
3901 }

3902 \end{warpall}

```

## 44 CSS

for HTML output: 3903 \begin{warpHTML}

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

```
3904 \newcommand*{\LWR@currentcss}{lwarp.css}
```

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

```
3905 \newcommand*{\LWR@newcssb}[1]{%
3906 \renewcommand*{\LWR@currentcss}{#1}
3907 \catcode'\_ =8
3908 }
3909
3910 \newcommand*{\CSSFilename}{
3911 \catcode'\_ =12
3912 \LWR@newcssb
3913 }
3914 \end{warpHTML}
```

**for PRINT output:** `3915 \begin{warpprint}`  
`3916 \newcommand*{\CSSFilename}[1]{}`  
`3917 \end{warpprint}`

## 45 HTML meta description and author

**for HTML & PRINT:** `3918 \begin{warpall}`

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

```
3919 \providecommand{\theauthor}{}
3920 \newcommand{\theHTMLAuthor}{\theauthor}
3921
3922 \newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}

3923 \end{warpall}
```

**for HTML & PRINT:** `3924 \begin{warpall}`

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



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

`\HTMLDescription` `{\langle New HTML meta description.\rangle}` Assigns the HTML file's description meta tag.


```

3925 \newcommand{\LWR@currentHTMLDescription}{}
3926
3927 \newcommand{\HTMLDescription}[1]{%
3928 \renewcommand{\LWR@currentHTMLDescription}{#1}
3929 }
3930
3931 \end{warpall}

```

## 46 Footnotes

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

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

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.

### 46.1 Regular page footnotes

In HTML documents, footnotes are placed at the bottom of the web page using the  $\TeX$  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 46.4 for the implementation.

### 46.2 Minipage footnotes

See section 46.5 for how minipage footnotes are gathered. See section 74.3 for how minipage footnotes are placed into the document.

### 46.3 Titlepage thanks

See section 53.6 for titlepage footnotes.

## 46.4 Regular page footnote implementation

**for HTML & PRINT:** 3932 `\begin{warpall}`

**Ctrl** `FootnoteDepth` Determines how deeply to place footnotes in the HTML files, similar to `tocdepth`.  
**Default:** 5 The default of 5 places footnotes before each `\subparagraph` or higher. See table 7 for a table of  $\TeX$  section headings.

```
3933 \newcounter{FootnoteDepth}
3934 \setcounter{FootnoteDepth}{5}

3935 \end{warpall}
```

**for HTML output:** 3936 `\begin{warpHTML}`

Patch  $\TeX$  footnotes to use a new `\box` for lwarp footnotes.

```
3937 \newbox\LWR@footnotes
```

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

`\@makefntext` `{\langle text \rangle}`

```
3938 \long\def\@makefntext#1{\textsuperscript{\@thefnmark} #1}
```

`\@makefnmark`

```
3939 \def\@makefnmark{\hbox{\textsuperscript{\@thefnmark}}}
```

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

In a `lateximage` during HTML output, the `lateximage` is placed inside a print-mode `minipage`, but the footnotes are broken out by:

```
\def\@mpfn{footnote}
\def\thempfn{\thefootnote}
\let\@footnotetext\LWR@footnotetext
```

`\LWR@footnotetext` `{\langle text \rangle}`

```
3940 \long\def\LWR@footnotetext#1{%
3941 \global\setbox\LWR@footnotes=\vbox{%
```

Add to any current footnotes:

```
3942    \unvbox\LWR@footnotes%
```

Remember the footnote number for \ref:

```
3943    \protected@edef\@currentlabel{%
3944        \csname p@footnote\endcsname\@thefnmark%
3945    }% @currentlabel
```

Open a group:

```
3946    \color@begingroup%
```

Use HTML superscripts even inside a lateximage:

```
3947    \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

Use paragraph tags if in a tabular data cell or a lateximage:

```
3948    \ifthenelse{%
3949        \boolean{LWR@doingstartpars} \AND%
3950        \cnttest{\value{LWR@lateximagedepth}}{=}{0}}%
3951    }%
3952    {%
3953        {\LWR@htmltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
3954    \@makefntext{#1}%
```

Closing paragraph tag:

```
3955    \ifthenelse{%
3956        \boolean{LWR@doingstartpars} \AND%
3957        \cnttest{\value{LWR@lateximagedepth}}{=}{0}}%
3958    }%
3959    {\par}%
3960    {%
3961        \LWR@htmltagc{/LWR@tagregularparagraph}%
3962        \LWR@orignewline%
3963    }%
```

Close the group:

```
3964    \color@endgroup%
3965    }% vbox
3966    }%
```

```
\@footnotetext {<text>}
```

```
3967 \let\@footnotetext\LWR@footnotetext
```

## 46.5 Minipage footnote implementation

```
\@mpfootnotetext {<text>}
```

```
3968 \long\def\@mpfootnotetext#1{%
3969 \global\setbox\@mpfootins\vbox{%
3970   \unvbox\@mpfootins
3971   \reset@font\footnotesize
3972   \hsize\columnwidth
3973   \@parboxrestore
3974   \protected@edef\@currentlabel
3975   {\csname p@mpfootnote\endcsname\@thefnmark}%
3976   \color@begingroup
```

Use paragraph tags if in a tabular data cell or a lateximage:

```
3977   \ifthenelse{%
3978     \boolean{LWR@doingstartpars} \AND%
3979     \cnttest{\value{LWR@lateximagedepth}}{=}{0}}%
3980   {%
3981     }%
3982     {\LWR@htmltagc{\LWR@tagregularparagraph}}%

3983   \@makefntext{%
3984     \ignorespaces#1%
3985   }%
```

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

```
3986   \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
3987   {%
3988     {%
3989       \LWR@htmltagc{/\LWR@tagregularparagraph}%
3990       \LWR@orignewline%
3991     }%
3992     \color@endgroup%
3993   }% vbox
3994 }
```

## 46.6 Printing pending footnotes

`\LWR@printpendingfootnotes` Enclose the footnotes in a class, print, then clear.

```

3995 \newcommand*\LWR@printpendingfootnotes}{%
3996 \ifvoid\LWR@footnotes\else
3997   \LWR@forcenewpage
3998   \begin{BlockClass}{footnotes}
3999   \LWR@origmedskip
4000   \unvbox\LWR@footnotes
4001   \setbox\LWR@footnotes=\vbox{}
4002   \end{BlockClass}
4003 \fi
4004 }
```

`\LWR@maybeprintpendingfootnotes` `{<depth>}` Used to print footnotes before sections only if formatting for an EPUB or word processor:

```

4005 \newcommand*\LWR@maybeprintpendingfootnotes}[1]{%
4006 \ifboolexpr{
4007   not test{\ifnumcomp{#1}{>}{\value{FootnoteDepth}}} or
4008   bool{FormatEPUB} or
4009   bool{FormatWP}
4010 }%
4011 {\LWR@printpendingfootnotes}%
4012 {}%
4013 }

4014 \end{warpHTML}
```

## 47 Marginpars

**for HTML output:** `4015 \begin{warpHTML}`

`\marginpar` [`<left>`] [`<right>`]

```

4016 \renewcommand{\marginpar}[2][ ]{%
4017 \ifbool{FormatWP}%
4018 {%
4019 \begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{\marginblock}
4020 #2
4021 \end{LWR@BlockClassWP}
4022 }%
4023 {\LWR@htmlspanclass{marginpar}{#2}}%
4024 }
```

```
4025 \end{warpHTML}
```

## 48 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:** 4026 \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.

```
4027 \newcounter{FileDepth}
4028 \setcounter{FileDepth}{-5}
```

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

```
4029 \newbool{CombineHigherDepths}
4030 \booltrue{CombineHigherDepths}
```

```
4031 \end{warppall}
```

**for HTML output:** 4032 \begin{warpHTML}

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

```
4033 \newcommand*{\LWR@thisfilename}{}
```

**\LWR@thisnewfilename** The filename being sanitized.

```
4034 \newcommand*{\LWR@thisnewfilename}{}
```

**\LWR@filenamenoblanks**  $\{\langle filename \rangle\}$

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`  $\TeX$  commands which appear in section names and TOC captions.

```
4035 \newcommand*{\LWR@filenamenoblanks}[1]{%
4036 \begingroup
```

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

```
4037 \LWR@nullfonts
4038 \renewcommand*{\LWR@htmltagc}[1]{%}
```

Replaces common symbols and short words with hyphens:

```
4039 \edef\LWR@thisnewfilename{#1}
4040 \fullexpandarg
```

Convert spaces into hyphens:

```
4041 \StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctuation into hyphens:

```
4042 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
4043 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
4044 \StrSubstitute{\LWR@thisnewfilename}{%
4045     {\LWR@origampersand}{-}[\LWR@thisnewfilename]
4046 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
4047 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
4048 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
4049 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
4050 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
4051 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
4052 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
4053 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
4054 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
4055 \StrSubstitute{\LWR@thisnewfilename}{%
4056     {\textless}{-}[\LWR@thisnewfilename]
4057 \StrSubstitute{\LWR@thisnewfilename}{%
4058     {\textgreater}{-}[\LWR@thisnewfilename]
4059 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
4060 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
4061 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
4062 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
4063 \StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
4064 \StrSubstitute{\LWR@thisnewfilename}{%
4065     {\textbackslash}{-}[\LWR@thisnewfilename]
4066 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
4067 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
```

```

4068 %      "~{}" for babel
4069 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
4070 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
4071 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]

```

Convert short words:

```

4072 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
4073 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
4074 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
4075 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
4076 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
4077 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
4078 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
4079 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
4080 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
4081 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
4082 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
4083 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
4084 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
4085 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
4086 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
4087 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
4088 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
4089 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

4090 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
4091 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
4092 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
4093 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
4094 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
4095 %      emdash
4096 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
4097 %      endash
4098 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
4099 \endgroup
4100 }

```

`\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.

```

4101 \newcommand*{\LWR@newhtmlfile}[1]{

```

At the bottom of the ending file:

```
4102 \LWR@htmlelementclassend{section}{textbody}
4103
4104 \LWR@printpendingfootnotes
4105
```

No footer between files if EPUB:

```
4106 \ifbool{FormatEPUB}
4107 {}
4108 {
4109     \LWR@htmlelement{footer}
4110
4111     \LWR@pagebottom
4112
4113     \LWR@htmlelementend{footer}
4114 }
```

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

```
4115 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4116 {}
4117 {\ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}}
```

End of this HTML file:

```
4118 \LWR@stoppars
4119 \LWR@htmltag{/body}\LWR@orignewline
4120 \LWR@htmltag{/html}\LWR@orignewline
4121 \LWR@orignewpage
4122
4123 \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.

```
4124 \ifbool{FileSectionNames}%
4125 {\LWR@filenamenoblanks{#1}}
4126 {\renewcommand*{\LWR@thisfilename}{\arabic{LWR@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.

```
4127 \LWR@htmlblockcomment{%
4128 |Start file|%
4129 \LWR@htmlsectionfilename{\LWR@thisfilename}|%
4130 }
```

At the top of the starting file:

```
4131 \LWR@stoppars
4132
4133 \LWR@filestart{ -- #1}% there is an Emdash in front of the #1
4134
```

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

```
4135 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4136 {}
4137 {\LWR@topnavigation}
4138
```

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

```
4139 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4140 {}
4141 {
4142   \LWR@htmlelement{header}
4143
4144   \LWR@pagetop
4145
4146   \LWR@htmlelementend{header}
4147 }
4148
```

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

```
4149 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4150 {}
4151 {\ifcvoid{thetitle}{}\LWR@printthetitle}}
4152
```

No sidetoc if formatting for an EPUB or word processor:

```
4153 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4154 {}
4155 {\LWR@sidetoc}
4156
```

Start of the <textbody>:

```
4157 \LWR@htmlclass{section}{\textbody}
4158
```

Keep paragraph tags disabled for now:

```
4159 \LWR@stoppars
4160
```

Track the page numbers:

```
4161 \setcounter{LWR@latestautopage}{\value{page}}
4162 }

4163 \end{warpHTML}
```

## 49 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing  $\TeX$  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  $\text{pdf\LaTeX}$  to generate characters and  $\text{pdf\LaTeX}$  to read them. If extra symbols appear in the text, it may be that  $\text{pdf\LaTeX}$  is actually producing a symbol over or under a character, resulting in  $\text{pdf\LaTeX}$  picking up the accent symbol separately.



$\text{Xe\LaTeX}$  and  $\text{Lua\LaTeX}$  directly support accented section and file names.

**for HTML output:** 4164 \begin{warpHTML}

### 49.1 User-level starred section commands

**\ForceHTMLPage** For HTML output, forces the next section to be on its own HTML page, if `FileDepth` allows, even if starred. For use with `\printindex` and others which generate a starred section which should be on its own HTML page. Also see `\ForceHTMLTOC`.

For print output, no effect.

```
4165 \newbool{LWR@forcinghtmlpage}
4166 \boolfalse{LWR@forcinghtmlpage}
4167
4168 \newcommand*{\ForceHTMLPage}{%
```

```
4169 \global\booltrue{LWR@forcinghtmlpage}%
4170 }
```

**\ForceHTMLTOC** For HTML output, forces the next section to have a TOC entry, even if starred. For use with `\printindex` and others which generate a starred section which should be in the TOC so that it may be accessed via HTML. Not necessary if used with `tocbibind`. Also see `\ForceHTMLPage`.

For print output, no effect.

```
4171 \newbool{LWR@forcinghtmltoc}
4172 \boolfalse{LWR@forcinghtmltoc}
4173
4174 \newcommand*{\ForceHTMLTOC}{%
4175 \global\booltrue{LWR@forcinghtmltoc}%
4176 }
```

```
4177 \end{warpHTML}
```


**for PRINT output:**

```
4178 \begin{warpprint}
4179 \newcommand*{\ForceHTMLPage}{}
4180 \newcommand*{\ForceHTMLTOC}{}
4181 \end{warpprint}
```

**for HTML output:**

```
4182 \begin{warpHTML}
```

## 49.2 Book class commands

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

```
4183 \newbool{LWR@mainmatter}
4184 \DeclareDocumentCommand{\mainmatter}{}{}%
4185 \booltrue{LWR@mainmatter}%
4186 }
```

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

```
4187 \DeclareDocumentCommand{\frontmatter}{}{}%
4188 \boolfalse{LWR@mainmatter}%
4189 }
```

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

```

4190 \DeclareDocumentCommand{\backmatter}{-}{-}{%
4191 \boolfalse{LWR@mainmatter}
4192 }

```

### 49.3 Sectioning support macros

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

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

```

4193 \newcommand*{\LWR@sectionnumber}[1]{%
4194 \InlineClass{sectionnumber}{#1}%
4195 }

```

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

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


Create an autosection tag.

```

4196 \newcommand*{\LWR@createautosec}[1]{%
4197 \LWR@htmltag{#1 id="autosec-\thepage"}{}}%
4198 }

```

`\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.


```

4199 \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.

```

4200 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%

```

Close any stacked sections up to this new one.

```

4201 \LWR@closeprevious{#1}%

```

Push a new section depth:

```
4202 \LWR@pushoneclose{#1}{#2}%
4203 }
```

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.

```
4204 \newcounter{LWR@prevFileDepth}
4205 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

\@secCNTformat {<sectiontype>}

```
4206 \def\@secCNTformat#1{\csname the#1\endcsname\protect\quad}
```

\simplechapterdelim Used by tocbibind and anonchap.

```
4207 \newcommand*{\simplechapterdelim}{}%
```

\@chapCNTformat {<sectiontype>}

\let to \@secCNTformat by default, but may be redefined by \simplechapter and \restorechapter from tocbibind or anonchap.

```
4208 \let\@chapCNTformat\@secCNTformat
```

Ctrl LWR@currentautosec Records the page number when the section was created. If a math expression is included in the section name, and SVG math is used, the corresponding lateximage will cause the page number to change by the time the following autosec label is created.

```
4209 \newcounter{LWR@currentautosec}
```

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

The common actions for the high-level sectioning commands.

```
4210 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
4211 \LWR@traceinfo{LWR@section |#2| |#3|}%
4212 \LWR@traceinfo{LWR@section: not an empty section}%
4213 \LWR@stoppars%
```



Cancel special minipage horizontal space interaction:

```
4214 \global\boolfalse{LWR@minipagethispar}%
```

Start a new HTML file unless starred, and if is a shallow sectioning depth.

Exception: Also start a new HTML file for \part\*, for appendix.

Generate a new  $\LaTeX$  page so that toc and index page number points to the section:

```
4215 \LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
4216 \IfBooleanTF{#1}{\LWR@traceinfo{LWR@section: starred}}{}%
4217 \ifbool{LWR@forcinghtmlpage}{\LWR@traceinfo{LWR@section: forcinghtmlpage}}{}%
4218 \ifthenelse{%
4219   \(%
4220     \(\NOT\equal{#1}{\BooleanTrue})\)\OR%
4221     \(\cnttest{csuse{LWR@depth#4}}{=}{\LWR@depthpart})\)\OR%
4222     \(\boolean{LWR@forcinghtmlpage})\)%
4223   \)%
4224   \AND%
4225   \cnttest{csuse{LWR@depth#4}}{<=}{\value{FileDepth}}%
4226   \AND%
4227   \(%
4228     \NOT\boolean{CombineHigherDepths}\OR%
4229     \cnttest{csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}%
4230   \)%
4231   \AND%

4232   \NOT\isempty{#3}% phantomsection
4233 }%
```

If so: start a new HTML file:

```
4234 {% new file
4235   \LWR@traceinfo{LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
4236   \IfNoValueTF{#2}%
```

If no optional entry

```
4237   {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
4238   {\LWR@newhtmlfile{#2}}%
4239 }% new file
```

Else: No new HTML file:

```
4240 {% not new file
```

Generate a new  $\LaTeX$  page so that toc and index page number points to the section:

```
4241 \LWR@traceinfo{LWR@section: not a new HTML file}%
4242 \LWR@orignewpage%
4243
4244}% not new file
```

Remember this section's name for \nameref:

```
4245 \IfValueT{#3}{%
4246 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
4247 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
4248}%
```

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

```
4249
4250 \ifbool{HTMLDebugComments}{%
4251 \begingroup%
4252 \LWR@nullfonts%
4253 \LWR@htmlcomment{Opening #4 ‘‘#3’’}%
4254 \endgroup%
4255 }{}%
4256
```

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

```
4257 \ifthenelse{%
4258 \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
4259 }%
4260 {\LWR@startpars}%
4261 {}%
```

Create the opening tag with an autosec:

```
4262 \LWR@createautosec{\csuse{LWR@tag#4}}%

4263 \setcounter{LWR@currentautosec}{\value{page}}
```

Check if starred:

```
4264 \IfBooleanTF{#1}%
4265 {%
```

Starred, but also forcing a TOC entry, so add unnumbered TOC name or regular name:

```
4266 \ifbool{LWR@forcinghtmltoc}%
4267 {\addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}}%
4268 {}%
4269 }% starred
```

Not starred, so step counter and add to TOC:

```
4270 {% not starred
```

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

```
4271 \ifthenelse{%
4272     \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
4273 }%
4274 {% if secnumdepth
```

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

```
4275 \LWR@traceinfo{LWR@section: about to test main matter}%
4276 \ifbool{LWR@mainmatter}%
4277 {%
4278     \LWR@traceinfo{LWR@section: yes mainmatter}%
4279     \refstepcounter{#4}%
```

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

```
4280 \LWR@traceinfo{LWR@section: about to addcontentsline}%
4281 \addcontentsline{toc}{#4}%
4282 {%
4283     \protect\numberline{\csuse{the#4}}%
4284     {\ignorespaces\IfValueTF{#2}{#2}{#3}\protect\relax}%
4285 }%
4286 \LWR@traceinfo{LWR@section: finished addcontentsline}%
4287 }% end of if main matter
```

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

```
4288 {% not main matter
4289     \LWR@traceinfo{LWR@section: no main matter}%
4290     \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4291 }% end of not main matter
4292 }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```

4293    {%
4294        \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4295    }%

```

For part, print the section type:

```

4296    \ifbool{LWR@mainmatter}%
4297    {%
4298        \ifthenelse{%
4299            \(\cnttest{\csuse{LWR@depth#4}}{<=}%
4300            {\value{secnumdepth}}\)\ AND%
4301            \(\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\)%
4302        }%
4303        {\csuse{#4name}~{}}%
4304        {}%

```

Print the section number:

```

4305        \LWR@traceinfo{LWR@section: about to print section number}%
4306        \ifthenelse{%
4307            \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
4308        }%
4309        {%
4310            \ifstrequal{#4}{chapter}%
4311            {\protect\LWR@sectionnumber{\@chapcntformat{#4}}}%
4312            {\protect\LWR@sectionnumber{\@secntformat{#4}}}%
4313        }%
4314        {}%
4315        \LWR@traceinfo{LWR@section: finished print section number}%
4316    }{}%
4317}% end of not starred

```

Print the section name:

```

4318 #3%

```

Close the heading tag, such as /H2:

```

4319 \LWR@htmltag{\csuse{LWR@tag#4end}}%

```

Generate a  $\TeX$  label:

```

4320 \label{autopage-\theLWR@currentautosec}%

```

Start paragraph handing unless is an inline paragraph or subparagraph:

```

4321 \ifthenelse{%
4322     \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}%

```

```

4323 }%
4324 {\LWR@startpars}%
4325 {}%

```

If not starred, remember the previous depth to possibly trigger a new HTML page.

HOWEVER, allow a `\part*` to start a new HTML page. This is used by appendix.

A starred section does not trigger a new HTML page at the beginning of this macro, so it should not affect it here at the end either. This became an issue when a `\listoftables` was tested in the middle of the document. The `\chapter*` for the list was not allowing a new HTML page for the section following it while `CombineHigherDepths` was true.

```

4326 \ifthenelse{%
4327     \NOT\equal{#1}{\BooleanTrue}\OR%
4328     \cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}%
4329 }%
4330 {% not starred
4331     \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
4332 }% not starred
4333 {}%

```

Reset to defaults if not a phantomsection:

```

4334 \ifstrempy{#3}%
4335 {}%
4336 {%
4337 \global\boolfalse{LWR@forcinghtmlpage}%
4338 \global\boolfalse{LWR@forcinghtmltoc}%
4339 }%
4340 %
4341 \LWR@traceinfo{LWR@section: done}%
4342 }

```

## 49.4 `\section` and friends

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

```

4343 \DeclareDocumentCommand{\part}{s o m}{%
4344 \LWR@maybeprintpendingfootnotes{\LWR@depthpart}%
4345 \LWR@stoppars%
4346
4347 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
4348
4349 \LWR@section{#1}{#2}{#3}{part}%

```

4350 }

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

```
4351 \let\@printcites\relax% for quotchap package
4352
4353 \@ifundefined{chapter}
4354 {}
4355 {%
4356 \DeclareDocumentCommand{\chapter}{s o m}{%
4357 \LWR@traceinfo{chapter #3}%
4358 \LWR@maybeprintpendingfootnotes{\LWR@depthchapter}%
4359 \LWR@stoppars%
4360
4361 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
4362
4363 \LWR@section{#1}{#2}{#3}{chapter}%
4364 \@printcites% for quotchap package
4365 }
4366 }
```

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

```
4367 \DeclareDocumentCommand{\section}{s o m}{%
4368 \LWR@maybeprintpendingfootnotes{\LWR@depthsection}%
4369 \LWR@stoppars%
4370
4371 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
4372
4373 \LWR@section{#1}{#2}{#3}{section}%
4374 }
```

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

```
4375 \DeclareDocumentCommand{\subsection}{s o m}{%
4376 \LWR@maybeprintpendingfootnotes{\LWR@depthsubsection}%
4377 \LWR@stoppars%
4378
4379 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
4380
4381 \LWR@section{#1}{#2}{#3}{subsection}%
4382 }
```

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

```
4383 \DeclareDocumentCommand{\subsubsection}{s o m}{%
4384 \LWR@maybeprintpendingfootnotes{\LWR@depthsubsubsection}%
4385 }
```

```

4385 \LWR@stoppars%
4386
4387 \LWR@startnewdepth{\LWR@depthsubsubsection}%
4388 {\LWR@printclosesubsubsection}%
4389
4390 \LWR@section{#1}{#2}{#3}{subsubsection}%
4391 }

```

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

```

4392 \DeclareDocumentCommand{\paragraph}{s o m}{%
4393 \LWR@maybeprintpendingfootnotes{\LWR@depthparagraph}%
4394 \LWR@stoppars%
4395
4396 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
4397
4398 \LWR@section{#1}{#2}{#3}{paragraph}%
4399 }

```

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

```

4400 \DeclareDocumentCommand{\subparagraph}{s o m}{%
4401 \LWR@maybeprintpendingfootnotes{\LWR@depthsubparagraph}%
4402 \LWR@stoppars%
4403
4404 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
4405
4406 \LWR@section{#1}{#2}{#3}{subparagraph}%
4407 }

4408 \end{warpHTML}

```

## 50 Starting a new file

**for HTML & PRINT:** 4409 \begin{warpall}

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

```

4410 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
4411
4412 \newcommand*{\HTMLLanguage}[1]{%
4413 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
4414 }

```

```
4415 \end{warpall}
```

**for HTML output:** 4416 \begin{warpHTML}

```
\LWR@filestart {\title_suffix}
```

Creates the opening HTML tags.

```
4417 \newcommand*{\LWR@filestart}[1]{
4418 \LWR@traceinfo{\LWR@filestart}
```

Locally temporarily disable direct-formatting commands:

```
4419 \begingroup
4420 \LWR@traceinfo{\LWR@filestart: A}
4421 \LWR@nullfonts
4422 \LWR@traceinfo{\LWR@filestart: B}
```

Create the page's HTML header:

```
4423 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
4424 \LWR@traceinfo{\LWR@filestart: C}
```

The language is user-adjustable:

```
4425 \LWR@htmltag{html lang="\LWR@currentHTMLLanguage"{}}\LWR@orignewline
```

Start of the meta data:

```
4426 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
4427 \LWR@htmltag{meta charset="UTF-8" /\LWR@orignewline
```

Author:

```
4428 \ifcempty{theHTMLAuthor}{}{
4429 \LWR@htmltag{meta name="author" content="\theHTMLAuthor" /\LWR@orignewline
4430 }
```

lwarp is the generator:

```
4431 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /\LWR@orignewline
4432 \LWR@orignewline
```



If there is a description, add it now:

```
4433 \ifdefempty{\LWR@currentHTMLDescription}{-}{-%
4434 \LWR@htmltag{%
4435 meta name="description" content="\LWR@currentHTMLDescription" /}%
4436 \LWR@orignewline
4437 }%
```

Mobile-friendly viewport:

```
4438 \LWR@htmltag{meta name="viewport" %
4439 content="width=device-width, initial-scale=1.0" /}%
4440 \LWR@orignewline
```

IE patch:

```
4441 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
4442 \LWR@htmltag{%
4443 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{}-%
4444 \LWR@htmltag{/script}\LWR@orignewline
4445 \LWR@htmltag{![endif]{-}{-}}\LWR@orignewline
```

The page's title:

```
4446 \ifcvoid{thetitle}{-}{-%
4447 \LWR@htmltag{title}\thetitle#1\LWR@htmltag{/title}\LWR@orignewline%
4448 }%
```

The page's stylesheet:

```
4449 \LWR@htmltag{%
4450 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
4451 \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.

```
4452 \ifbool{mathjax}%
4453 {%
4454 \boolfalse{LWR@verbtags}
4455 \verbatiminput{lwarp_mathjax.txt}%
4456 \booltrue{LWR@verbtags}
4457 \LWR@stoppars
4458 }% end of mathjax
4459 {}%
```

End of the header:

```
4460 \LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
4461 \LWR@htmltag{body}\LWR@orignewline
4462 \endgroup
4463 \LWR@traceinfo{LWR@filestart: done}
4464 }
```

```
4465 \end{warpHTML}
```

## 51 Starting HTML output

**for HTML output:** 4466 \begin{warpHTML}

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

```
4467 \catcode'\$=\active
4468 \newcommand*{\LWR@LwarpStart}
4469 {%
4470 \LWR@traceinfo{LWR@lwarpStart}
```

If formatting for a word processor, force filedepth to single-file only, force HTML debug comments off.

```
4471 \ifbool{FormatWP}{%
4472   \setcounter{FileDepth}{-5}%
4473   \boolfalse{HTMLDebugComments}%
4474 }{}
```

Expand and detokenize `\HomeHTMLFilename` and `\HTMLFilename`:

```
4475 \edef\LWR@strresult{\HomeHTMLFilename}
4476 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
4477 \edef\LWR@strresult{\HTMLFilename}
4478 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn and empty page style:

```
4479 \LWR@origonecolumn%
4480 \LWR@origpagestyle{empty}
```

Reduce chance of line overflow in verbatim environments:

```
4481 \LWR@origscriptsize%
```

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

```
4482 \LWR@origraggedright%
4483 \LetLtxMacro{\}\{\LWR@endoffline}%
```

Spread the lines for pdftotext to read them well:

```
4484 \linespread{1.3}%
```

For pdftotext to reliably identify paragraph splits:

```
4485 \setlength{\parindent}{0pt}
4486 \setlength{\parskip}{2ex}
```

For the lateximages record file:

```
4487 \immediate\openout\LWR@lateximagesfile=lateximages.txt
```

Removes space after the caption in the HTML:

```
4488 \setlength{\belowcaptionskip}{-3ex}
```

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

```
4489 \renewcommand{\ps@plain}{}%
```

```
\centering Not used in the HTML environment:
\raggedleft
\raggedright 4490 \renewcommand*{\centering}{}
4491 \renewcommand*{\raggedleft}{}
4492 \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:

```
4493 \LetLtxMacro{\LWR@origtabular}{\tabular}
4494 \LetLtxMacro{\LWR@origendtabular}{\endtabular}
4495 \LetLtxMacro{\tabular}{\LWR@tabular}
4496 \LetLtxMacro{\endtabular}{\endLWR@tabular}
```

Float captions:

```
4497 \let\LWR@origcaption\caption
```

Labels: \ltx@label is used in amsmath environments and is also patched by cleveref.

[Label in HTML](#)

```
4498 \let\LWR@origltx@label\ltx@label
4499 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
4500 \let\TeX\LWR@TeX
4501 \let\LaTeX\LWR@LaTeX
4502 \let\LuaTeX\LWR@LuaTeX
4503 \let\LuaLaTeX\LWR@LuaLaTeX
4504 \let\XeTeX\LWR@XeTeX
4505 \let\XeLaTeX\LWR@XeLaTeX
4506 \let\ConTeXt\LWR@ConTeXt
```

Not yet started any paragraph handling:

```
4507 \global\boolfalse{LWR@doingapar}
4508 \global\boolfalse{LWR@doingstartpars}
```

Start a new HTML file and a header:

```
4509 \LWR@traceinfo{LWR@lwarpStart: Starting new file.}
4510 \LWR@filestart{}
4511 \LWR@traceinfo{LWR@lwarpStart: Generating first header.}
4512 \LWR@htmltag{header}\LWR@orignewline
4513 \LWR@startpars
4514 \LWR@firstpagetop
4515 \LWR@stoppars
4516 \LWR@htmltag{/header}\LWR@orignewline
4517 \LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
4518 \LWR@htmltag{section class="textbody"{} }
```

Document and page settings:

```
4519 \mainmatter
4520 \LWR@origpagenumbering{arabic}
```

Patch the itemize, enumerate, and description environments and `\item`. This works with the native  $\TeX$  environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```
4521 \LWR@patchlists
```

Ensure that math mode is active to call lwarp's patches:

```
4522 \catcode'\$=\active
```

Allow HTML paragraphs to begin:

```

4523 \LWR@startpars
4524 \LWR@traceinfo{LWR@lwarpStart: done}
4525 }
4526 \catcode'\$=3% math shift until lwarp starts
4527 \end{warpHTML}

```

## 52 Ending HTML output

**for HTML output:** 4528 \begin{warpHTML}

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

```

4529 \newcommand*{\LWR@requesttoc}[2]{%
4530 \ifbool{#1}
4531 {
4532     \expandafter\newwrite\csuse{tf@#2}
4533     \immediate\openout \csuse{tf@#2} \jobname.#2\relax
4534 }{}
4535 }

```

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

```

4536 \newcommand*{\LWR@LwarpEnd}
4537 {
4538 \LWR@stoppars
4539 \LWR@closeprevious{\LWR@depthfinished}

```

At the bottom of the ending file:

Close the textbody:

```
4540 \LWR@htmlclassend{section}{textbody}
```

Print any pending footnotes:

```
4541 \LWR@printpendingfootnotes
```

Create the footer:

```

4542 \LWR@htmlclassend{footer}
4543
4544 \LWR@pagebottom
4545
4546 \LWR@htmlclassend{footer}

```

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

Presumably has a table-of-contents.

```
4547 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4548 {}
4549 {
4550     \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{-}
4551 }
```

```
4552 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
4553 \LWR@htmltag{/body}\LWR@orignewline
4554 \LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
4555 \LWR@orignewpage
```

For lateximage commands:

```
4556 \immediate\closeout\LWR@lateximagesfile
4557 }
```

```
4558 \end{warpHTML}
```

## 53 Title page

**package support** lwarp supports the native  $\TeX$  titling commands, and also supports the packages **authblk** and **titling**. If both are used, **authblk** should be loaded before **titling**.





**load order**

**\published and \subtitle** If using the titling package, additional titlepage fields for **\published** and **\subtitle** may be added by using **\AddSubtitlePublished** in the preamble. See section 53.7.

**affiliation** lwarp provides for the **\author** macro an additional **\affiliation** macro to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using titlingpage's **\theauthor** in the main text.

**reusing titlepage information** The titling package maintains the definitions of **\thetitle**, **\theauthor**, etc., after the title has been typeset. These commands are to be used to refer to the document's title and author, etc., in the main text. These definitions have the **\thanks** and

-  `\theauthor`, `authblk` `\affiliation` removed, and for `\author` the `\and` is replaced to generate a simple inline list of authors separated by commas. Note: `\theauthor` does not work well with `authblk` unless the traditional  $\LaTeX$  syntax is used.
- `custom titlepages` `\printtitle`, `\printauthor`, etc., are provided for use inside a custom titlepage or titlingpage environment, and these retain the `\thanks` and `\affiliation`.
- `\printthanks` `\printthanks` has been added to force the printing of thanks inside a titlingpage environment when `\maketitle` is not used.
-  Inside a `\titlepage` or `\titlingpage` environment, use `\thanks` instead of `\footnote` for acknowledgements, etc.

### 53.1 Setting the title, etc.

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

`\author` `\and` `{\author}` While using `\maketitle` and print mode, the author is treated as a single-column tabular and the `\and` feature finishes the current tabular then starts a new one for the next author. Each author thus is placed into its own tabular, and an affiliation may be placed on its own line such as

```
\author{Name \\ Affiliation \and Second Name \\ Second Affiliation}
```

For HTML, the entire author block is placed inside a `<div>` of class `author`, and each individual author is inside a `<div>` of class `oneauthor`.

`\@title` `\@author` `\@date` `\@title`, `\@author`, etc. store the values as originally assigned, including any `\thanks`, `\and`, or `\affiliation`. These are low-level macros intended to be used by other macros only inside a titlepage or titlingpage, and are used by `\maketitle`. The author is printed inside a single-column tabular, which becomes multiple single-column tabulars if multiples authors are included. For HTML these tabulars become side-by-side `<div>`s of class `oneauthor`, all of which are combined into one `<div>` of class `author`.

`\printtitle` `\printauthor` `\printdate` `\printtitle`, etc. are user-level macros intended to be used in custom titlepage or titlingpage environments in cases where `\maketitle` is not desired. These commands preserve the `\thanks`, etc., and should not be used in the main text.


`\thetitle` `\theauthor` `\thedata` `\thetitle`, `\theauthor`, and `\thedata` are available if titling has been loaded, and are sanitized user-level versions from which have been removed the `\thanks` and `\affiliation`, and `\and` is changed for inline text usage. The author is printed inline without `\affiliation` or `\thanks`, with `\and` placing commas between multiple authors. Thus, these commands are to be used in the main text whenever the user wishes to refer to the document's title and such. One practical use for this is to place the authors at the bottom of each HTML page, such as:


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

---

```
\HTMLPageBottom{
\begin{center}\textcopyright~2016 \theauthor\end{center}
}
```

---

 **\theauthor** `\theauthor` does not work well if `authblk` is used. If `\theauthor` is important, it is recommended to use the standard  $\TeX$  syntax for `\author`, optionally with `lwarp`'s `\affiliation` macro as well.

 **affiliations** After `\maketitle` has completed, `\theauthor` retains the definition of the author, but `\and` is changed to become a comma and a space, intending to print the authors names separated by spaces. This fails when affiliations are included on their own table rows.

**\affiliation** A solution, provide here, is to define a macro `\affiliation` which, during `\maketitle`, starts a new row and adds the affiliation, but after `\maketitle` is finished `\affiliation` is re-defined to discard its argument, thus printing only the author names when `\author` is later used inline.

## 53.2 Changes for `\affiliation`

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

Adds the affiliation to the author for use in `\maketitle`.

Inside `titlepage`, this macro prints its argument. Outside, it is null.

**for HTML & PRINT:**

```
4559 \begin{warppall}
4560 \newrobustcmd{\affiliation}[1]{\}
4561 \end{warppall}
```

**for PRINT output:**

```
4562 \begin{warpprint}

4563 \AtBeginEnvironment{titlepage}{\}
4564 \renewrobustcmd{\affiliation}[1]{\ \textsc{\small#1}}
4565 }
4566
4567 \AtBeginDocument{
4568 \@ifpackageloaded{titling}{
4569 \AtBeginEnvironment{titlingpage}{
4570 \renewrobustcmd{\affiliation}[1]{\ \textsc{\small#1}}
4571 }
4572 }{}% titling loaded
4573 }% AtBeginDocument
```



```
4574 \end{warpprint}
```

**for HTML output:** 4575 \begin{warpHTML}

Env titlepage Sets up a <div> of class titlepage.

```
4576 \renewenvironment*{titlepage}
4577 {
4578 \renewrobustcmd{\affiliation}[1]{\ \InlineClass{affiliation}{##1}}
4579 \LWR@printpendingfootnotes
4580 \LWR@forcenewpage
4581 \BlockClass{titlepage}
4582 }
4583 {
4584 \endBlockClass
4585 \LWR@printpendingfootnotes
4586 }

4587 \end{warpHTML}
```

### 53.3 Printing the thanks

**for HTML & PRINT:** 4588 \begin{warppall}

\printthanks Forces the \thanks to be printed.

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

```
4589 \newcommand*{\printthanks}{\@thanks}

4590 \end{warppall}
```

### 53.4 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:** 4591 \begin{warpHTML}

\printtitle

```
4592 \newcommand*{\printtitle}
4593 {
```

```

4594 \LWR@stoppars
4595 \LWR@htmltag{\LWR@tagtitle}%
4596 \@title%
4597 \LWR@htmltag{\LWR@tagtitleend}
4598 \LWR@startpars
4599 }

```

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

```

4600 \newcommand*{\LWR@printthetitle}
4601 {
4602 \LWR@stoppars
4603 \LWR@htmltag{\LWR@tagtitle}%
4604 \thetitle%
4605 \LWR@htmltag{\LWR@tagtitleend}
4606 \LWR@startpars
4607 }

```

`\printauthor` HTML version.

```

4608 \newcommand*{\printauthor}{

```

The entire author block is contained in a <div> named author:

```

4609 \begin{BlockClass}{author}

```

`\and` finishes one author and starts the next:

```

4610 \renewcommand{\and}{%
4611 \end{BlockClass}
4612 \begin{BlockClass}{oneauthor}
4613 }

```

Individual authors are contained in a <div> named oneauthor:

```

4614 \begin{BlockClass}{oneauthor}
4615 \@author
4616 \end{BlockClass}
4617 \end{BlockClass}
4618 }

```

`\printdate`

```

4619 \newcommand*{\printdate}{%
4620 \begin{BlockClass}{titledate}

```

```

4621 \@date
4622 \end{BlockClass}
4623 }

4624 \end{warpHTML}

```

### 53.5 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:** 4625 \begin{warpprint}

\printtitle

```

4626 \newcommand*{\printtitle}{\Huge\@title}

```

\printauthor    Print mode.

```

4627 \newcommand*{\printauthor}
4628    {\large\begin{tabular}[t]{c}\@author\end{tabular}}

```

\printdate

```

4629 \newcommand*{\printdate}{\small\textit{\@date}}

4630 \end{warpprint}

```

### 53.6 \maketitle for HTML output

An HTML <div> of class titlepage is used.

\thanks are a form of footnotes used in the title page. See section 46 for other kinds of footnotes.

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

**for HTML output:** 4631 \begin{warpHTML}

```

4632 \if@titlepage
4633 \newcommand{\LWR@setfootnoteseries}{%
4634    \renewcommand\thefootnote{\@arabic\c@footnote}%

```

```

4635 }
4636 \else
4637 \newcommand{\LWR@setfootnoteseries}{%
4638     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
4639 }
4640 \fi

```

\LWR@maketitlesetup Patches \thanks macros.

```

4641 \newcommand*{\LWR@maketitlesetup}{%

```

Redefine the footnote mark:

```

4642 \LWR@setfootnoteseries%
4643 \def\@makefnmark{\textsuperscript{\thefootnote}}

```

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

Redefine the footnote text:

```

4644 \long\def\@makefntext##1{%

```

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

```

4645 \textsuperscript{\@thefnmark} \LWR@orighspace{1in}

```

\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒  
 \@thefnmark ⇒ \itshape a (or similar)

Print the text:

```

4646 ##1%
4647 }%
4648 }

```

\@fnsymbol {<counter>}

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used \| which was not being seen by pdftotext.

```

4649 \def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
4650     \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLUnicode{2016}}\or
4651     **\or \HTMLentity{dagger}\HTMLentity{dagger} \or
4652     \HTMLentity{Dagger}\HTMLentity{Dagger} \else\@ctrerr\fi}

```

`\maketitle` HTML mode. Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

```
4653 \renewcommand*{\maketitle}{%
```

An HTML titlepage <div> is used for all classes.

```
4654 \begin{titlepage}
```

Set up special patches:

```
4655 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4656 \@maketitle
```

Immediately generate any \thanks footnotes:

```
4657 \@thanks
```

Close the HTML titlepage div and cleanup:

```
4658 \end{titlepage}
4659 \setcounter{footnote}{0}%
4660 \global\let\thanks\relax
4661 \global\let\maketitle\relax
4662 \global\let\@maketitle\relax
4663 \global\let\@thanks\@empty
4664 \global\let\@author\@empty
4665 \global\let\@date\@empty
4666 \global\let\@title\@empty
4667 \global\let\title\relax
4668 \global\let\author\relax
4669 \global\let\date\relax
4670 \global\let\and\relax
4671 }
```

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

```
4672 \DeclareDocumentCommand{\@maketitle}{-}{%
4673   \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
4674   \@title
4675   \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
4676   \begin{BlockClass}{author}
4677   \renewcommand{\and}{
```

```

4678         \end{BlockClass}
4679         \begin{BlockClass}{oneauthor}
4680     }
4681     \begin{BlockClass}{oneauthor}
4682         \@author
4683     \end{BlockClass}
4684 \end{BlockClass}
4685 \begin{BlockClass}{titledate}
4686     \@date
4687 \end{BlockClass}
4688 }
```

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

```
4689 \newcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
4690 \@thanks
```

Set up special patches:

```
4691 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4692 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
4693 \@thanks
```

```
4694 }
```


```
4695 \end{warpHTML}
```

### 53.7 `\published` and `\subtitle`

`\subtitle` and `\published` To add `\subtitle` and `\published` to the titlepage, load the titling package and use `\AddSubtitlePublished` in the preamble.

The default `lwarp.css` has definitions for the `published` and `subtitle` classes.

After titling is loaded, `\AddSubtitlePublished` is created, which when used then creates a number of additional macros, and also assigns some of the titling hooks.

 **titling hooks** Do not use `\AddSubtitlePublished` if the user has patched the titling hooks for

some other reason. Portions are marked `\warpprintonly` to reduce extra tags in HTML. Similarly, `BlockClass` has no effect in print mode. Thus, the following may be marked `warpall`.

**for HTML & PRINT:** 4696 `\begin{warpall}`

`\AddSubtitlePublished` Adds `\published` and `\subtitle`, and related.

```

4697 \AfterPackage{titling}{
4698 \newcommand*{\AddSubtitlePublished}{%
4699
4700 \newcommand{\@published}{%
4701
4702 \newcommand{\published}[1]{\gdef\@published{##1}}
4703
4704 \renewcommand*{\maketitlehooka}{\printpublished}
4705
4706 \newcommand*{\printpublished}{%
4707 \warpprintonly{\begin{center}\unskip}%
4708 \begin{BlockClass}{published}%
4709 \warpprintonly{\large\itshape}%
4710 \@published%
4711 \end{BlockClass}%
4712 \warpprintonly{\end{center}}%
4713 }
4714
4715 \newcommand{\@subtitle}{%
4716
4717 \newcommand{\subtitle}[1]{\gdef\@subtitle{##1}}
4718
4719 \renewcommand*{\maketitlehookb}{\printsubtitle}
4720
4721 \newcommand*{\printsubtitle}{%
4722 \warpprintonly{\begin{center}\unskip}%
4723 \begin{BlockClass}{subtitle}%
4724 \warpprintonly{\Large\itshape}%
4725 \@subtitle%
4726 \end{BlockClass}%
4727 \warpprintonly{\end{center}}%
4728 }
4729
4730 }% \AddSubtitlePublished
4731 }% AfterPackage

4732 \end{warpall}

```

## 54 Abstract

The following code replaces the  $\LaTeX$  default, and will itself be replaced later if the abstract package is loaded.

```

for HTML output: 4733 \begin{warpHTML}

\abstractname User-redefinable title for the abstract.

                Also over-written by the babel package.

4734 \providecommand*{\abstractname}{Abstract}

Env abstract

4735 \DeclareDocumentEnvironment{abstract}{}
4736 {
4737 \LWR@forcenewpage
4738 \BlockClass{abstract}
4739 \BlockClassSingle{abstracttitle}{\abstractname}
4740 }
4741 {
4742 \endBlockClass
4743 }

4744 \end{warpHTML}

```

## 55 Quote and verse

### 55.1 Citations and attributions

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

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

```

for HTML output: 4745 \begin{warpHTML}
4746 \newcommand{\attribution}[1]{%
4747 \InlineClass{attribution}{--\,#1}}% emdash
4748 \end{warpHTML}

```

```

for PRINT output: 4749 \begin{warpprint}
4750 \newcommand{\attribution}[1]{\textsc{---\,#1}}
4751 \end{warpprint}

```



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

```

for HTML output: 4752 \begin{warpHTML}
                   4753 \newcommand{\citetitle}[1]{%
                   4754 \InlineClass{citetitle}{--\,#1}}% emdash
                   4755 \end{warpHTML}

for PRINT output: 4756 \begin{warpprint}
                   4757 \newcommand{\citetitle}[1]{\textsl{---\,#1}}
                   4758 \end{warpprint}

```

## 55.2 Quotes, quotations

```

for HTML output: 4759 \begin{warpHTML}

```

Env `quote`

```

4760 \renewenvironment*{quote}
4761 {
4762 \LWR@forcenewpage
4763 \LWR@htmlblocktag{blockquote}
4764 }
4765 {\LWR@htmlblocktag{/blockquote}}
4766
4767 \renewenvironment*{quotation}
4768 {
4769 \LWR@forcenewpage
4770 \LWR@htmlblocktag{blockquotation}
4771 }
4772 {\LWR@htmlblocktag{/blockquotation}}

4773 \end{warpHTML}

```

## 55.3 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` 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 (`-layout` option) of the text in the HTML-tagged PDF output. For some settings of `\HTMLleftmargini` or `\HTMLleftskip` the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

**for HTML & PRINT:** 4774 `\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.

```

4775 \newlength{\HTMLvleftskip}
4776 \setlength{\HTMLvleftskip}{1em}

```

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

```

4777 \newlength{\HTMLleftmargini}
4778 \setlength{\HTMLleftmargini}{4.5em}

4779 \end{warpall}

```

## 56 Verbatim and tabbing

**for HTML & PRINT:** 4780 `\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.

```
4781 \newlength{\VerbatimHTMLWidth}
4782 \setlength{\VerbatimHTMLWidth}{4in}
4783 \end{warpall}
```

**for HTML output:** 4784 \begin{warpHTML}

Bool LWR@verbtags Used to temporarily turn off verbatim tags while doing \verbatiminput in the HTML head.

```
4785 \newbool{LWR@verbtags}
4786 \booltrue{LWR@verbtags}
```

\LWR@atbeginverbatim [*<style>*] [*<class>*]

Encloses a verbatim environment with the given css class.

```
4787 \newcommand*{\LWR@atbeginverbatim}[2] []
4788 {%
```

Avoid excessive space between lines:

```
4789 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
4790 \LWR@stoppars%
```

Create a new pre of the given class. The tags may temporarily be turned off for internal use, such as loading the MathJax script.

```
4791 \ifbool{LWR@verbtags}{%
4792   \LWR@htmltag{pre class="#2"
4793   \ifthenelse{\equal{#1}{}}{}{style="#1"}}%
4794 }% pre
4795 }{}%
```

Use a mono-spaced font to preserve horizontal positioning. If horizontal alignment is important for the user, use a mono-spaced font in the css for the verse class.

Also turn off babel-french extra space before punctuation:

```
4796 \begingroup%
4797 \LWR@origttfamily%
4798 \LWR@FBcancel%
```

Do not produce HTML tags for \hspace inside a verse par. Restore plain  $\TeX$  \hspace functionality:

```
4799 \LetLtxMacro{\hspace}{\LWR@origvspace}%
4800 }
```

`\LWR@afterendverbatim` Finishes enclosing a verbatim environment.

```
4801 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the pre:

```
4802 \endgroup%
4803 \unskip%
```

At the end of the environment, close the pre:

```
4804 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}}
4805
4806 }{}%
```

Resume regular paragraph handling:

```
4807 \LWR@startpars%
4808 }
```

`\verbatiminput` `{<filename>}`

Patch \verbatiminput to add HTML tags:

```
4809 \let\LWRV@origverbatim@input\verbatim@input
4810
4811 \renewcommand{\verbatim@input}[2]{%
4812 \ifbool{LWR@verbtags}{\LWR@forcenewpage}{}%
4813 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%%
4814 \LWRV@origverbatim@input{#1}{#2}%
4815 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
4816 }
```

Env `verbatim`

```
4817 \AfterEndPreamble{
4818 \LWR@traceinfo{Patching verbatim.}
4819 \AtBeginEnvironment{verbatim}{%
4820 \LWR@forcenewpage
4821 \LWR@atbeginverbatim{verbatim}\unskip\LWR@origvspace*{-\baselineskip}%%
4822 }
```

```

4823 \AfterEndEnvironment{verbatim}{%
4824     \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
4825 }
4826 }

```

Env    `tabbing`

```

4827 \AfterEndPreamble{
4828 \LWR@traceinfo{Patching tabbing.}
4829 \AtBeginEnvironment{tabbing}{%
4830 \LWR@forcenewpage
4831 \LWR@atbeginverbatim{tabbing}\unskip\LWR@origvspace*{-\baselineskip}%
4832 }
4833 \AfterEndEnvironment{tabbing}{%
4834     \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
4835 }
4836 }

4837 \end{warpHTML}

```

## 57 Theorems

`\newtheorem`     $\{\langle text \rangle\} [\langle counter \rangle] \text{-or-} [\langle oldname \rangle] \{\langle text \rangle\}$

A few minor changes are made to supply HTML tags.

- The entire theorem is placed into a `<div>` of class `theoremcontents`.
- The label for each theorem is placed inside a `<span>` of class `theoremlabel`.
- The contents are placed inside a `<div>` of class `theoremcontents`.

**for HTML output:**    4838 `\begin{warpHTML}`

`\@begintheorem`     $\{\langle name \rangle\} \{\langle number \rangle\}$

```

4839 \renewcommand{\@begintheorem}[2]{%
4840 \LWR@forcenewpage
4841 \BlockClass{theoremcontents}
4842 \InlineClass{theoremlabel}{#1\ #2\ }
4843 }

```

`\@opargbegintheorem`     $\{\langle name \rangle\} \{\langle number \rangle\} \{\langle oparg \rangle\}$

```

4844 \renewcommand{\@opargbegintheorem}[3]{%
4845 \LWR@forcenewpage
4846 \BlockClass{theoremcontents}
4847 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
4848 }

```

`\@endtheorem`

```

4849 \renewcommand*\@endtheorem{%
4850 \endBlockClass% theoremcontents
4851 }

```

```

4852 \end{warpHTML}

```

## 58 Lists

The environments `itemize`, `enumerate`, and `description` are patched when `lwarp` is started. These patches support the standard  $\TeX$  environments, as well as those of `enumerate`, `enumitem`, and `paralist`, and at least the French version of `babel`. Additional patches are done on a package-specific basis.

The  $\TeX$  source for `itemize` and `enumerate` are found in `source2e`, but the source for `description` is found in `article.cls`, etc.

**empty item** To have an empty item, use `\mbox{}` or a trailing backslash. This forces a new line in print output, matching the new line which will appear in HTML output. Ex:

---

```

begin{itemize}
item \mbox{}
    \begin{itemize}
...
    \end{itemize}
item \
    \begin{itemize}
...
    \end{itemize}

```

---

### 58.1 Itemize

**for HTML output:** 4853 `\begin{warpHTML}`

`\LWR@itemizeitem` [*<label>*]

Handles `\item` inside an itemize or enumerate.

See `\LWR@openparagraph` where extra `\hspace` is used to leave room for the label while inside a list during paragraph construction.

```
4854 \newcommand*{\LWR@itemizeitem}{%
4855 \LWR@stoppars%
4856 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloseitemize{}}%
4857 \LWR@htmltag{li}%
4858 \LWR@startpars%
4859 \LWR@origitem%
4860 }
```

Env `itemize` [*<options>*]

```
4861 \newcommand*{\LWR@itemizestart}{%
4862 \LWR@stoppars%
4863 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4864 \LWR@htmltag{ul style="list-style-type:none"{}}%
4865 \LWR@startpars%
4866 \let\item\LWR@itemizeitem%
4867 }
4868
4869 \newcommand*{\LWR@itemizeend}{%
4870 \LWR@stoppars%
4871 \LWR@closeprevious{\LWR@depthlistitem}%
4872 \LWR@closeoneprevious{}}%
4873 \LWR@startpars%
4874 }
```

## 58.2 Enumerate

An HTML unordered list is used with customized  $\text{\LaTeX}$ -generated labels.

Env `enumerate` [*<options>*]

```
4875 \newcommand*{\LWR@enumeratestart}{%
4876 \LWR@stoppars%
4877 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4878 \LWR@htmltag{ul style="list-style-type:none"{}}%
4879 \LWR@startpars%
4880 \let\item\LWR@itemizeitem%
4881 }
4882
```

```

4883 \newcommand*{\LWR@enumerateend}{%
4884 \LWR@stoppars%
4885 \LWR@closeprevious{\LWR@depthlistitem}%
4886 \LWR@closeoneprevious{}}%
4887 \LWR@startpars%
4888 }

```

### 58.3 Description

`\LWR@descitem` [*<label>*] Handles an `\item` inside a description.

```

4889 \newcommand*{\LWR@descitem}[1][]%
4890 {%
4891 \LWR@stoppars%
4892 \LWR@setlatestname{#1}%
4893 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem{}}%

```

Temporarily disable `\hspace`, which `article.cls`, etc. use per `\item` for descriptions only. This causes `lwarp` to mistakenly place an empty span between HTML list tags.

```

4894 \LetLtxMacro{\hspace}{\LWR@nohspace}%

```

Process the original `\item` code:

```

4895 \LWR@origitem[]%

```

Restore `\hspace` for use in the item text:

```

4896 \LetLtxMacro{\hspace}{\LWR@hspace}%

```

Be sure the label doesn't print to the left of the rest of the file:

```

4897 \LWR@orighspace{1in}
4898 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
4899 \LWR@orignewline%
4900 \LWR@htmltag{dd}%
4901 \LWR@startpars%
4902 }

```

Env `description` [*<options>*]

```

4903 \newcommand*{\LWR@descriptionstart}{%
4904 \LWR@stoppars%
4905 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}%
4906 \LWR@htmltag{dl}%

```



```

4907 \LWR@startpars%
4908 \let\item\LWR@descitem%
4909 }
4910
4911 \newcommand*\LWR@descriptionend}{%
4912 \LWR@stoppars%
4913 \LWR@closeprevious{\LWR@depthlistitem}%
4914 \LWR@closeoneprevious{}%
4915 \LWR@startpars%
4916 }

```

`\LWR@patchlists` Patches list environments.

`\LWR@patchlists` remembers `\item` as defined by whatever packages have been loaded, then patches the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native  $\TeX$  environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```

4917 \newcommand*\LWR@patchlists}{%
4918 \let\item\LWR@origitem}\item}
4919 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
4920 \AtEndEnvironment{itemize}{\LWR@itemizeend}
4921 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
4922 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
4923 \AtBeginEnvironment{description}{\LWR@descriptionstart}
4924 \AtEndEnvironment{description}{\LWR@descriptionend}
4925 }

4926 \end{warpHTML}

```

## 59 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

### 59.1 Limitations

Tabular mostly works as expected, but pay special attention to the following, especially if working with environments, macros inside tabulars, multirows, \* column specifiers, siunitx  $S$  columns, or the packages `multirow`, `longtable`, `supertabular`, or `xtab`.

#### Defining environments:

⚠ misplaced alignment  
alignment tab character &

⚠ tabular inside another  
environment

- When defining environments or macros which include tabular and instances of the & character, it may be necessary to make & active before the environment or macro is defined, then restore & to its default catcode after, using the following commands. These are ignored in print mode.

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

- When creating a new environment which contains a tabular environment, lwarp's emulation of the tabular does not automatically resume when the containing environment ends, resulting in corrupted HTML rows. To fix this, use \ResumeTabular as follows. This is ignored in print mode.

```
\StartDefiningTabulars % because & is used in a
definition
\newenvironment{outerenvironment}
{
\tabular{cc}
left & right \\
}
{
\TabularMacro\ResumeTabular
left & right \\
\endtabular
}
\EndDefiningTabulars
```

### Cell contents:

⚠ paragraphs

⚠ \multirow

- Multiple paragraphs in one cell of a p, b, m column must have \newline between paragraphs.
- For multirow, insert \mrowcell into any empty multi-row cells. This will be a null function for the print output, and is a placeholder for parsing the table for HTML output.

```
... & \multirow{2}{.5in}{text} & ...
... & \mrowcell & ...
```

vposn

Note that recent versions of multirow include a new optional vposn argument.

- The multirow documentation regarding colored cells recommends using a negative number of rows. This will not work with lwarp, so \warpprintonly and \warpHTMLonly must be used to make versions for print and HTML.
- See section 180.2 for \multicolumnrow.

lwarp does not support directly combining \multicolumn and \multirow. Use \multicolumnrow instead. To create a 2 column, 3 row cell:

⚠ \multicolumn &  
\multirow

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[Opt]{Text}
```

The two arguments for `\multicolumn` come first, followed by the five arguments for `\multirow`, many of which are optional, followed by the contents.

⚠ skipped cells

⚠ empty cells

As per `\multirow`, skipped cells to the right of the `\multicolumnrow` statement are not included in the source code on the same line. On the following lines, `\mcolrowcell` must be used for each cell of each column and each row to be skipped:

```
... & \multicolumnrow{2}{c}[c]{3}[0]{1in}[Opt]{Text} & ...
... & \mcolrowcell & \mcolrowcell & ...
... & \mcolrowcell & \mcolrowcell & ...
```

`vposn`

Note that recent versions of `multirow` include a new optional `vposn` argument.

⚠ macro in a table  
custom macros

- Using a custom macro inside a tabular data cell may result in an extra HTML data cell tag, corrupting the HTML table. To avoid this, use `\TabularMacro` just before the macro. This is ignored in print mode.

```
\TabularMacro\somemacro & more row contents \
```

#### Column specifiers:

⚠ \* column specification

@ and !

`\multirow`

⚠ `\newcolumntype`

- \* in a column specification is not used (so far). Repeat the column type the correct number of times.
- Only one each of @ and ! is used at each column, and they are used in that order.
- In `\multirow` cells, the print version may have extra instances of <, >, @, and ! cells on the second and later rows in the `\multirow` which do not appear in the HTML version.
- `\newcolumntype` is ignored; unknown column types are set to l.

#### Rules:

vertical rules

width and trim

full-width rules

combined rules

- Vertical rules next to either side of an @ or ! column are displayed on both sides of the column.
- Width options are honored. Trim options are converted to rounded top corners. Trim corners are not rounded with @ or ! columns, and full-width rules ignore trim.
- `\toprule`, `\midrule`, `\bottomrule`, and `\hline` ignore trim. When given an optional width, each cell is styled to create the custom border. Without an optional width, the entire row is given a class to assign the standard border.
- If you wish to use `\cmidrule` followed by `\bottomrule`, it may be necessary to use:
 

```
\cmidrule{2-3} \[-2ex]
\bottomrule
```

⚠ `\warpprintonly`  
misplaced `\noalign`

The optional `-2ex` is ignored in HTML but improves the visual formatting in the print output.

- For `\toprule` and `\bottomrule`, when combined with a `warpprint` or `warpHTML` environment, if a “misplaced `\noalign`” error occurs, change

`This & That \endhead`

to

`\warpprintonly{This & That \endhead}`

and likewise with the other `\end` headings. Keep the `\endfirsthead` row unchanged, as it is still relevant to HTML output.

#### Other:

`longtable` headings

⚠ `S` columns

- `tabularx` ignores the width, but `X` columns do produce paragraph columns or multicolumns.
- For `longtable`, place headings and footings which do not apply to HTML inside `\warpprintonly{}`.
- For `S` columns (from the `siunitx` package), while producing print output, anything non-numeric must be placed inside `{ }` braces, including commands such as `\multirow`. While producing HTML output, though, anything placed inside braces is not seen by `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 \\}`

## 59.2 Token lookahead

Used by `\LWR@futurenonSPACElet` to look at the next token.

for HTML output: 4927 `\begin{warpHTML}`

`\LWR@mynexttoken`

4928 `\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@futurenonpacelet`

```
4929 \def\LWR@futurenonpacelet#1{\def\LWR@cs{#1}%
4930 \afterassignment\LWR@fnslone\let\nexttoken= }
4931 \def\LWR@fnslone{\expandafter\futurelet\LWR@cs\LWR@fnsltwo}
4932 \def\LWR@fnsltwo{%
4933 \expandafter\ifx\LWR@cs\@sptoken\let\next=\LWR@fnslthree%
4934 \else\let\next=\nexttoken\fi\next}
4935 \def\LWR@fnslthree{\afterassignment\LWR@fnslone\let\next= }
```

`\LWR@getmynexttoken` Looks ahead and copies the next token into `\LWR@mynexttoken`.

```
4936 \newcommand*{\LWR@getmynexttoken}{%
4937 \LWR@traceinfo{\LWR@getmynexttoken}%
4938 % nothing must follow this next line
4939 \LWR@futurenonpacelet\LWR@mynexttoken\LWR@tabledatacolumnstag
4940 }
```

### 59.3 Booleans

Bool `LWR@startedrow` True if should print a row tag before this column.

```
4941 \newbool{LWR@startedrow}
4942 \boolfalse{LWR@startedrow}
```

Bool `LWR@tabularcelladded` True if have added a data cell for this position.

```
4943 \newbool{LWR@tabularcelladded}
4944 \boolfalse{LWR@tabularcelladded}
```

Bool `LWR@doinghline` True if the next row will have an hline or midrule above it. Also used for `\midrule`.

```
4945 \newbool{LWR@doinghline}
4946 \boolfalse{LWR@doinghline}
```

Bool `LWR@doingtbrule` True if the next row will have a top/bottom rule above it.

```
4947 \newbool{LWR@doingtbrule}
4948 \boolfalse{LWR@doingtbrule}
```

Bool `LWR@doingcmidrule` True if the next row will have a cmidrule above it.

This is used by `\LWR@tabularfinishrow` to force a final empty row to create the border for the `\cmidrule`.

```
4949 \newbool{LWR@doingcmidrule}
4950 \boolfalse{LWR@doingcmidrule}
```

Bool LWR@tableparcell True if are handling a paragraph inside a table cell, so must close the paragraph tag before moving on.

```
4951 \newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty \multirow cell, and thus there is no data tag to close.

```
4952 \newbool{LWR@skippingmrowcell}
```

Bool LWR@skippingmcolrowcell True if are doing an empty \multicolumnrow cell, and thus there is no data tag to close, and do not print @ and ! columns.

```
4953 \newbool{LWR@skippingmcolrowcell}
```

Bool LWR@skipatbang True if just finished a \multicolumn so should not create the trailing @ or ! columns table data cells.

```
4954 \newbool{LWR@skipatbang}
```

Bool LWR@emptyatbang True if finishing a row and should print empty @ or ! column table data cells.

```
4955 \newbool{LWR@emptyatbang}
```

Bool LWR@intabularmetadata True if are in a tabular but not in a data cell. Used to prevent extra HTML breaks if not inside table data.

```
4956 \newbool{LWR@intabularmetadata}
4957 \boolfalse{LWR@intabularmetadata}
```

## 59.4 Handling &, @, !, and bar

For technical discussion regarding problems redefining \&, See:

<http://tex.stackexchange.com/questions/11638/where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860>

\LWR@insertatbangcols

```
4958 \newcommand*{\LWR@insertatbangcols}{%
4959 \ifbool{LWR@skipatbang}%
4960 {}%
4961 {%
4962     \LWR@printatbang{at}{\arabic{LWR@tablecolindex}}%
```

```

4963      \LWR@printatbang{bang}{\arabic{LWR@tablecolindex}}%
4964 }%
4965 }

```

`\LWR@closetabledatacell` If `LWR@skippingmrowcell` or `LWR@skippingmcolrowcell` then there is no data tag to close. Otherwise, close any paragraphs, then close the data tag.

```

4966 \newcommand*{\LWR@closetabledatacell}{%
4967 \global\booltrue{LWR@intabularmetadata}%
4968 \ifbool{LWR@exitingtabular}{}%
4969 {% not exiting tabular
4970      \ifboolexpr{bool{LWR@skippingmrowcell} or bool{LWR@skippingmcolrowcell}}%
4971      {%

```

If not skipping a `\multicolumnrow` cell, insert the @ and ! columns after this non-existent column.

```

4972      \ifbool{LWR@skippingmcolrowcell}%
4973      {%
4974      {\LWR@insertatbangcols}%
4975      }%
4976      {% not skippingmrowcell

```

Insert any < then any @ and ! column contents, unless muted for the `\bottomrule` or a `\multicolumn`:

```

4977      \unskip%
4978      \ifboolexpr{%
4979          bool{LWR@tabularmutemods} or
4980          bool{LWR@skipatbang} or
4981          bool{LWR@emptyatbang}
4982      }%
4983      {%
4984      {\LWR@getexparray{LWR@colafterspec}{\arabic{LWR@tablecolindex}}}%

```

Close paragraphs:

```

4985      \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
4986      \global\boolfalse{LWR@tableparcell}%

```

Close the table data cell. Skip the @ and ! cells if are closing a multicolumn cell.

```

4987      \leavevmode\unskip\LWR@htmltag{/td}\LWR@orignewline%
4988      \global\booltrue{LWR@tabularcelladded}%
4989      \LWR@insertatbangcols%
4990      }% not skipping mrowcell
4991      }% not exiting tabular
4992      \global\boolfalse{LWR@skippingmrowcell}%

```

```

4993 \global\boolfalse{LWR@skippingmcolrowcell}%
4994 \global\boolfalse{LWR@skipatbang}%
4995 }

```

LWR@tabulardepth tracks whether & is being used inside a tabular.

```

4996 \newcounter{LWR@tabulardepth}
4997 \setcounter{LWR@tabulardepth}{0}
4998

```

When not used inside a tabular, & performs its original function as recorded here ( with catcode 4 ).

```

4999 \let\LWR@origampmacro&

5000 \end{warpHTML}

```

### 59.4.1 Localizing & catcodes

**for HTML & PRINT:** 5001 \begin{warpall}

 misplaced alignment  
tab character &

Place \StartDefiningTabulars and \EndDefiningTabulars before and after defining macros or environments which include the tabular & character in their definitions.

The catcode of & must be changed before the definitions begin, and must be restored afterwards. Doing so avoids the error

misplaced alignment tab character &

\StartDefiningTabulars Place before defining something with & in it.

```

5002 \newcommand{\StartDefiningTabulars}{%
5003 \LWR@traceinfo{StartDefiningTabulars}
5004 \warpHTMLonly{\catcode'\&=\active}%
5005 }

```

\EndDefiningTabulars Place after defining something with & in it.

```

5006 \newcommand{\EndDefiningTabulars}{%
5007 \LWR@traceinfo{EndDefiningTabulars}
5008 \warpHTMLonly{\catcode'\&=4}%
5009 }

```

```

5010 \end{warpall}

```



### 59.4.2 Handling &

for HTML output: 5011 \begin{warpHTML}

& Will behave depending on whether it is being used inside tabular.

& is redefined to test whether it is inside a tabular environment, in which case it performs special processing for HTML conversion. If not, it behaves normally.

```
5012 \newcommand*{\LWR@tabularampersand}{%
5013 \LWR@traceinfo{\LWR@tabularampersand}%
5014 \ifthenelse{\cnttest{\value{\LWR@tabulardepth}}{>}{0}}{%
5015 {%
```

If not skipping a multirow cell, close the current data cell.

```
5016 \unskip%
5017 \LWR@closetabledatacell%
```

Move to the next column.

```
5018 \addtocounter{\LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
5019 \boolfalse{\LWR@tabularcelladded}%
```

Look at the next token to decide multi or single column data tag.

```
5020 \LWR@getmynexttoken%
5021 }%
```

If not inside a tabular, performs the original action:

```
5022 {\LWR@origampmacro}%
5023 }
```

& is left with its original catcode for now.

tikz package seems to require & be left alone until after tikz has been loaded. Also, cleveref uses the ampersand in one of its options.

& is made active inside a tabular.

& is left alone when in math alignments.

### 59.4.3 Filling an unfinished row

`\LWR@tabularfinishrow` Adds empty table cells if necessary to finish the row.

At the end of the table, if any bottom rules are requested then an empty row must be generated to form the borders which show the rules.

```
5024 \newcommand*{\LWR@tabularfinishrow}{%
```

If not exiting the tabular, or doing a rule, or have already started a row, finish this row:

```
5025 \ifbool{expr}{%
5026     not bool {LWR@exitingtabular} or%
5027     bool{LWR@doingtbrule} or%
5028     bool{LWR@doingcmidrule} or%
5029     bool{LWR@doinghline} or%
5030     bool{LWR@startedrow}%
5031 }{%
```

To locally temporarily turn off `LWR@exitingtabular` so that table data tags will still be generated:

```
5032 \begingroup%
```

If generating a final row for the `\bottomrule` borders, turn off the `@`, `!`, `<`, and `>` column output:

```
5033 \ifbool{LWR@exitingtabular}{%
5034     \booltrue{LWR@tabularmutemods}%
5035 }{}%
```

Reenable the table data tags until finished with the final row:

```
5036 \global\boolfalse{LWR@exitingtabular}%
```

Generate table data tags and ampersands until the right edge:

```
5037 \whiledo{%
5038     \(%
5039         \boolean{LWR@intabularmetadata} \AND%
5040         \NOT\boolean{LWR@tabularcelladded} \AND%
5041         \cnttest{\value{LWR@tablecolindex}}{=}{\value{LWR@tabletotalcols}}%
5042     \) \OR%
5043     \cnttest{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}%
5044 }{}%
5045     \LWR@tabledatasinglecolumntag%
```

The following is essentially `\LWR@tabularampersand` with `LWR@emptyatbang` added to empty the following cells:

```
5046 \LWR@closetabledatacell%
5047 \addtocounter{LWR@tablecolindex}{1}%
5048 \boolfalse{LWR@tabularcelladded}%
5049 \global\booltrue{LWR@emptyatbang}%
```

Starts the next cell:

```
5050 \ifthenelse{%
5051   \cnttest{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}%
5052 }%
5053 {\LWR@getmynexttoken}%
5054 {}%
5055 }%
```

Reenable the original `LWR@exitingtabular` to close the entire table:

```
5056 \endgroup%
5057 \global\boolfalse{LWR@emptyatbang}%
5058 }{}% ifboolexpr
5059 }
```

## 59.5 Handling `\\`

Inside `tabular`, `\\` is redefined to `\LWR@tabularendoffline`

Throws away options `\\[dim]` or `\\*`

`\LWR@tabularendoffline`

```
5060 \NewDocumentCommand{\LWR@tabularendoffline}{s o}
5061 {%
5062 \ifthenelse{\value{LWR@tablecolindex}<\value{LWR@tabletotalcols}}{%
5063   \LWR@tabularfinishrow%
5064 }%
5065 {%
5066   \LWR@closetabledatacell%
5067 }%
5068 % \begin{macrocode}
5069 \LWR@htmltag{tr}\LWR@orignewline
5070 \global\booltrue{LWR@intabularmetadata}
```

Not yet started a table row:

```
5071 \global\boolfalse{LWR@startedrow}
```

Additional setup:

```
5072 \global\boolfalse{LWR@doinghline}%
5073 \global\boolfalse{LWR@doingtbrule}%
5074 \global\boolfalse{LWR@doingcmidrule}%
5075 \LWR@clearmidrules%
```

Start at first column:

```
5076 \setcounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
5077 \boolfalse{LWR@tabularcelladded}%
```

Look at the next token to decide between single column data tag or a special case:

```
5078 \LWR@getmynexttoken%
5079 }
```

## 59.6 Variables

```
5080 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
5081 \newcommand*{\LWR@pposition}{}
5082 \newcommand*{\LWR@pleft}{}
5083 \newcommand*{\LWR@pright}{}%
```

**\LWR@tablecolspec** Holds the parsed column specification, of total width `LWR@tabletotalcols`, not counting @ and ! columns.

Will contain a string such as `llrrccpc`, exactly one letter per  $\LaTeX$  table column, without @, !, >, <, or the vertical bar.

```
5084 \newcommand*{\LWR@tablecolspec}{}%
```

**\LWR@strresult** Holds the result of Str functions.

```
5085 \newcommand*{\LWR@strresult}{}
5086 \newcommand*{\LWR@strresulttwo}{}%
```

**\LWR@origcolspec** Holds the original column specs given to tabular.

```
5087 \newcommand*{\LWR@origcolspec}{}%
```

Ctr LWR@tablecolspecwidth	<p>Holds the number of tokens in the table columns specification.</p> <p>This includes one for each @, !, &lt;, &gt; column, and also one for each of the parameters of p, @, !, &lt;, &gt; columns, and three for each D column.</p> <p>(This is not the total # of <math>\text{\LaTeX}</math> columns in the table.)</p> <p>5088 \newcounter{LWR@tablecolspecwidth}</p>
Ctr LWR@tablecolspecindex	<p>While parsing the <math>\text{\LaTeX}</math> table column specification, starts at 1 and is incremented per token of the specification. While producing the table, resets to 1 at the start of the table and also at each end of line, and is incremented by 1 by each ampersand.</p> <p>5089 \newcounter{LWR@tablecolspecindex}</p>
Ctr LWR@tablecolindex	<p>While parsing the <math>\text{\LaTeX}</math> table column specification, starts at 1 and is incremented per token of the specification. While producing the table, resets to 1 at the start of the table and also at each end of line, and is incremented by 1 by each ampersand.</p> <p>5090 \newcounter{LWR@tablecolindex}</p>
Ctr LWR@tabletotalcols	<p>While parsing a table column specification, begins at 0 and increments by 1 per <math>\text{\LaTeX}</math> table column. Eventually holds the final number of <math>\text{\LaTeX}</math> table columns in each row, not counting @ and ! columns. (In HTML, @ and ! cells become their own columns, but are not included in LWR@tabletotalcols.)</p> <p>5091 \newcounter{LWR@tabletotalcols}</p>
Ctr LWR@tabletotalcolsnext	<p>Holds the next <math>\text{\LaTeX}</math> table column index while parsing, equal to one more than LWR@tabletotalcols.</p> <p>5092 \newcounter{LWR@tabletotalcolsnext}</p>
LWR@colatspec	<p>A data array of specifications for @ columns. The leftmost's index is leftedge, the others are counter values. See section 33.</p>
LWR@colbangspec	<p>A data array of specifications for ! columns. The leftmost's index is leftedge, the others are counter values. See section 33.</p>
LWR@colbeforespec	<p>A data array of specifications for &gt; columns.</p>
LWR@colafterspec	<p>A data array of specifications for &lt; columns.</p>
LWR@colbarspec	<p>A data array of specifications for vertical rules.</p>

## 59.7 Parsing @, >, <, !, bar columns

Holds the parsed argument for @, >, <, or ! columns:

```
5093 \newcommand*{\LWR@colparameter}{}
```

`\LWR@parseatcolumn` Handles @{text} columns.

```
5094 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
5095 \LWR@traceinfo{at column}%
5096 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token into `\LWR@colparameter`, expanding once:

```
5097 \LWR@traceinfo{about to read the next token:}%
5098 \expandarg%
5099 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
5100 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
5101 \LWR@traceinfo{have now read the next token}%
5102 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}{
5103 {% left edge of the table:
5104   \LWR@traceinfo{at the left edge}%
5105   \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@colparameter}%
5106   \LWR@traceinfo{at the left edge: %
5107   \LWR@getexparray{LWR@colatspec}{leftedge}}%
5108 }%
5109 {% not at the left edge:
5110   \LWR@traceinfo{not at the left edge}%
5111   \LWR@setexparray{LWR@colatspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
5112   \LWR@traceinfo{at \arabic{LWR@tabletotalcols}: %
5113   \LWR@getexparray{LWR@colatspec}{\arabic{LWR@tabletotalcols}}}%
5114 }%
5115 \let\LWR@colparameter\relax%
5116 \booltrue{LWR@validtablecol}%
5117 }
```

`\LWR@parsebangcolumn` Handles !{text} columns.

```
5118 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
5119 \LWR@traceinfo{bang column}%
5120 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token into \LWR@colparameter, expanding once:

```
5121 \LWR@traceinfo{about to read the next token:}%
5122 \expandarg%
5123 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
5124 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
5125 \LWR@traceinfo{have now read the next token}%
5126 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}{
5127 {% left edge of the table:
5128     \LWR@traceinfo{at the left edge}%
5129     \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@colparameter}%
5130 }%
5131 {% not at the left edge:
5132     \LWR@traceinfo{not at the left edge}%
5133     \LWR@setexparray{LWR@colbangspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
5134     \LWR@traceinfo{bang \arabic{LWR@tabletotalcols}: \LWR@colparameter!}%
5135 }%
5136 \let\LWR@colparameter\relax%
5137 \booltrue{LWR@validtablecol}%
5138 }
```

\LWR@parsebeforecolumn Handles >{text} columns.

```
5139 \newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
5140 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into \LWR@colparameter:

```
5141 \expandarg%
5142 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
5143 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
5144 \LWR@setexparray{LWR@colbeforespec}{\arabic{LWR@tabletotalcolsnext}}{\LWR@colparameter}%
5145 \let\LWR@colparameter\relax%
```

```
5146 \booltrue{LWR@validtablecol}%
5147 }
```

`\LWR@parseaftercolumn` Handles `<{text}` columns.

```
5148 \newcommand*{\LWR@parseaftercolumn}{%
```

Move to the next token after the '`<`':

```
5149 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into `\LWR@colparameter`:

```
5150 \expandarg%
5151 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
5152 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
5153 \LWR@setexparray{LWR@colafterspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
5154 \let\LWR@colparameter\relax%
5155 \booltrue{LWR@validtablecol}%
5156 }
```

`\LWR@parsebarcolumn` Handles vertical rules.

```
5157 \newcommand*{\LWR@parsebarcolumn}{%
5158 \LWR@traceinfo{bar column}%
```

Remember the bar at this position:

```
5159 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}%
5160 {% left edge of the table:
5161   \LWR@setexparray{LWR@colbarspec}{leftedge}{tvertbarl}%
5162 }%
5163 {% not at the left edge:
5164   \LWR@setexparray{LWR@colbarspec}{\arabic{LWR@tabletotalcols}}{tvertbarr}%
5165 }%
5166 \booltrue{LWR@validtablecol}%
5167 }
```

## 59.8 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.

```

5168 \newcommand*{\LWR@parsenormalcolumn}[1]{%
5169 \appto\LWR@tablecolspec{#1}%
5170 \addtocounter{\LWR@tabletotalcols}{1}%
5171 \addtocounter{\LWR@tabletotalcolsnext}{1}%
5172 \LWR@traceinfo{normal column \arabic{\LWR@tabletotalcols}: #1}%
5173 \LWR@setexparray{\LWR@colatspec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5174 \LWR@setexparray{\LWR@colbangspec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5175 \LWR@setexparray{\LWR@colbeforespec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5176 \LWR@setexparray{\LWR@colafterspec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5177 \LWR@setexparray{\LWR@colbarspec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5178 \booltrue{\LWR@validtablecol}%
5179 }
```

## 59.9 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn`  $\{ \langle thiscolumn \rangle \}$  The width will be ignored.

```

5180 \newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type:

```

5181 \LWR@parsenormalcolumn{#1}%
```

Skips the following width token:

```

5182 \addtocounter{\LWR@tablecolspecindex}{1}%
5183 }
```

## 59.10 Parsing ‘D’ columns

From the dcolumn package.

`\LWR@parseDcolumn`  $\{ \langle thiscolumn \rangle \}$  The three parameters will be ignored.

```

5184 \newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```

5185 \LWR@parsenormalcolumn{#1}%
```

Table 8: Tabular baseline

l	p	m	b	r
			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

Skips the following three parameters.

```
5186 \addtocounter{LWR@tablecolspecindex}{3}%
5187 }
```

59.11 Parsing the column specifications


 HTML CSS cannot exactly match the  $\text{\LaTeX}$  concept of a baseline for a table row. Table 8 shows the  $\text{\LaTeX}$  results for various vertical-alignment choices, with the baseline of the first column drawn across all the columns for comparison. See the p column specification in table 9 for details.

Table 9 describes how each kind of column is converted to HTML.

Bool LWR@validtablecol True if found a valid table column type.

```
5188 \newbool{LWR@validtablecol}
```

$\backslash$ LWR@parsetablecols  $\{\langle colspecs \rangle\}$

Scans the column specification left to right.

Builds  $\backslash$ LWR@tablecolspec with the final specification, one column per entry. The final number of cells in each row is stored in LWR@tabletotalcols.

```
5189 \newcommand*{\LWR@parsetablecols}[1]{%
5190 \LWR@traceinfo{LWR@parsetablecols started}%
```

Remember the original supplied column spec:

```
5191 \renewcommand*{\LWR@origcolspec}{#1}%
```

Table 9: 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 8, $\text{\LaTeX}$ 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> :</b>	Vertical rule.
<b>Unknown:</b>	Converted to 'l'.
<b>\newcolumn:</b>	Currently treated as unknown.

---

Remove spaces:

```
5192 \expandarg%
5193 \StrSubstitute{\LWR@origcolspec}{ }{}[\LWR@origcolspec]%
```

Clear the parsed resulting column spec:

```
5194 \renewcommand*\LWR@tablecolspec{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
5195 \setcounter{LWR@tabletotalcols}{0}%
5196 \setcounter{LWR@tabletotalcolsnext}{1}%
5197 \LWR@setexparray{LWR@colatspec}{leftedge}{}%
5198 \LWR@setexparray{LWR@colatspec}{1}{}%
5199 \LWR@setexparray{LWR@colatspec}{2}{}%
5200 \LWR@setexparray{LWR@colatspec}{3}{}%
5201 \LWR@setexparray{LWR@colbangspec}{leftedge}{}%
5202 \LWR@setexparray{LWR@colbangspec}{1}{}%
5203 \LWR@setexparray{LWR@colbangspec}{2}{}%
5204 \LWR@setexparray{LWR@colbangspec}{3}{}%
5205 \LWR@setexparray{LWR@colbeforespec}{1}{}%
5206 \LWR@setexparray{LWR@colbeforespec}{2}{}%
5207 \LWR@setexparray{LWR@colbeforespec}{3}{}%
5208 \LWR@setexparray{LWR@colafterspec}{1}{}%
5209 \LWR@setexparray{LWR@colafterspec}{2}{}%
5210 \LWR@setexparray{LWR@colafterspec}{3}{}%
5211 \LWR@setexparray{LWR@colbarspec}{leftedge}{}%
5212 \LWR@setexparray{LWR@colbarspec}{1}{}%
5213 \LWR@setexparray{LWR@colbarspec}{2}{}%
5214 \LWR@setexparray{LWR@colbarspec}{3}{}%
```

Starting at the first column specification:

```
5215 \setcounter{LWR@tablecolspecindex}{1}%
```

Place the colspecs string length into \LWR@strresult, and remember the number of characters in the column specification:

```
5216 \LWR@traceinfo{about to StrLen}%
5217 \expandarg%
5218 \StrLen{\LWR@origcolspec}[\LWR@strresult]%
5219 \fullexpandarg%
5220 \LWR@traceinfo{finished StrLen}%
5221 \setcounter{LWR@tablecolspecwidth}{\LWR@strresult}%
```

Scan through the column specifications:

```
5222 \whiledo{\not\value{LWR@tablecolspecindex}>\value{LWR@tablecolspecwidth}}{%
```

Place the next single-character column type into \LWR@strresult:

```
5223 \expandarg%
5224 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@strresult]%
5225 \LWR@traceinfo{position \arabic{LWR@tablecolspecindex}: \LWR@strresult}%
5226 \fullexpandarg%
```

Not yet found a valid column type

```
5227 \boolfalse{LWR@validtablecol}%
```



Note that the parameter for a p{spec} column is a token list which will NOT match l,c,r,p.

```
5228 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{}%
5229 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{}%
5230 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{}%
5231 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{}%
5232 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{}%
5233 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{}%
5234 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{}%
5235 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{}%
5236 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
5237 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
5238 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
5239 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
5240 \IfStrEq{\LWR@strresult}{|}{\LWR@parsebarcolumn}{}%
5241 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
5242 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
5243 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%
```

From the dcolumn package:

```
5244 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{}%
```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```
5245 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%
```

---

Many people define centered versions “P”, “M”, and “B”:

```
\newcolumnntype{P}[1]{>\centering\arraybackslash}p{#1}}
```

---

```
5246 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
```

```

5247 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
5248 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%

```

If this column was an invalid column type, convert it to an l column:

```

5249 \ifbool{\LWR@validtablecol}{}{}%
5250     \LWR@traceinfo{invalid column type: \LWR@strresult}%
5251     \LWR@parsenormalcolumn{l}%
5252 }%
5253 \addtocounter{\LWR@tablecolspecindex}{1}%
5254 }% whiledo
5255 }%

```

## 59.12 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.

```

5256 \newcommand*{\LWR@maybenewtablerow}
5257 {%
5258 \ifbool{\LWR@startedrow}%
5259 {}% started the row
5260 {}% not started the row

```

Remember that now have started the row:

```

5261     \global\booltrue{\LWR@startedrow}%

```

Create the row tag, with a class if necessary.

```

5262     \global\booltrue{\LWR@intabularmetadata}%
5263     \ifbool{\LWR@doinghline}%
5264     {%
5265         \LWR@htmltag{tr class="hline" }%
5266         \LWR@orignewline%
5267     }%
5268     {% not doing hline
5269         \ifbool{\LWR@doingtbrule}%
5270         {%
5271             \LWR@htmltag{tr class="tbrule"%}
5272             \LWR@orignewline%
5273         }%
5274         {\LWR@htmltag{tr}\LWR@orignewline}%
5275     }% end of not doing hline
5276 }% end of not started the row
5277 }

```

### 59.13 Printing vertical bar tags

`\LWR@printbartag`  $\{\langle index \rangle\}$

Adds to a tabular data cell an HTML class name for a left/right vertical bar.

```

5278 \newcommand*\LWR@printbartag}[1]{%
5279 \ifboolexpr{bool{\LWR@tabularmutemods} or bool{\LWR@emptyatbang}}%
5280 {}% muting or empty
5281 {}% not muting
5282     \edef\LWR@tempone{\LWR@getexparray{\LWR@colbarspec}{#1}}%
5283     \ifdefempty{\LWR@tempone}{ }\LWR@tempone}%
5284 }% not muting
5285 }
```

### 59.14 Printing at or bang tags

`\LWR@printatbang`  $\{\langle at-or-bang \rangle\} \{\langle index \rangle\}$

```

5286 \newcommand*\LWR@printatbang}[2]{%
```

Fetch the column at or bang spec:

```

5287 \edef\LWR@atbangspec{\LWR@getexparray{\LWR@col#1spec}{#2}}%
5288 \LWR@traceinfo{atbang: #2 !\LWR@atbangspec!}%
```

Only generate if is not empty;

```

5289 \ifdefempty{\LWR@atbangspec}%
5290 {}%
5291 {}% not empty
5292     \LWR@htmltag{%
5293         td class="td#1%
5294         \LWR@subaddcmidruletrim{}{}%
5295         \LWR@printbartag{#2}%
5296         "%
5297         \LWR@tdstartstyles%
5298         \LWR@addcmidrulewidth%
5299         \LWR@tdendstyles%
5300     }%
```

Create an empty cell if muting for the `\bottomrule`:

```

5301     \ifboolexpr{bool{\LWR@tabularmutemods} or bool{\LWR@emptyatbang}}%
5302     {}%
```

```

5303     {\LWR@atbangspec}%
5304 %
5305     \LWR@htmltag{/td}\LWR@orignewline%
5306     \global\booltrue{\LWR@tabularcelladded}%
5307 }% not empty
5308 }%

```

`\LWR@addleftmostbartag`

```

5309 \newcommand*{\LWR@addleftmostbartag}{%
5310 \ifnumcomp{\value{\LWR@tablecolindex}}{=}{1}{%
5311     \LWR@printbartag{leftedge}%
5312 }{}%
5313 }

```

`\LWR@tabularleftedge`

```

5314 \newcommand*{\LWR@tabularleftedge}{%
5315 \ifthenelse{\cnttest{\value{\LWR@tablecolindex}}=1}{%
5316 {%
5317     \LWR@printatbang{at}{leftedge}%
5318     \LWR@printatbang{bang}{leftedge}%
5319 }% left edge
5320 {}% not left edge
5321 }

```

## 59.15 Data opening tag

`\LWR@tabledatasinglecolumnntag` Print a table data opening tag with style for alignment

```

5322 \newcommand*{\LWR@tabledatasinglecolumnntag}%
5323 {%
5324 \LWR@maybenewtablerow%

```

Don't start a new paragraph tag if have already started one:

```

5325 \ifbool{\LWR@intabularmetadata}%
5326 {%

```

If have found the end of tabular command, do not create the next data cell:

```

5327     \ifbool{\LWR@exitingtabular}{}%
5328     {% not exiting tabular

```



Print the @ and ! contents before first column:

```
5329      \LWR@tabularleftedge%
```

Fetch the current column's alignment character into \LWR@strresult:

```
5330      \StrChar{\LWR@tablecolspec}{\arabic{\LWR@tablecolindex}}[\LWR@strresult]%
```

print the start of a new table data cell:

```
5331      \LWR@htmltag{td class="td%
```

append this column's spec:

```
5332      \LWR@strresult%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag. Also add vertical bar tags.

```
5333      \LWR@addcmidruletrim%
5334      \LWR@addleftmostbartag%
5335      \LWR@printbartag{\arabic{\LWR@tablecolindex}}%
5336      "%
```

```
5337      \LWR@tdstartstyles%
5338      \LWR@addcmidrulewidth%
5339      \LWR@addformatwpalignment%
5340      \LWR@tdendstyles%
5341      }%
```

If this is a p, m, b, or X column, allow paragraphs:

```
5342      \ifthenelse{%
5343          \equal{\LWR@strresult}{p}\OR%
5344          \equal{\LWR@strresult}{m}\OR%
5345          \equal{\LWR@strresult}{b}\OR%
5346          \equal{\LWR@strresult}{P}\OR%
5347          \equal{\LWR@strresult}{M}\OR%
5348          \equal{\LWR@strresult}{B}\OR%
5349          \equal{\LWR@strresult}{X}}%
5350      }%
5351      {% allow pars
5352          \LWR@startpars%
5353          \global\booltrue{\LWR@tableparcell}}%
5354      }% allow pars
5355      }% no pars
```

Print the > contents unless muted for the \bottomrule:

```

5356      \ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
5357      {}%
5358      {%
5359          \LWR@getexparray{LWR@colbeforespec}{\arabic{LWR@tablecolindex}}%
5360      }%
5361      \global\boolfalse{LWR@intabularmetadata}%
5362  }% not exiting tabular
5363 }{}% in tabular metadata
5364 }%
```

## 59.16 Midrules

**LWR@midrules** LWR@midrules is a data array (section 33) of columns each containing a non-zero width if a midrule should be created for this column.

**LWR@trimlrules** LWR@trimlrules is a data array (section 33) of columns containing 1 if a midrule should be left trimmed for each column.

**LWR@trimrrules** LWR@trimrrules is a data array (section 33) of columns containing r if a midrule should be right trimmed for each column.

**Ctr LWR@midrulecounter** Indexes across the LWR@midrules and LWR@trim<l/r>rules data arrays.

```
5365 \newcounter{LWR@midrulecounter}
```

**Len \LWR@heavyrulewidth** The default width of the rule.

```
5366 \newlength{\LWR@heavyrulewidth}
5367 \setlength{\LWR@heavyrulewidth}{.08em}
```

**Len \LWR@lightrulewidth** The default width of the rule.

```
5368 \newlength{\LWR@lightrulewidth}
5369 \setlength{\LWR@lightrulewidth}{.05em}
```

**Len \LWR@cmidrulewidth** The default width of the rule.

```
5370 \newlength{\LWR@cmidrulewidth}
5371 \setlength{\LWR@cmidrulewidth}{.03em}
```

**Len \LWR@thiscmidrulewidth** The width of the next rule, defaulting to \LWR@cmidrulewidth.

If not \LWR@cmidrulewidth, a style will be used to generate the custom width.

Assigned from the LWR@midrules array.

```
5372 \newlength{\LWR@thiscmidrulewidth}
5373 \setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}
```

**\LWR@clearmidrules** Start new midrules. Called at beginning of tabular and also at \.

Clears all LWR@midrules and LWR@trimrules markers for this line.

```
5374 \newcommand*{\LWR@clearmidrules}
5375 {%
5376 \setcounter{LWR@midrulecounter}{1}%
5377 \whiledo{%
5378 \cnttest{\value{LWR@midrulecounter}}{<=}{\value{LWR@tablecolspecwidth}}%
5379 }%
5380 {%
5381 \LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{0pt}%
5382 \setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
5383 \LWR@setexparray{LWR@trimlrules}{\arabic{LWR@midrulecounter}}{}%
5384 \LWR@setexparray{LWR@trimrrules}{\arabic{LWR@midrulecounter}}{}%
5385 \addtocounter{LWR@midrulecounter}{1}%
5386 }%
5387 }
```

**\LWR@subcmidrule** [*width*] [*trim*] [*leftcolumn*] [*rightcolumn*]

Marks LWR@midrules data array elements to be non-zero widths from left to right columns. Also marks trimming for the L and/or R columns.

LWR@doingcmidrule is set to force an empty row at the end of the tabular to create the rule.

```
5388 \newcommand*{\LWR@subcmidrule}[4]{%
5389 \setcounter{LWR@midrulecounter}{#3}%
5390 \whiledo{\cnttest{\value{LWR@midrulecounter}}{<=}{#4}}%
5391 {%
5392 \LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{#1}%
5393 \addtocounter{LWR@midrulecounter}{1}%
5394 }% \whiledo
5395 \IfSubStr{#2}{l}{\LWR@setexparray{LWR@trimlrules}{#3}{l}}{}%
5396 \IfSubStr{#2}{r}{\LWR@setexparray{LWR@trimrrules}{#4}{r}}{}%
5397 \booltrue{LWR@doingcmidrule}%
5398 }
```

**\LWR@docmidrule** [*width*] [*trim*] [*leftcolumn-rightcolumn*]

Marks LWR@midrules array elements to be “Y” from left to right columns. Also marks trimming for the L and/or R columns.

```
5399 \NewDocumentCommand{\LWR@docmidrule}{0{\LWR@cmidrulewidth} D(){ } >{\SplitArgument{1}{-}}m}%
5400 {\LWR@subcmidrule{#1}{#2}#3}
```

Used to compute margins, tabular trims:

```
5401 \newlength{\LWR@templengthone}%
5402 \newlength{\LWR@templengthtwo}%
```

Used to add a style to a table data cell:

```
5403 \newboolean{\LWR@tdhavecellstyle}
```

`\LWR@tdstartstyles` Begins possibly adding a table data cell style.

```
5404 \newcommand*{\LWR@tdstartstyles}{\global\boolfalse{\LWR@tdhavecellstyle}}
```

`\LWR@tdaddstyle` Starts adding a table data cell style.

```
5405 \newcommand*{\LWR@tdaddstyle}{%
5406 \ifbool{\LWR@tdhavecellstyle}%
5407 {; }%
5408 { style="%
5409 \booltrue{\LWR@tdhavecellstyle}%
5410 }
```

`\LWR@tdendstyles` Finishes possibly adding a table data cell style. Prints the closing quote.

```
5411 \newcommand*{\LWR@tdendstyles}{%
5412 \ifbool{\LWR@tdhavecellstyle}{%
5413     "%
5414     \global\boolfalse{\LWR@tdhavecellstyle}%
5415 }{}%
5416 }
```

`\LWR@subaddcmidruletrim` `{<lefttrim>} {<righttrim>}` Adds a `\cmidrule` with optional trim.

```
5417 \newcommand*{\LWR@subaddcmidruletrim}[2]{%
5418 \setlength{\LWR@templengthone}{\LWR@getexparray{\LWR@midrules}{\arabic{\LWR@tablecolindex}}}%
5419 \ifdimcomp{\LWR@templengthone}{>}{0pt}%
5420 {%
```

Print the class without left and right trim letters appended:

```
5421 \LWR@origtilde tdrule#1#2%
```

Remember the width of the rule:

```
5422 \setlength{\LWR@thiscmidrulewidth}{\LWR@templengthone}%
5423 }%
5424 {%
5425 \setlength{\LWR@thiscmidrulewidth}{0pt}%
5426 }%
5427 }
```

`\LWR@addcmidruletrim` Adds left or right trim to a `\cmidrule`.

```
5428 \newcommand*{\LWR@addcmidruletrim}{%
5429 \LWR@subaddcmidruletrim%
5430 {\LWR@getexparray{\LWR@trimlrules}{\arabic{\LWR@tablecolindex}}}%
5431 {\LWR@getexparray{\LWR@trimrrules}{\arabic{\LWR@tablecolindex}}}%
5432 }
```

`\LWR@addrulewidth` `{\langle thiswidth \rangle}{\langle defaultwidth \rangle}`

If not default width, add a custom style with width and color depending on `thiswidth`.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
5433 \newcommand{\LWR@addrulewidth}[2]{%
```

Only add a custom width if `thiswidth` is different than the `defaultwidth`:

```
5434 \ifboolexpr{%
5435 test{\ifdimcomp{#1}{=}{Opt}}}%
5436 or ( test{\ifdimcomp{#1}{=}{#2}} and not bool{FormatWP} )%
5437 }%
5438 {}% default width
5439 {}% custom width
```

Ensure that the width is wide enough to display in the browser:

```
5440 \LWR@forceminwidth{#1}%
```

Begin adding a style:

```
5441 \LWR@tdaddstyle%
```

The style itself:

```

5442 \uselengthunit{PT}%
5443 border-top:\rndprintlength{\LWR@atleastonept} solid %

```

The darkness of the color depends on the thickness of the rule:

```

5444 \ifdimcomp{#1}<{\LWR@lightrulewidth}%
5445 {\#AOAOAO}%
5446 {% lightrule or heavier
5447 \ifdimcomp{#1}<{\LWR@heavyrulewidth}%
5448 {\#808080}%
5449 {black}%
5450 }% lightrule or heavier
5451 }% custom width
5452 }

```

`\LWR@addcmidrulewidth` Adds a style for the rule width.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```

5453 \newcommand{\LWR@addcmidrulewidth}{%
5454 \LWR@addrulewidth{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
5455 }

```

`\LWR@WPcell` `{\langle text-align \rangle}{\langle vertical-align \rangle}`

```

5456 \newcommand*{\LWR@WPcell}[2]{%
5457 \LWR@tdaddstyle%
5458 text-align:#1; vertical-align:#2%
5459 }

```

`\LWR@addformatwpalignment` If `FormatWP`, adds a style for the alignment.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```

5460 \newcommand*{\LWR@addformatwpalignment}{%
5461 \ifbool{FormatWP}{%
5462 \StrChar{\LWR@tablecolspec}{\arabic{\LWR@tablecolindex}}[\LWR@strresult]%
5463 \ifdefstring{\LWR@strresult}{l}{\LWR@WPcell{left}{middle}}{%
5464 \ifdefstring{\LWR@strresult}{c}{\LWR@WPcell{center}{middle}}{%
5465 \ifdefstring{\LWR@strresult}{r}{\LWR@WPcell{right}{middle}}{%
5466 \ifdefstring{\LWR@strresult}{p}{\LWR@WPcell{left}{bottom}}{%
5467 \ifdefstring{\LWR@strresult}{m}{\LWR@WPcell{left}{middle}}{%
5468 \ifdefstring{\LWR@strresult}{b}{\LWR@WPcell{left}{top}}{%
5469 \ifdefstring{\LWR@strresult}{P}{\LWR@WPcell{center}{bottom}}{%
5470 \ifdefstring{\LWR@strresult}{M}{\LWR@WPcell{center}{middle}}{%
5471 \ifdefstring{\LWR@strresult}{B}{\LWR@WPcell{center}{top}}{%
5472 }{}%
5473 }

```

## 59.17 Multicolumns

### 59.17.1 Parsing multicolumns

5474 \newcounter{LWR@tablemulticolwidth}

Indexes into the multicolumn specification:

5475 \newcounter{LWR@tablemulticolspos}

Remembers multicolumn vertical rules if found in the column spec.

5476 \newbool{LWR@mccolvertbarl}

5477 \newbool{LWR@mccolvertbarr}

\LWR@printmccoltype  $\{\langle colspec \rangle\}$  Print any valid column type found. Does not print @, !, >, or < columns or their associated tokens.

This is printed as part of the table data tag's class.

5478 \newcommand\*{\LWR@printmccoltype}[1]{%

5479 \LWR@traceinfo{lw@printmccoltype -#1-}%

Get one token of the column spec:

5480 \StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%

Add to the HTML tag depending on which column type is found:

5481 \IfStrEq{\LWR@strresult}{l}{l}{}%

5482 \IfStrEq{\LWR@strresult}{c}{c}{}%

5483 \IfStrEq{\LWR@strresult}{r}{r}{}%

5484 \IfStrEq{\LWR@strresult}{p}{p}{}%

5485 \IfStrEq{\LWR@strresult}{m}{m}{}%

5486 \IfStrEq{\LWR@strresult}{b}{b}{}%

5487 \IfStrEq{\LWR@strresult}{P}{P}{}%

5488 \IfStrEq{\LWR@strresult}{M}{M}{}%

5489 \IfStrEq{\LWR@strresult}{B}{B}{}%

5490 \IfStrEq{\LWR@strresult}{S}{r}{}%

5491 \IfStrEq{\LWR@strresult}{X}{p}{}%

5492 \IfStrEq{\LWR@strresult}{|}{}%

5493 \ifthenelse{\value{LWR@tablemulticolspos}=1}{%

5494 {\booltrue{LWR@mccolvertbarl}}%

5495 {\booltrue{LWR@mccolvertbarr}}%

5496 }{}}%

5497 \LWR@traceinfo{lw@printmccoltype done}%

5498 }

`\LWR@multicolpartext` Print the data with paragraph tags:

```
5499 \newcommand*{\LWR@multicolpartext}{%
5500 \LWR@startpars%
5501 \LWR@multicoltext%
5502 \LWR@stoppars%
5503 }
```

`\LWR@multicolother` `{<colspec>}` For @, !, >, <, print the next token without paragraph tags:

```
5504 \newcommand*{\LWR@multicolother}[1]{%
5505 \addtocounter{\LWR@tablemulticolspos}{1}%
5506 \StrChar{#1}{\arabic{\LWR@tablemulticolspos}}[\LWR@strresult]%
5507 \LWR@strresult%
```

A valid column data type was found:

```
5508 \booltrue{\LWR@validtablecol}%
5509 }
```

`\LWR@multicolskip` Nothing to print for this column type.

```
5510 \newcommand*{\LWR@multicolskip}{%
```

A valid column data type was found:

```
5511 \booltrue{\LWR@validtablecol}%
5512 }
```

`\LWR@printmccoldata` `{<colspec>}` Print the data for any valid column type found.

```
5513 \newcommand*{\LWR@printmccoldata}[1]{%
5514 \LWR@traceinfo{\LWR@printmccoldata -#1}%
```

Not yet found a valid column type:

```
5515 \boolfalse{\LWR@validtablecol}%
```

Get one token of the column spec:

```
5516 \StrChar{#1}{\arabic{\LWR@tablemulticolspos}}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
5517 \IfStrEq{\LWR@strresult}{!}{\LWR@multicoltext}{}%
```



```

5518 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
5519 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
5520 \IfStrEq{\LWR@strresult}{D}{%
5521 \addtocounter{LWR@tablemulticolspos}{3}% skip parameters
5522 \LWR@multicoltext%
5523 }{}%
5524 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolpartext}{}%
5525 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolpartext}{}%
5526 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolpartext}{}%
5527 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolpartext}{}%
5528 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolpartext}{}%
5529 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolpartext}{}%
5530 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolpartext}{}%
5531 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolpartext}{}%
5532 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
5533 \IfStrEq{\LWR@strresult}{\detokenize@}{\LWR@multicolother{#1}}{%
5534 \IfStrEq{\LWR@strresult}{\detokenize!}{\LWR@multicolother{#1}}{%
5535 \IfStrEq{\LWR@strresult}{\detokenize>}{\LWR@multicolother{#1}}{%
5536 \IfStrEq{\LWR@strresult}{\detokenize<}{\LWR@multicolother{#1}}{%

```

If an invalid column type:

```

5537 \ifbool{LWR@validtablecol}{\LWR@multicoltext}%

```

Tracing:

```

5538 \LWR@traceinfo{lwr@printmccoldata done}%
5539 }

```

`\parsemulticolumnalignment`  $\{(1: \textit{colspec})\} \{(2: \textit{printresults})\}$

Scan the multicolumn specification and execute the printfunction for each entry.

Note that the spec for a  $p\{\textit{spec}\}$  column, or @, >, <, is a token list which will NOT match l, c, r, or p.

```

5540 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
5541 \setcounter{LWR@tablemulticolspos}{1}%
5542 \StrLen{#1}[\LWR@strresult]%
5543 \setcounter{LWR@tablemulticolwidth}{\LWR@strresult}%

```

Scan across the tokens in the column spec:

```

5544 \whiledo{%
5545   \NOT\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolwidth}%
5546 }{%
5547 {%

```

Execute the assigned print function for each token in the column spec:

```
5548 #2{#1}%
```

Move to the next token in the column spec:

```
5549 \addtocounter{LWR@tablemulticolspos}{1}%
5550 }%
5551 }
```

### 59.17.2 Multicolumn factored code

```
5552 \newcommand{\LWR@multicoltext}{}%
```

To find multicolumn right trim:

```
5553 \newcounter{LWR@lastmulticolumn}
```

```
\LWR@domulticolumn  [\langle 1: vpos \rangle] [\langle 2: #rows \rangle] {\langle 3: numLaTeXcols \rangle} {\langle 4: numHTMLcols \rangle} {\langle 5: colspec \rangle}
{\langle 6: text \rangle}
```

```
5554 \NewDocumentCommand{\LWR@domulticolumn}{o o m m m +m}{%
5555 \LWR@traceinfo{LWR@domulticolumn -#1- -#2- -#4- -#5-}%
```

Remember the text to be inserted, and remember that a valid column type was found:

```
5556 \renewcommand{\LWR@multicoltext}{%
5557 #6%
5558 \booltrue{LWR@validtablecol}%
5559 }%
```

Compute the rightmost column to be included. This is used to create the right trim.

```
5560 \setcounter{LWR@lastmulticolumn}{\value{LWR@tablecolindex}}%
5561 \addtocounter{LWR@lastmulticolumn}{#3}%
5562 \addtocounter{LWR@lastmulticolumn}{-1}%
```

Row processing:

```
5563 \LWR@maybe newtablerow%
```

Begin the opening table data tag:

```
5564 \LWR@htmltag{td colspan="#4" %
5565 \IfValueT{#2}{% rows?
5566 rowspan="#2" %
```

```

5567 \IfValueT{#1}{% vpos?
5568 \ifstrequal{#1}{b}{style="vertical-align:bottom" }{}%
5569 \ifstrequal{#1}{t}{style="vertical-align:top" }{}%
5570 }% vpos?
5571 }% rows?

```

```

5572 class="td%

```

Print the column type and vertical bars:

```

5573 \boolfalse{LWR@mcolvertbarl}%
5574 \boolfalse{LWR@mcolvertbarr}%
5575 \LWR@parsemulticolumnalignment{#5}{\LWR@printmccoltype}%

```

If this column has a cmidrule, add “rule” to the end of the HTML class tag.

If this position had a “Y” then add “rule” for a horizontal rule:

```

5576 \LWR@subaddcmidruletrim%
5577 {\LWR@getexparray{LWR@trimlrules}{\arabic{LWR@tablecolindex}}}%
5578 {\LWR@getexparray{LWR@trimrrules}{\arabic{LWR@lastmulticolumn}}}%

```

Also add vertical bar class.

```

5579 \ifbool{LWR@mcolvertbarl}{ tvertbarl}{}%
5580 \ifbool{LWR@mcolvertbarr}{ tvertbarr}{}%

```

Close the class tag’s opening quote:

```

5581 "%

```

```

5582 \LWR@tdstartstyles%
5583 \LWR@addcmidrulewidth%
5584 \LWR@addformatwpalignment%
5585 \LWR@tdendstyles%
5586 }% end of the opening table data tag
5587 \global\boolfalse{LWR@intabularmetadata}%
5588 \LWR@parsemulticolumnalignment{#5}{\LWR@printmccoldata}%
5589 }

```

### 59.17.3 Multicolumn

`\LWR@htmlmulticolumn` {<numcols>} {<alignment>} {<text>}

```

5590 \NewDocumentCommand{\LWR@htmlmulticolumn}{m m +m}%
5591 {%

```

Figure out how many extra HTML columns to add for @ and ! columns:

```
5592 \LWR@tabularhtmlcolumns{\arabic{LWR@tablecolindex}}{#1}
```

Create the multicolumn tag:

```
5593 \LWR@domulticolumn{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#3}%
```

Move to the next  $\LaTeX$  column:

```
5594 \addtocounter{LWR@tablecolindex}{#1}%
5595 \addtocounter{LWR@tablecolindex}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
5596 \booltrue{LWR@skipatbang}%
5597 }
```

#### 59.17.4 Longtable captions

longtable captions use \multicolumn.

Bool LWR@starredlongtable Per the caption package, step the counter if longtable\*.

```
5598 \newbool{LWR@starredlongtable}
5599 \boolfalse{LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
5600 \providecommand*{\LTcaptype}{table}
```

\LWR@longtabledatacaptiontag \* [*<toc entry>*] {<caption>}

```
5601 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
5602 {%
```

Remember the latest name for \nameref:

```
5603 \IfValueTF{#2}{% optional given?
5604   \ifthenelse{\equal{#2}{}}{% optional empty?
5605     {\LWR@setlatestname{#3}}% empty
5606     {\LWR@setlatestname{#2}}% given and non-empty
5607 }% optional given
5608 {\LWR@setlatestname{#3}}% no optional
```

Create a multicolumn across all the columns:

Figure out how many extra HTML columns to add for @ and ! columns found between the first and the last column:

```
5609 \LWR@tabularhtmlcolumns{1}{\arabic{LWR@tabletotalcols}}
```

Create the multicolumn tag:

```
5610 \LWR@domulticolumn{\arabic{LWR@tabletotalcols}}{\arabic{LWR@tabhtmlcoltotal}}{P}%
5611 {% \LWR@domulticolumn
5612 \IfBooleanTF{#1}% star?
```

Star version, show a caption but do not make a LOT entry:

```
5613 {% yes star
5614   \LWR@figcaption%
5615   #3%
5616   \endLWR@figcaption%
5617 }%
5618 {% No star:
```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```
5619   \ifbool{LWR@starredlongtable}%
5620   {%
5621   \ifthenelse{\equal{#2}{}}{% TOC entry
5622   }{%
5623   {%
5624     \refstepcounter{LTcaption}%
5625     \protected@edef\@currentlabel{%
5626     \csuse{p@LTcaption}\csuse{theLTcaption}}%
5627   }%
5628   }{%}
```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```
5629   \LWR@figcaption%
5630   \csuse{fnum@LTcaption}\CaptionSeparator#3%
5631   \endLWR@figcaption%
```

See if an optional caption was given:

```
5632   \ifthenelse{\equal{#2}{}}{% TOC entry empty
```

if the optional caption was given, but empty, do not form a TOC entry

5633     {}%

If the optional caption was given, but might only be []:

5634     {% TOC entry not empty  
5635         \IfNoValueTF{#2}% No TOC entry?

The optional caption is []:

5636         {% No TOC entry  
5637             \addcontentsline%  
5638             {\csuse{ext@LTcapttype}}%  
5639             {LTcapttype}%  
5640             {%  
5641                 \protect\numberline%  
5642                 {\csuse{p@LTcapttype}\csuse{theLTcapttype}}%  
5643                 {\ignorespaces #3\protect\relax}%  
5644             }%  
5645         }% end of No TOC entry

The optional caption has text enclosed:

5646         {% yes TOC entry  
5647             \addcontentsline%  
5648             {\csuse{ext@LTcapttype}}%  
5649             {LTcapttype}%  
5650             {%  
5651                 \protect\numberline%  
5652                 {\csuse{p@LTcapttype}\csuse{theLTcapttype}}%  
5653                 {\ignorespaces #2\protect\relax}%  
5654             }%  
5655         }% end of yes TOC entry  
5656     }% end of TOC entry not empty  
5657 }% end of no star

Skip any trailing @ or ! columns for this cell:

5658 \booltrue{LWR@skipatbang}%  
5659 }% end of \LWR@domulticolumn  
5660  
5661 \addtocounter{LWR@tablecolindex}{\arabic{LWR@tabletotalcols}}  
5662 \addtocounter{LWR@tablecolindex}{-1}  
5663  
5664 }

### 59.17.5 Counting HTML tabular columns

The  $\text{\LaTeX}$  specification for a table includes a number of columns separated by the `&` character. These columns differ in content from line to line. Additional virtual columns may be specified by the special `@` and `!` columns. These columns are identical from line to line, but may be skipped during a multicolumn cell.

For HTML output, `@` and `!` columns are placed into their own tabular columns. Thus, a  $\text{\LaTeX}$  `\multicolumn` command may span several additional `@` and `!` columns in HTML output. These additional columns must be added to the total number of columns spanned by an HTML multi-column data cell.

```
5665 \newcounter{LWR@tabhtmlcolindex}
5666 \newcounter{LWR@tabhtmlcolend}
5667 \newcounter{LWR@tabhtmlcoltotal}
```

```
\LWR@subtabularhtmlcolumns {\langle index \rangle}
```

Factored from `\LWR@tabularhtmlcolumns`, which follows.

```
5668 \newcommand*{\LWR@subtabularhtmlcolumns}[1]{%
```

Temporarily define a macro equal to the `@` specification for this column:

```
5669     \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{#1}}%
```

If the `@` specification is not empty, add to the count:

```
5670     \ifdefempty{\LWR@atbangspec}%
5671         {}%
5672         {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the `!` columns:

```
5673     \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{#1}}%
5674     \ifdefempty{\LWR@atbangspec}%
5675         {}%
5676         {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
5677 }
```

```
\LWR@tabularhtmlcolumns {\langle starting  $\text{\LaTeX}$  column \rangle} {\langle number  $\text{\LaTeX}$  columns \rangle}
```

Compute the total number of HTML columns being spanned, considering the starting  $\text{\LaTeX}$  table column and the number of  $\text{\LaTeX}$  tabular columns being spanned. Any `@` and `!` columns within this span are included in the total count. The resulting number of HTML columns is returned in the counter `LWR@tabhtmlcoltotal`.

```
5678 \newcommand*{\LWR@tabularhtmlcolumns}[2]{%
```

Count the starting index, compute ending index, and begin with the count being the  $\LaTeX$  span, to which additional @ and ! columns may be added:

```
5679 \setcounter{LWR@tabhtmlcolindex}{#1}%
5680 \setcounter{LWR@tabhtmlcoltotal}{#2}%
5681 \setcounter{LWR@tabhtmlcolend}{#1}%
5682 \addtocounter{LWR@tabhtmlcolend}{#2}%
```

If at the left edge, add the at/bang columns for the left edge:

```
5683 \ifthenelse{\value{LWR@tabhtmlcolindex}=1}{%
5684   \LWR@subtabularhtmlcolumns{leftedge}%
5685 }{}%
```

Walk across the  $\LaTeX$  columns looking for @ and ! columns:

```
5686 \whiledo{\value{LWR@tabhtmlcolindex}<\value{LWR@tabhtmlcolend}}{%
5687   \LWR@subtabularhtmlcolumns{\arabic{LWR@tabhtmlcolindex}}%
5688   \addtocounter{LWR@tabhtmlcolindex}{1}%
5689 }% \whiledo
5690 }

5691 \end{warpHTML}
```

## 59.18 Multicolumnrow

A print-mode version is defined here, and is also used during HTML output while inside a `lateximage`.

See section 180 for the HTML versions.

**for HTML & PRINT:** 5692 \begin{warpall}

```
\multicolumnrow {<1:cols>} {<2:halign>} [<3:vpos>] {<4:numrows>} [<5:bigstruts>] {<6:width>} [<7:fixup>]
{<8:text>}
```

For discussion of the use of `\DeclareExpandableDocumentCommand`, see:  
[https://tex.stackexchange.com/questions/168434/](https://tex.stackexchange.com/questions/168434/problem-with-abbreviation-of-multirow-and-multicolumn-latex)  
[problem-with-abbreviation-of-multirow-and-multicolumn-latex](https://tex.stackexchange.com/questions/168434/problem-with-abbreviation-of-multirow-and-multicolumn-latex)

After the user may have

```
5693 \AtBeginDocument{
```



`\@ifundefined{@xmultirow}` determines if multirow was never loaded.

```
5694 \@ifundefined{@xmultirow}
5695 {}% no version of multirow was loaded
5696 {% \@xmultirow defined, so some version of multirow was loaded
```

`\@ifpackageloaded{multirow}` determines if v2.0 or later of multirow was used, which included the `\ProvidesPackage` macro.

```
5697 \@ifpackageloaded{multirow}{% v2.0 or newer
5698 \@ifpackagelater{multirow}{2016/09/01}% 2016/09/27 for v2.0
5699 {% v2.0+:
5700 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
5701     {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
5702 {\multicolumn{#1}{#2}{\@xmultirow[#3]{#4}[#5]{#6}[#7]{#8}}}%
5703 }
5704 {% loaded but older, probably not executed:
5705 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
5706     {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
5707 {\multicolumn{#1}{#2}{\@xmultirow{#4}[#5]{#6}[#7]{#8}}}%
5708 }
5709 }% packageloaded{multirow}
```

If not `\@ifpackageloaded{multirow}` but `\@xmultirow` is defined, then this must be v1.6 or earlier, which did not `\ProvidesPackage{multirow}`, and did not have the `vposn` option.

```
5710 {% v1.6 or older did not \ProvidePackage
5711 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
5712     {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
5713 {\multicolumn{#1}{#2}{\@xmultirow{#4}[#5]{#6}[#7]{#8}}}%
5714 }
```

The user-level interface. This is provided if the HTML version was not already given.

```
5715 \providecommand*{\multicolumnrow}{\LWR@origmulticolumnrow}
5716 }% \@xmultirow defined, so multirow was loaded
5717 }% AtBeginDocument

5718 \end{warpall}
```

## 59.19 Utility macros inside a table

for HTML output: 5719 `\begin{warpHTML}`

```
5720 \newcommand*{\LWR@donothing}{}

```

In case bigdelim is not loaded:

```
5721 \newcommand*{\ldelim}{}

```

```
5722 \newcommand*{\rdelim}{}

```

```
5723 \end{warpHTML}

```

## 59.20 Special-case tabular markers

**for HTML & PRINT:** 5724 \begin{warpall}

`\TabularMacro` Place this just before inserting a custom macro in a table data cell. Doing so tells lwarp not to automatically start a new HTML table data cell yet. See section 8.7.

```
5725 \newcommand*{\TabularMacro}{}

```

```
5726 \end{warpall}

```

`\ResumeTabular` Used to resume tabular entries after resuming an environment.

### tabular inside another environment

When creating a new environment which contains a tabular environment, lwarp's emulation of the tabular does not automatically resume when the containing environment ends, resulting in corrupted HTML rows. To fix this, use `\ResumeTabular` as follows. This is ignored in print mode.

```
\StartDefiningTabulars % because & is used in a definition
\newenvironment{outerenvironment}
{
\tabular{cc}
left & right \\\
}
{
\TabularMacro\ResumeTabular
left & right \\\
\endtabular
}
\EndDefiningTabulars

```

**for HTML output:** 5727 \begin{warpHTML}

```

5728 \newcommand*{\ResumeTabular}{%
5729 \global\boolfalse{LWR@exitingtabular}%
5730 \global\boolfalse{LWR@tabularmutemods}%
5731 \LWR@getmynexttoken%
5732 }

```

```

5733 \end{warpHTML}

```

**for PRINT output:** 5734 \begin{warpprint}

```

5735 \newcommand*{\ResumeTabular}{}

```

```

5736 \end{warpprint}

```

## 59.21 Checking for a new table cell

**for HTML output:** 5737 \begin{warpHTML}

Bool LWR@exitingtabular When \end is found, turns off the next opening data tag.

```

5738 \newbool{LWR@exitingtabular}

```

Bool LWR@tabularmutemods Mutes HTML output for @, !, < and >.

This is used while printing the final row to generate \bottomrules.

```

5739 \newbool{LWR@tabularmutemods}

```

\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:

```

5740 \newcommand*{\LWR@tabledatacolumnntag}%
5741 {%
5742 \LWR@traceinfo{LWR@tabledatacolumnntag}%

```

\show\LWR@mynexttoken to see what tokens to look for

If not any of the below, start a new table cell:

```

5743 \let\mynext\LWR@tabledatasinglecolumnntag%

```

If exiting the tabular:

```

5744 \ifdefequal{\LWR@mynexttoken}{\end}%
5745 {\global\booltrue{LWR@exitingtabular}}{}%

```

longtable can have a caption in a cell

```
5746 \ifdefequal{\LWR@mynexttoken}{\caption}%
5747     {\let\mynext\LWR@donothing}{}%
```

Look for other things which would not start a table cell:

```
5748 \ifdefequal{\LWR@mynexttoken}{\multicolumn}%
5749     {\let\mynext\LWR@donothing}{}%
5750 \ifdefequal{\LWR@mynexttoken}{\multirow}%
5751     {\let\mynext\LWR@donothing}{}%
5752 \ifdefequal{\LWR@mynexttoken}{\multicolumnrow}%
5753     {\let\mynext\LWR@donothing}{}%
```

If an `\mrowcell`, this is a cell to be skipped over:

```
5754 \ifdefequal{\LWR@mynexttoken}{\mrowcell}%
5755     {\let\mynext\LWR@donothing}{}%
```

If an `\mcolrowcell`, this is a cell to be skipped over:

```
5756 \ifdefequal{\LWR@mynexttoken}{\mcolrowcell}%
5757     {\let\mynext\LWR@donothing}{}%
5758 %
5759 \ifdefequal{\LWR@mynexttoken}{\TabularMacro}%
5760     {\let\mynext\LWR@donothing}{}%
5761 %
5762 \ifdefequal{\LWR@mynexttoken}{\hline}%
5763     {\let\mynext\LWR@donothing}{}%
5764 %
5765 \ifdefequal{\LWR@mynexttoken}{\toprule}%
5766     {\let\mynext\LWR@donothing}{}%
5767 %
5768 \ifdefequal{\LWR@mynexttoken}{\midrule}%
5769     {\let\mynext\LWR@donothing}{}%
5770 %
5771 \ifdefequal{\LWR@mynexttoken}{\cmidrule}%
5772     {\let\mynext\LWR@donothing}{}%
5773 %
5774 \ifdefequal{\LWR@mynexttoken}{\specialrule}%
5775     {\let\mynext\LWR@donothing}{}%
5776 %
5777 \ifdefequal{\LWR@mynexttoken}{\cline}%
5778     {\let\mynext\LWR@donothing}{}%
5779 %
5780 \ifdefequal{\LWR@mynexttoken}{\bottomrule}%
5781     {\let\mynext\LWR@donothing}{}%
5782 %
5783 \ifdefequal{\LWR@mynexttoken}{\warpprintonly}%
```

```

5784     {\let\mynext\LWR@donothing}{}%
5785 %
5786 \ifdefequal{\LWR@mynexttoken}{\warpHTMLonly}%
5787     {\let\mynext\LWR@donothing}{}%
5788 %
5789 \ifdefequal{\LWR@mynexttoken}{\ldelim}%
5790     {\let\mynext\LWR@donothing}{}%
5791 %
5792 \ifdefequal{\LWR@mynexttoken}{\rdelim}%
5793     {\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:

```


5794 \mynext%
5795 }

5796 \end{warpHTML}

```

## 59.22 \mrowcell

**for HTML & PRINT:** 5797 \begin{warpall}

**\mrowcell**  The user must insert \mrowcell into any \multirow cells which must be skipped. This command has no action during print output.

```


5798 \newcommand*{\mrowcell}{}

5799 \end{warpall}

```

## 59.23 \mcolrowcell

**for HTML & PRINT:** 5800 \begin{warpall}

**\mcolrowcell**  The user must insert \mcolrowcell into any \multicolumnrow cells which must be skipped. This command has no action during print output.

```

5801 \newcommand*{\mcolrowcell}{}

```

```
5802 \end{warpall}
```

## 59.24 New \tabular definition

for HTML output: 5803 \begin{warpHTML}

These are default definitions in case booktabs is not loaded, and are not expected to be used, but must exist as placeholders.

```
5804 \newcommand*\LWR@origtoprule}[1]{\hline}
5805 \newcommand*\LWR@origmidrule}[1]{\hline}
5806 \LetLtxMacro\LWR@origcmidrule\cline
5807 \newcommand*\LWR@origbottomrule}[1]{\hline}
5808 \newcommand*\LWR@origaddlinespace}[1][{}]{ }
5809 \newcommand*\LWR@origmorecmidrules[{}]{ }
5810 \newcommand*\LWR@origspecialrule}[3]{\hline}
```

**\LWR@hline** The definition of \hline depends on whether tabs has been loaded. If so, optional space below the line may be specified, but will be ignored.

```
5811 \AtBeginDocument{
5812 \@ifpackageloaded{lwarp-tables}
5813 {
5814 \newcommand*\LWR@hline}[1][{}]{%
5815   \ifbool{FormatWP}{%
5816     {\LWR@docmidrule{1-\arabic{LWR@tabletotalcols}}}{%
5817       {\booltrue{LWR@doinghline}}}{%
5818       \LWR@getmynexttoken}%
5819 }
5820 {
5821 \newcommand*\LWR@hline[{}]{%
5822   \ifbool{FormatWP}{%
5823     {\LWR@docmidrule{1-\arabic{LWR@tabletotalcols}}}{%
5824       {\booltrue{LWR@doinghline}}}{%
5825       \LWR@getmynexttoken}%
5826 }
5827 }% AtBeginDocument
```

**\LWR@nullifyNoAutoSpacing** For babel-french, turn off auto spacing at the start of the tabular, then nullify the autospacing commands inside the tabular, since they were not compatible with the tabular column parsing code, which uses xstring.

```
5828 \AtBeginDocument{
5829 \@ifundefined{frenchbsetup}{%
5830 {% no babel-french
```

```

5831 \newcommand*{\LWR@nullifyNoAutoSpacing}{%
5832 }% no babel-french
5833 {% yes babel-french
5834 \newcommand*{\LWR@nullifyNoAutoSpacing}{%
5835 \NoAutoSpacing%
5836 \renewcommand*{\NoAutoSpacing}{}%
5837 \renewcommand*{\LWR@FBcancel}{}%
5838 }
5839 }% yes babel-french
5840 }% AtBeginDocument

```

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.

```

5841 \StartDefiningTabulars
5842
5843 \newenvironment*{LWR@tabular}[2][]
5844 {%
5845 \LWR@traceinfo{LWR@tabular started}%
5846 \addtocounter{LWR@tabulardepth}{1}%

```

Not yet started a table row:

```

5847 \global\boolfalse{LWR@startedrow}%

```

Not yet doing any rules:

```

5848 \global\boolfalse{LWR@doinghline}%
5849 \global\boolfalse{LWR@doingtbrule}%
5850 \global\boolfalse{LWR@doingcmidrule}%

```

For `babel-french`, turn off auto spacing one time, then nullify the autospacing commands since were not compatible with the tabular parsing code.

```

5851 \LWR@nullifyNoAutoSpacing%

```

Have not yet found the end of tabular command. Unmute the `@` and `!` columns.

```

5852 \global\boolfalse{LWR@exitingtabular}%
5853 \global\boolfalse{LWR@tabularmutemods}%

```

Create the table tag:

```

5854 \global\booltrue{LWR@intabularmetadata}%
5855 \LWR@forcenewpage
5856 \LWR@htmlblocktag{table}%

```

Parse the table columns:

```
5857 \LWR@parsetablecols{#2}%
```

Table col spec is: \LWR@tablecolspec which is a string of llccrr, etc.

Do not place the table inside a paragraph:

```
5858 \LWR@stoppars%
```

Track column #:

```
5859 \setcounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
5860 \boolfalse{LWR@tabularcelladded}%
```

Start looking for midrules:

```
5861 \LWR@clearmidrules%
```

\\ becomes a macro to end the table row:

```
5862 \LetLtxMacro{\\}{\LWR@tabularendoffline}%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with \LWR@getmynexttoken to see if the next token might create a new data cell:

The optional parameter for \hline supports the tabs package.

```
5863 \LWR@traceinfo{LWR@tabular: redefining macros}%
```

```
5864 \renewcommand*{\hline}{\LWR@hline}%
```

```
5865 %
```

```
5866 \RenewDocumentCommand{\cline}{m}%
```

```
5867 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%
```

```
5868 \DeclareDocumentCommand{\toprule}{o d() }
```

```
5869     {%
```

```
5870         \IfValueTF{##1}%
```

```
5871             {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
```

```
5872         {%
```

```
5873             \ifbool{FormatWP}%
```

```
5874                 {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
```

```
5875                 {\booltrue{LWR@doingtbrule}}}%
```

```
5876         }%
```

```
5877     \LWR@getmynexttoken}%
```



```

5878 %
5879 \DeclareDocumentCommand{\midrule}{o d()}%
5880   {%
5881     \IfValueTF{##1}%
5882       {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
5883       {%
5884         \ifbool{FormatWP}%
5885           {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
5886           {\booltrue{LWR@doinghline}}%
5887       }%
5888     \LWR@getmynexttoken}%
5889 %
5890 \DeclareDocumentCommand{\cmidrule}{O{\LWR@cmidrulewidth} d() m}%
5891 {\LWR@docmidrule[##1](##2){##3}\LWR@getmynexttoken}%
5892 %
5893 \DeclareDocumentCommand{\bottomrule}{o d()}
5894   {%
5895     \IfValueTF{##1}%
5896       {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
5897       {%
5898         \ifbool{FormatWP}%
5899           {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
5900           {\booltrue{LWR@doingtbrule}}%
5901       }%
5902     \LWR@getmynexttoken}%
5903 %
5904 \DeclareDocumentCommand{\addlinespace}{o}{}%
5905 \DeclareDocumentCommand{\morecmidrules}{l}{}%
5906 \DeclareDocumentCommand{\specialrule}{m m m d()}%
5907   {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}\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.

```

5908 \renewcommand{\multicolumn}{\LWR@htmlmulticolumn}%
5909 \renewcommand*\{mrowcell}{%
5910   \LWR@maybenewtablerow%
5911   \LWR@tabularleftedge%
5912   \global\booltrue{LWR@skippingmrowcell}%
5913 }%
5914 \renewcommand*\{mcolrowcell}{%
5915   \LWR@maybenewtablerow%
5916   \global\booltrue{LWR@skippingmcolrowcell}%
5917 }%
5918 \LetLtxMacro{\caption}{\LWR@longtabledatacaptiontag}%

```

Reset for new processing:

```

5919 \global\boolfalse{LWR@tableparcell}%

```

```

5920 \global\boolfalse{LWR@skippingmrowcell}%
5921 \global\boolfalse{LWR@skippingmcolrowcell}%
5922 \global\boolfalse{LWR@skipatbang}%
5923 \global\boolfalse{LWR@emptyatbang}%

```

Set & for its special meaning inside the tabular:

```

5924 \StartDefiningTabulars%
5925 \protected\gdef&{\LWR@tabularampersand}%

```

Look ahead for a possible table data cell:

```

5926 \LWR@traceinfo{LWR@tabular: about to LWR@getmynexttoken}%
5927 \LWR@getmynexttoken%
5928 }%

```

Ending the environment:

```

5929 {%
5930 \LWR@traceinfo{LWR@tabular ending}%
5931 \ifthenelse{%
5932   \(%
5933     \boolean{LWR@intabularmetadata} \AND%
5934     \NOT\boolean{LWR@tabularcelladded} \AND%
5935     \cnttest{\value{LWR@tablecolindex}}{=}{\value{LWR@tabletotalcols}}%
5936   \) \OR%
5937   \cnttest{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}%
5938 }{%
5939   \LWR@tabularfinishrow%
5940 }%
5941 {%
5942   \LWR@closetabledatacell%
5943 }%
5944 \LWR@htmlblocktag{/tr}%
5945 \LWR@htmlblocktag{/table}%
5946 \global\boolfalse{LWR@intabularmetadata}%

```

Unnest one level of tabular:

```

5947 \addtocounter{LWR@tabulardepth}{-1}%

```

Restore & to its usual meaning:

```

5948 \protected\gdef&{\LWR@origampmacro}%
5949 \EndDefiningTabulars%
5950 \LWR@traceinfo{LWR@tabular finished ending}%
5951 }
5952
5953 \EndDefiningTabulars

```

```
5954 \end{warpHTML}
```

## 59.25 Array

Pkg array

array is also automatically loaded by siunitx.

## 60 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 10 shows the data structures related to cross-referencing.

for HTML output: 5955 \begin{warpHTML}

### 60.1 Setup

\@currentlabelname To remember the most recently defined section name, description, or caption, for \nameref.

```
5956 \newcommand*\@currentlabelname{}
```

\LWR@stripperperiod {\text} [⟨.⟩]

Removes a trailing period.

```
5957 \def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%
```

\LWR@setlatestname {\object name}

Removes \label, strips any final period, and remembers the result.

```
5958 \newcommand*\LWR@setlatestname[1]{%
```

Table 10: Cross-referencing data structures

<b>Original <math>\LaTeX</math>:</b>	(print and HTML)
<code>\refstepcounter:</code> Steps the counter and sets <code>\@currentlabel</code> . <code>\@currentlabel:</code> <code>\p@&lt;ctr&gt;\the&lt;ctr&gt;</code> Updated by <code>\refstepcounter</code> . <code>\label:</code> Writes to the .aux file: <code>\newlabel{&lt;label&gt;}{\@currentlabel}{\thepage}}</code> <code>\newlabel:</code> When the .aux file is read, sets <code>\r@&lt;label&gt;</code> . <code>\r@&lt;label&gt;:</code> Set to: <code>{\@currentlabel}{\thepage}}</code> <code>\ref:</code> Returns the first part of <code>\r@&lt;label&gt;</code> . <code>\pageref:</code> Returns the second part of <code>\r@&lt;label&gt;</code> .	
<b>Added by lwarp:</b>	(HTML only)
<code>\label:</code> Adds HTML tags (section 60.3), plus <code>\splabel</code> data (section 60.2): <code>zLWR@name:</code> The section name for this label. <code>zLWR@htmlfilenumber:</code> The file number or name for this label. <code>zLWR@lateximagedepth:</code> The <code>lateximagedepth</code> for this label. <code>zLWR@lateximagenumber:</code> The <code>lateximagenumber</code> for this label. <code>\nameref:</code> Emulated from <code>hyperref</code> for <code>lwarp</code> . See section 60.4. <code>\ref</code> and <code>\nameref:</code> Adds HTML tags. See section 60.4.	
<b>Added by amsmath:</b>	(print and HTML)
<code>\label:</code> Execution is delayed until the math environment is completed. <code>\ltx@label:</code> $\LaTeX$ <code>\label</code> , (HTML: patched by <code>lwarp</code> ), later patched by <code>cleveref</code> .	
<b>Added by cleveref:</b>	(print and HTML)
<code>\refstepcounter:</code> Added: sets <code>\cref@currentlabel</code> . <code>\cref@currentlabel:</code> ( <code>&lt;type&gt;=&lt;ctr&gt;</code> unless an alias is used): <code>[&lt;type&gt;][\arabic{&lt;ctr&gt;}][&lt;parent&gt; &lt;trs&gt;]{\p@&lt;ctr&gt;\the&lt;ctr&gt;}</code> Also see section 46.4 for use with footnotes. <code>\label:</code> Writes to the .aux file: <code>\newlabel{&lt;label&gt;@cref}{\cref@currentlabel}{\thepage}}</code> <code>\newlabel:</code> (Unchanged.) When the .aux file is read, sets <code>\r@&lt;label&gt;@cref</code> . <code>\r@&lt;label&gt;@cref:</code> Set to: <code>{\cref@currentlabel}{\thepage}}</code> <b>Utility functions:</b> See <code>\cref@getlabel</code> , <code>\cref@gettype</code> , <code>\cref@getcounter</code> , <code>\cref@getprefix</code> . <b>Cross-referencing names:</b> <code>\crefname</code> and <code>\Crefname</code> assign human-readable names for references to this counter type.	
<b>Additionally patched by lwarp:</b>	(HTML only)
<code>\cref, etc.:</code> Modified for <code>lwarp</code> . See section 72. <code>\label inside math:</code> See section 65.5.1.	
<b>Footnotes:</b> See <code>\noteentry</code> in section 46.4.	

Remove `\label` and other commands from the name, the strip any final period. See `zref-titleref` and `getttitlestring`.

```
5959 \GetTitleStringExpand{#1}%
5960 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
5961 \edef\@currentlabelname{%
5962 \expandafter\LWR@stripperperiod\@currentlabelname%
5963 \ltx@empty.\ltx@empty\@nil%
5964 }%
5965 }
```

## 60.2 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`:

```
5966 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
5967 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
5968 \zref@newprop{zLWR@htmlfilenumber}{%
5969 \ifbool{FileSectionNames}{\LWR@thisfilename}{\arabic{LWR@htmlfilenumber}}}%
5970 }%
```

Additional properties for `lateximages`:

```
5971 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
5972 \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`:

```
5973 \zref@addprop{special}{zLWR@name}
5974 \zref@addprop{special}{zLWR@htmlfilenumber}
5975 \zref@addprop{special}{zLWR@lateximagedepth}
5976 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
5977 \newcommand*{\LWR@spref}[2]{%
5978 \zref@extractdefault{#1}{#2}{??}}
```

`\LWR@nameref`  $\{\langle label \rangle\}$  Returns the section name for this label:

```
5979 \newcommand*{\LWR@nameref}[1]{%
5980 \LWR@spref{#1}{z\LWR@name}%
5981 }
```

`\LWR@htmlfileref`  $\{\langle label \rangle\}$  Returns the file number for this label:

```
5982 \newcommand*{\LWR@htmlfileref}[1]{%
5983 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
5984 \LWR@spref{#1}{z\LWR@htmlfilenumber}%
5985 }
```

`\LWR@lateximagedepthref`  $\{\langle label \rangle\}$  Returns the lateximagedepth for this label:

```
5986 \newcommand*{\LWR@lateximagedepthref}[1]{%
5987 \LWR@spref{#1}{z\LWR@lateximagedepth}%
5988 }
```

`\LWR@lateximagenumberref`  $\{\langle label \rangle\}$  Returns the lateximagenumber for this label:

```
5989 \newcommand*{\LWR@lateximagenumberref}[1]{%
5990 \LWR@spref{#1}{z\LWR@lateximagenumber}%
5991 }
```

`\LWR@splabel`  $\{\langle label \rangle\}$  Sanitize the name and then creates the label:

```
5992 \newcommand*{\LWR@splabel}[1]{%
5993 \LWR@setlatestname{\@currentlabelname}%
5994 \zref@labelbylist{#1}{special}}
```

## 60.3 Labels

`\LWR@subsublabel`  $\{\langle label \rangle\}$  Creates an HTML id tag.

```
5995 \newcommand*{\LWR@subsublabel}[1]{%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```
5996 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{%
5997 {}%
5998 {% not lateximage
```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```
5999 \ifbool{LWR@doingstartpars}%
6000 {% pars allowed
6001 \ifbool{LWR@doingapar}
6002 {% par started
6003 \LWR@htmltag{a id="#1"{} }\LWR@htmltag{/a}%
6004 }% par started
6005 {% par not started
6006 \LWR@stoppars%
6007 \LWR@htmltag{a id="#1"{} }\LWR@htmltag{/a}%
6008 \LWR@startpars%
6009 }% par not started
6010 }% pars allowed
6011 {% pars not allowed
6012 \LWR@htmltag{a id="#1"{} }\LWR@htmltag{/a}%
6013 }% pars not allowed
6014 }% not lateximage
6015 }
```

`\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  $\TeX$  `\label`.

`cleveref` later encases this to add its own cross-referencing.

The optional  $\langle type \rangle$  is per the `ntheorem` package, and is ignored.

```
6016 \NewDocumentCommand{\LWR@newlabel}{m o}{%
6017 \LWR@traceinfo{LWR@newlabel: starting}%
6018 \LWR@traceinfo{LWR@newlabel: !#1!}%
6019 % \@bsphack%
```

Create a traditional  $\TeX$  label, as modified by `cleveref`:

```
6020 \LWR@origlabel{#1}%
```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```

6021 \LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
6022 \LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
6023 \LWR@traceinfo{LWR@newlabel: LWR@htmlfilenumber is \arabic{LWR@htmlfilenumber}}%
6024 \LWR@splabel{#1}%
6025 \LWR@subsublabel{#1}%
6026 % \@esphack%
6027 \LWR@traceinfo{LWR@newlabel: done}%
6028 }

```

## 60.4 References

`\LWR@startref` `{\label}` (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, # character, and a label.

```

6029 \newcommand*{\LWR@startref}[1]
6030 {%
6031 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
6032 \LWR@traceinfo{LWR@startref A: !#1!}%

```

Create the filename part of the link:

```

6033 \LWR@htmltag{a href="%
6034 \LWR@traceinfo{LWR@startref B}%
6035 \LWR@htmlrefsectionfilename{#1}%
6036 \LWR@traceinfo{LWR@startref C}%
6037 \#%

```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```

6038 \LWR@traceinfo{LWR@startref D: !#1!}%
6039 \ifthenelse{\equal{\LWR@lidref}{??}}%

```

“??” if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:

```

6040 {%
6041 \LWR@traceinfo{LWR@startref D0: ??}%
6042 ??%
6043 }%

```

If `LWR@lateximagedepth` is known. Use a `lateximage` if the depth is greater than zero, or a regular link otherwise:



```

6044 {%
6045   \LWR@traceinfo{LWR@startref D1: \LWR@lidref}%
6046   \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}%
6047   {%
6048     \LWR@traceinfo{LWR@startref D2: \LWR@lidref}%
6049     lateximage\LWR@lateximagenumberref{#1}%
6050   }%
6051   {%
6052     \LWR@traceinfo{LWR@startref D3}%
6053     #1%
6054   }%
6055 }%
6056 \LWR@traceinfo{LWR@startref E}%

```

Closing quote:

```

6057 "{}}%
6058 \LWR@traceinfo{LWR@startref F}%
6059 }

```

`\LWR@subnewref` `{\label}` `{\label or sub@label}`

Factored for the subfig package. Uses the original label for the hyper-reference, but prints its own text, such as “1(b)”.

```

6060 \NewDocumentCommand{\LWR@subnewref}{m m}{%
6061 \LWR@traceinfo{LWR@subnewref #1 #2}%
6062 \LWR@startref{#1}%
6063 \LWR@origref{#2}%
6064 \LWR@htmltag{/a}%
6065 }

```

`\ref` \* `{\label}` `\ref` is `\let` to `\LWR@newref`

`\LWR@newref` \* `{\label}` Create an internal document reference link, or without a link if starred per hyperref.

```

6066 \NewDocumentCommand{\LWR@newref}{s m}{%
6067 \LWR@traceinfo{LWR@newref #2}%
6068 \IfBooleanTF{#1}%
6069 {\LWR@origref{#2}}%
6070 {\LWR@subnewref{#2}{#2}}%
6071 }

```

`\pagerefPageFor` Text for page references.

```

6072 \newcommand*{\pagerefPageFor}{see }

```

`\pageref` \*  $\langle label \rangle$  Create an internal document reference, or just the unlinked number if starred, per `hyperref`.

```
6073 \NewDocumentCommand{\LWR@newpageref}{s m}{%
6074 \IfBooleanTF{#1}%
6075 {(\pagerefPageFor\LWR@origref{#2})}%
6076 {(\cpageref{#2})}%
6077 }
```

`\nameref`  $\langle label \rangle$

```
6078 \DeclareRobustCommand*\nameref[1]{%
6079 \LWR@traceinfo{nameref A}%
6080 \LWR@startref{#1}%
6081 \LWR@traceinfo{nameref B}%
6082 \LWR@nameref{#1}%
6083 \LWR@traceinfo{nameref C}%
6084 \LWR@htmltag{/a}%
6085 \LWR@traceinfo{nameref D}%
6086 }
```

`\Nameref`  $\langle label \rangle$  In print, adds the page number. In HTML, does not.

```
6087 \let\Nameref\nameref
```

## 60.5 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.

```
6088 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
6089 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Create a link with a text name:

`\LWR@subhyperref`  $\{\langle URL \rangle\} \{\langle text \rangle\}$

```
6090 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
6091 \LWR@htmltag{a href="{#1}" target="_{blank}"\LWR@orignewline}#2\LWR@htmltag{/a}%
6092 \LWR@ensuredoingapar%
6093 }
```

`\LWR@subhyperrefclass`  $\{\langle URL \rangle\} \{\langle text \rangle\} \{\langle htmlclass \rangle\}$

```
6094 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
6095 \LWR@htmltag{a href="{#1}"
6096 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
6097 \LWR@ensuredoingapar%
6098 }
```

`\href`  $[\langle options \rangle] \{\langle URL \rangle\} \{\langle text \rangle\}$

Create a link with accompanying text:

```
6099 \NewDocumentCommand{\LWR@hrefb}{0{} m +m}{%
6100 \LWR@subhyperref{#2}{#3}%
6101 \endgroup%
6102 \LWR@ensuredoingapar%
6103 }
6104
6105 \DeclareRobustCommand*\href}{%
6106 \LWR@ensuredoingapar%
6107 \begingroup%
6108 \catcode'\_ =12
6109 \LWR@hrefb%
6110 }
```

`\nolinkurl`  $\{\langle URL \rangle\}$

Print the name of the link without creating the link:

```
6111 \newcommand*\LWR@nolinkurlb}[1]{#1\endgroup\LWR@ensuredoingapar}
6112
6113 \DeclareRobustCommand*\nolinkurl}{%
6114 \LWR@ensuredoingapar%
6115 \begingroup\catcode'\_ =12
6116 \LWR@nolinkurlb%
6117 }
```

`\url`  $\{\langle URL \rangle\}$

Create a link whose text name is the address of the link. The url package may redefine \url, so it is \let to \LWR@urlahere and also redefined by lwarp-url.

```

6118 \newcommand*\LWR@urlb}[1]{%
6119 \href{#1}{#1}%
6120 \endgroup%
6121 \LWR@ensuredoingapar%
6122 }
6123
6124 \DeclareRobustCommand*\LWR@urla{%
6125 \LWR@ensuredoingapar%
6126 \begingroup\catcode'\_ =12
6127 \LWR@urlb%
6128 }
6129
6130 \let\url\LWR@urla

```

\LWR@subinlineimage [*<alttag>*] [*<class>*] [*<filename>*] [*<extension>*] [*<style>*]

```

6131 \newcommand*\LWR@subinlineimage}[5][]{%
6132 \ifthenelse{\equal{#1}{}}{%
6133 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"{}{}}}%
6134 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{}{}}}%
6135 }

```

6136 \end{warpHTML}

Table 11: 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  $\TeX$  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.
- 

## 61 Floats

Floats are supported, although partially through emulation.

Table 11 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”.

## 61.1 Float captions

for HTML output: 6137 \begin{warpHTML}

\LWR@floatbegin {<type>} [<placement>]

Begins a \newfloat environment.

```
6138 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
6139 \ifbool{FormatWP}{\newline}{}%
6140 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
6141 \addtocounter{LWR@thisfloat}{1}%
6142 \booltrue{LWR@freezethisfloat}%
```

```
6143 \begingroup%
```

Settings while inside the environment:

```
6144 \LWR@origraggedright%
```

Open an HTML figure tag:

```
6145 \LWR@htmltag{figure id="autofloat-\arabic{LWR@thisfloat}" class="#1"}%
6146 \ifbool{FormatWP}{%
6147   \LWR@orignewline%
6148   \LWR@BlockClassWP}{\wp#1}%
6149 }{}%
```

```
6150 \renewcommand*{\@capttype}{#1}%
6151 \caption@settype{#1}%
6152 \LWR@startpars%
6153 \ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%
6154
6155 === begin #1 ===
6156
6157 }{}%
6158 }
```

\@float Support packages which create floats directly.  
 \@dblfloat

```
6159 \let\@float\LWR@floatbegin
6160 \let\@dblfloat\LWR@floatbegin
```

`\LWR@floatend` Ends a `\newfloat` environment.

```
6161 \newcommand*{\LWR@floatend}{%
6162 \ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%
6163
6164 === end ===
6165
6166 }{}%
6167 \LWR@stoppars%
```

Close an HTML figure tag:

```
6168 \ifbool{FormatWP}{\endLWR@BlockClassWP}{}%
6169 \LWR@htmlelementend{figure}%
6170 \endgroup%
6171 \boolfalse{LWR@freezethisfloat}%
6172 \LWR@startpars%
6173 \ifbool{FormatWP}{\newline}{}%
6174 }
```

`\end@float` Support packages which create floats directly.  
`\end@dblfloat`

```
6175 \let\end@float\LWR@floatend
6176 \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.

```
6177 \newcounter{LWR@thisfloat}
```

**Bool** `LWR@freezethisfloat` Prevents multiple increments of `\LWR@thisfloat` inside a float.

```
6178 \newbool{LWR@freezethisfloat}
6179 \boolfalse{LWR@freezethisfloat}
```

`\LWR@maybeinthisfloat`

```
6180 \newcommand*{\LWR@maybeinthisfloat}{%
6181 \ifbool{LWR@freezethisfloat}{\addtocounter{LWR@thisfloat}{1}}%
6182 }
```

`\@capytype` Remembers which float type is in use.

```
6183 \newcommand*{\@capytype}{}
```

### 61.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```
6184 \AtBeginDocument{\providecommand*\CaptionSeparator}{:~}}
```

`\@makecaption`  $\{\langle name \text{ and } num \rangle\} \{\langle text \rangle\}$

Prints the float type and number, the caption separator, and the caption text.

```
6185 \AtBeginDocument{\renewcommand{\@makecaption}[2]{\#1\CaptionSeparator\#2}}
```

### 61.1.2 Caption and LOF linking and tracking

When a new HTML file is marked in the  $\LaTeX$  PDF file, the  $\LaTeX$  page number at that point is stored in `LWR@latestautopage`, (and the associated filename is remembered by the special  $\LaTeX$  labels). This page number is used to generate an autofloat HTML `<id>` in the HTML output at the start of the new HTML file. Meanwhile, there is a float counter used to generate an HTML autofloat `<id>` at the start of the float itself in the HTML file. The autopage and autofloat values to use for each float are written to the `.lof`, etc. files just before each float's entry. These values are used by `\l@figure`, etc. to create the HTML links in the List of Figures, etc.

Ctr `LWR@nextautofloat` Tracks autofloat for floats. Tracks autopage for floats.

Ctr `LWR@nextautopage` These are updated per float as the `.lof` file is read.

```
6186 \newcounter{LWR@nextautofloat}
```

```
6187 \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.

```
6188 \newcommand*\LWRsetnextfloat[2]{%
```

```
6189 \setcounter{LWR@nextautopage}{\#1}%
```

```
6190 \setcounter{LWR@nextautofloat}{\#2}%
```

```
6191 }
```

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.

```
6192 \newcounter{LWR@latestautopage}
```

```
6193 \setcounter{LWR@latestautopage}{1}
```



Env LWR@figcaption Encapsulates a caption inside <figcaption>, and if FormatWP then also a <div> with an italic style.

```

6194 \newenvironment*{LWR@figcaption}
6195 {%
6196 \LWR@htmlblocktag{figcaption}
6197 \ifbool{FormatWP}{%
6198 \begin{BlockClass}[font-style:italic]{italic}
6199 \LWR@origvspace*{\baselineskip}
6200 }{}%
6201 }
6202 {%
6203 \ifbool{FormatWP}{\end{BlockClass}}{}%
6204 \LWR@htmlblocktag{/figcaption}%
6205 }

6206 \let\LWR@origcaption@begin\caption@begin
6207 \let\LWR@origcaption@end\caption@end
6208 \let\LWR@orig@@par\@@par

```

\LWR@caption@begin Low-level patches to create HTML tags for captions.

```

6209 \newcommand{\LWR@caption@begin}
6210 {
6211 \LWR@traceinfo{LWR@caption@begin}%

```

Keep par and minipage changes local:

```

6212 \begingroup%

```

The caption code was not allowing the closing par tag:

```

6213 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%

```

No need for a minipage or \parbox inside the caption:

```

6214 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
6215 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{##5}%

```

Enclose the original caption code inside an HTML tag:

```

6216 \LWR@figcaption%
6217 \LWR@origcaption@begin%
6218 }

```

\LWR@caption@end Low-level patches to create HTML tags for captions.

```

6219 \newcommand{\LWR@caption@end}
6220 {%
6221 \LWR@origcaption@end%

```

Subcaptions were being over-written by the closing HTML tag:

```

6222 \LWR@origvspace*{\baselineskip}%

```

Closing tag:

```

6223 \endLWR@figcaption%
6224 \endgroup%
6225 % \leavevmode% avoid bad space factor (0) error
6226 \LWR@traceinfo{\LWR@caption@end: done}%
6227 }

```

`\caption@begin` Low-level patches to create HTML tags for captions.  
`\caption@end`

```

6228 \AtBeginDocument{
6229 \let\caption@begin\LWR@caption@begin
6230 \let\caption@end\LWR@caption@end
6231 }

```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```

6232 \let\LWR@origcaptionlistentry\captionlistentry
6233
6234 \renewcommand*{\captionlistentry}{%
6235 \LWR@maybeinthisfloat%
6236 \LWR@ensuredoingapar%
6237 \LWR@htmltag{a id="autofloat-\arabic{\LWR@thisfloat}"{}}\LWR@htmltag{/a}%
6238 \LWR@origcaptionlistentry%
6239 }
6240
6241 \def\LWR@LTcaptionlistentry{%
6242 \LWR@ensuredoingapar%
6243 \LWR@htmltag{a id="autofloat-\arabic{\LWR@thisfloat}"{}}\LWR@htmltag{/a}%
6244 \bgroup
6245 \@ifstar{\egroup\LWR@LTcaptionlistentry}% gobble *
6246 {\egroup\LWR@LTcaptionlistentry}%
6247 \def\LWR@LTcaptionlistentry#1{%
6248 \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.

```

6249 \let\LWR@origaddcontentsline\addcontentsline
6250
6251 \renewcommand*{\addcontentsline}[3]{%
6252 \ifthenelse{\equal{#1}{toc}}{}{%
6253   \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{}
6254   \addtocontents{\@nameuse{ext@#2}}{%
6255     \protect\LWRsetnextfloat%
6256     {\arabic{LWR@latestautopage}}%
6257     {\arabic{LWR@thisfloat}}%
6258   }% addtocontents
6259 }% not toc
6260 \LWR@origaddcontentsline{#1}{#2}{#3}%
6261 }

```

`\captionof` Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

6262 \AtBeginDocument{
6263 \let\LWR@origcaptionof\captionof
6264
6265 \renewcommand*{\captionof}{%
6266 \LWR@maybeinthisfloat%
6267 \LWR@stoppars
6268 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
6269 \LWR@origcaptionof%
6270 }
6271 }

6272 \end{warpHTML}

```

## 62 Table of Contents, LOF, LOT

This section controls the generation of the TOC, LOF, and LOT.

The `.toc`, `.lof`, and `.lot` files are named by the source code `\jobname`.

In HTML, the printed tables are placed inside a `<div>` of class `toc`, `lof`, or `lot`.

A “`sidetoc`” is provided which prints a subset of the toc on the side of each page other than the homepage.

The regular  $\TeX$  infrastructure is used for toc, along with some patches to generate HTML output.

for HTML output: 6273 \begin{warpHTML}

## 62.1 Reading and printing the TOC

`\LWR@myshorttoc`  $\{\langle toc/lof/lot \rangle\}$

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  $\TeX$ , 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 sideroc.

```
6274 \newcommand*\LWR@myshorttoc}[1]{
6275 \LWR@ensuredoingapar
```

Only if the file exists:

```
6276 \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 $\LaTeX$ , 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.

```
6277 \begingroup
6278 % \ifxetexorluatex%
6279 % \else
6280 % \inputencoding{latin1}% currently disabled
6281 % \fi
6282 \makeatletter
```

Read in the TOC file:

```
6283 \@input{\jobname.#1}
6284 % \makeatother
6285 \endgroup
6286 }%
6287 {}%
6288 }
```

`\LWR@subtableofcontents`  $\{\langle toc/lof/lot \rangle\} \{\langle sectionstarname \rangle\}$

Places a TOC/LOF/LOT at the current position.

```
6289 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```
6290 \@ifundefined{chapter}
6291 {\LWR@closeprevious{\LWR@depthsection}}
6292 {\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
6293 \LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
6294 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}
```

Create a new HTML nav containing the TOC/LOF/LOT:

```
6295 \LWR@htmlelementclass{nav}{#1}
```

Create the actual list:

```
6296 \LWR@myshorttoc{#1}
```

Close the nav:

```
6297 \LWR@htmlelementclassend{nav}{#1}
6298 }
```

```
\@starttoc {\langle ext\rangle}
```

Patch \@starttoc to encapsulate the TOC inside HTML tags:

```
6299 \let\LWR@orig@starttoc\@starttoc
6300
6301 \renewcommand{\@starttoc}[1]{
6302 \LWR@htmlelementclass{nav}{#1}
6303 \LWR@orig@starttoc{#1}
6304 \LWR@htmlelementclassend{nav}{#1}
6305 }
```

`\tableofcontents` Patch `\tableofcontents`, etc. to print footnotes first. newfloat uses `\listoffigures` for all future float types.

```
6306 \let\LWR@origtableofcontents\tableofcontents
6307
6308 \renewcommand*\tableofcontents{%
```

Do not print the table of contents if formatting for a word processor, which will presumably auto-generate its own updated table of contents:

```
6309 \ifboolexpr{bool{FormatWP} and bool{WPMarkTOC}}{
6310
6311 === table of contents ===
6312
6313 }
6314 {
```

Copy the .toc file to .sidetoc for printing the sideroc. The original .toc file is renewed when \tableofcontents is finished.

```
6315 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
6316 \LWR@printpendingfootnotes
6317 \LWR@origtableofcontents
6318 }
6319 }
```

#### \listoffigures

```
6320 \let\LWR@origlistoffigures\listoffigures
6321
6322 \renewcommand*{\listoffigures}{
6323 \ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{
6324
6325 === list of figures ===
6326
6327 }
6328 {
6329 \LWR@printpendingfootnotes
6330 \LWR@origlistoffigures
6331 }
6332 }
```

#### \listoftables

```
6333 \let\LWR@origlistoftables\listoftables
6334
6335 \renewcommand*{\listoftables}{
6336 \ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{
6337
6338 === list of tables ===
6339
6340 }
6341 {
6342 \LWR@printpendingfootnotes
6343 \LWR@origlistoftables
```

```
6344 }
6345 }
```

## 62.2 High-level TOC commands

`\listof`  $\{\langle type \rangle\} \{\langle title \rangle\}$

Emulate the `\listof` command from the float package (section 132). Used to create lists of custom float types. Also used to redefine the standard  $\text{\LaTeX}$  `\listoffigures` and `\listoftables` commands.

```
6346 \NewDocumentCommand{\listof}{m +m}{%
6347 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
6348 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
6349 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
6350 \jobname.\csuse{ext@#1}\relax
6351 }
```

## 62.3 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.

---

```
6352 \end{warpHTML}
```

**for HTML & PRINT:** 6353 `\begin{warpall}`

**Ctrl SideTOCDepth** Controls how deep the side-TOC gets. Use a standard  $\text{\LaTeX}$  section level similar to `tocdepth`.

```
6354 \newcounter{SideTOCDepth}
6355 \setcounter{SideTOCDepth}{1}
```

`\sitetocname` Holds the default name for the sideroc.

```
6356 \newcommand{\sitetocname}{Contents}
```

```
6357 \end{warpall}
```

for HTML output: `6358 \begin{warpHTML}`

`\LWR@sitetoc` Creates the actual side-TOC.

```
6359 \newcommand*{\LWR@sitetoc}{
6360 \LWR@forcenewpage
6361 \LWR@stoppars
6362 }
```

The entire sideroc is placed into a nav of class `sitetoc`.

```
6363 \LWR@htmlelementclass{nav}{sitetoc}
6364
6365 \setcounter{tocdepth}{\value{SideTOCDepth}}
6366 }
```

The title is placed into a `<div>` of class `sidetocitle`, and may contain paragraphs.

```
6367 \begin{BlockClass}{sitetocitle}
6368 \sitetocname
6369 \end{BlockClass}
```

The table of contents is placed into a `<div>` of class `sitetoccontents`.

```
6370 \begin{BlockClass}{sitetoccontents}
6371 \LinkHome
6372
6373 \LWR@myshorttoc{sitetoc}
6374 \end{BlockClass}
6375 \LWR@htmlelementclassend{nav}{sitetoc}
6376 }
```

## 62.4 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:



```

6377 \renewcommand*{\numberline}[1]{%
6378 \LWR@sectionnumber{#1}\quad%
6379 }

```

`\hypertoc`  $\langle 1: \textit{depth} \rangle \langle 2: \textit{type} \rangle \langle 3: \textit{name} \rangle \langle 4: \textit{page} \rangle$

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

```

6380 \NewDocumentCommand{\hypertoc}{m m +m m}{%

```

Respond to `tocdepth`:

```

6381 \ifthenelse{\cnttest{#1}{<=}}{\value{tocdepth}}{%
6382     \LWR@startpars%

```

Create an HTML link to `filename#autosec-(page)`, with text of the caption, of the given HTML class.

```

6383     \LWR@subhyperrefclass{%
6384     \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
6385     \LWR@stoppars%
6386 }
6387 {}
6388 }

```

Ctrl `lofdepth` TOC depth for figures.

```

6389 \newcounter{lofdepth}
6390 \setcounter{lofdepth}{1}

```

Ctrl `lotdepth` TOC depth for tables.

```

6391 \newcounter{lotdepth}
6392 \setcounter{lotdepth}{1}

```

`\hypertocfloat`  $\langle 1: \textit{depth} \rangle \langle 2: \textit{type} \rangle \langle 3: \textit{ext of parent} \rangle \langle 4: \textit{caption} \rangle \langle 5: \textit{page} \rangle$

**#1** is depth

**#2** is figure, table, etc.

**#3** is lof, lot, of the parent.

**#4** the text of the caption

**#5** page number

```
6393 \newcommand{\hypertocfloat}[5]{%
6394 \LWR@startpars
```

If some float-creation package has not yet defined the float type's lofdepth counter, etc, define it here:

```
6395 \@ifundefined{c@#3depth}{%
6396 \newcounter{#3depth}%
6397 \setcounter{#3depth}{1}%
6398 }{}%
```

Respond to lofdepth, etc.:

```
6399 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
6400 \ifthenelse{\cnttest{#1}{<=}}{\arabic{#3depth}}{}%
6401 \LWR@startpars%
```

Create an HTML link to filename#autofloat-(float number), with text of the caption, of the given HTML class.

```
6402 \LWR@subhyperrefclass{%
6403 \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
6404 \#autofloat-\arabic{LWR@nextautofloat}}%
6405 {#4}{toc#2}%
6406 \LWR@stoppars%
6407 }{}%
6408 }
```

Automatically called by \contentsline:

```
6409 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
6410 \DeclareDocumentCommand{\l@chapter}{m m}
6411 {\hypertoc{0}{chapter}{#1}{#2}}
6412 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
6413 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
6414 \renewcommand{\l@subsubsection}[2]
6415 {\hypertoc{3}{subsubsection}{#1}{#2}}
6416 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
```

```

6417 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
6418 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
6419 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

6420 \end{warpHTML}

```

## 63 Index and glossary

See:

[http://tex.stackexchange.com/questions/187038/  
how-to-mention-section-number-in-index-created-by-imakeidx](http://tex.stackexchange.com/questions/187038/how-to-mention-section-number-in-index-created-by-imakeidx)

Index links are tracked by the counter LWR@autoindex. This counter is used to create a label for each index entry, and a reference to this label for each entry in the index listing. This method allows each index entry to link directly to its exact position in the document.

for HTML output: 6421 \begin{warpHTML}

```

6422 \newcounter{LWR@autoindex}
6423 \setcounter{LWR@autoindex}{0}
6424
6425 \newcounter{LWR@autoglossary}
6426 \setcounter{LWR@autoglossary}{0}

```

\printindex

```

6427 \let\LWR@origprintindex\printindex
6428
6429 \renewcommand*\printindex{
6430 {
6431 \LWR@startpars
6432 \LWR@origprintindex
6433 }

```

Env theindex

```

6434 \@ifundefined{chapter}
6435 {\newcommand*\LWR@indexsection}[1]{\section*{#1}}
6436 {\newcommand*\LWR@indexsection}[1]{\chapter*{#1}}
6437
6438 \renewenvironment*{theindex}{%
6439 \LWR@indexsection{\indexname}%
6440 \let\item\LWR@indexitem%

```

```

6441 \let\subitem\LWR@indexsubitem%
6442 \let\subsubitem\LWR@indexsubsubitem%
6443 \{}

```

\LWR@indexitem

```

6444 \newcommand{\LWR@indexitem}{
6445
6446 \InlineClass{indexitem}{}}
6447 }

```

\LWR@indexitem

```

6448 \newcommand{\LWR@indexsubitem}{
6449
6450 \InlineClass{indexsubitem}{}}
6451 }

```

\LWR@indexitem

```

6452 \newcommand{\LWR@indexsubsubitem}{
6453
6454 \InlineClass{indexsubsubitem}{}}
6455 }

```

\@wrindex {<term>}      Redefined to write the LWR@latestautopage counter instead of page

```

6456 \def\@wrindex#1{%
6457 \addtocounter{LWR@autoindex}{1}%
6458 \LWR@newlabel{LWRindex-\theLWR@autoindex}%
6459 \protected@write\@indexfile{%
6460 {\string\indexentry{#1}{\theLWR@autoindex}}%
6461 \endgroup
6462 \@esphack}

```

\@wrglossary {<term>}      Redefined to write the LWR@latestautopage counter instead of page

```

6463 \def\@wrglossary#1{%
6464 \addtocounter{LWR@autoglossary}{1}%
6465 \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
6466 \protected@write\@glossaryfile{%
6467 {\string\glossaryentry{#1}{\theLWR@autoglossary}}%
6468 \endgroup
6469 \@esphack}

```

\hyperindexref {<autosecnumber>}

`\hyperindexref{web address}` is inserted into `*.ind` by the xindy style file `lwarp.xdy`

```
6470 \newcommand*\hyperindexref}[1]{\nameref{LWRindex-#1}}
```

```
6471 \end{warpHTML}
```

**for PRINT output:** A null command for print mode, in case hyperref was not used:

```
6472 \begin{warpprint}
6473 \newcommand*\hyperindexref}[1]{#1}
6474 \end{warpprint}
```

**for HTML & PRINT:** For the glossaries package, try to prevent an error where `\glo@name` was not found:

```
6475 \begin{warppall}
6476 \providecommand*\glo@name{}
6477 \end{warppall}
```

## 64 Restoring original formatting

`\LWR@restoreorigformatting` Used to temporarily restore the print-mode meaning of a number of formatting, graphics, and symbols-related macros while generating svg math or a lateximage. A number of packages will `\appto` additional actions to this macro.

Various packages add to this macro using `\appto`.

```
6478 \newcommand*\LWR@restoreorigformatting{%
6479 \LWR@traceinfo{\LWR@restoreorigformatting}%
6480 \linespread{1}%

6481 \LetLtxMacro\ref\LWR@origref%{} syntax highlighting
6482 \RenewDocumentCommand{\InlineClass}{o m +m}{##3}%
6483 \RenewDocumentEnvironment{BlockClass}{o m}{}{}%
6484 \renewcommand*\BlockClassSingle[2]{##2}%
6485 \LetLtxMacro{\hspace}{\LWR@orighspace}%
6486 \LetLtxMacro\rule\LWR@origrule%
6487 \let\,\LWR@origcomma% disable HTML short unbreakable space
6488 \let\textellipsis\LWR@origtextellipsis%
6489 \let\textless\LWR@origtextless%
6490 \let\textgreater\LWR@origtextgreater%
6491 \LetLtxMacro{\textrm}{\LWR@origtextrm}%
6492 \LetLtxMacro{\textsf}{\LWR@origtextsf}%
6493 \LetLtxMacro{\texttt}{\LWR@origtexttt}%
6494 \LetLtxMacro{\textbf}{\LWR@origtextbf}%
6495 \LetLtxMacro{\textmd}{\LWR@origtextmd}%

```

```
6496 \LetLtxMacro{\textit}{\LWR@origtextit}%
6497 \LetLtxMacro{\textsl}{\LWR@origtextsl}%
6498 \LetLtxMacro{\textsc}{\LWR@origtextsc}%
6499 \LetLtxMacro{\textup}{\LWR@origtextup}%
6500 \LetLtxMacro{\textnormal}{\LWR@origtextnormal}%
6501 \LetLtxMacro{\emph}{\LWR@origemph}%
6502 \LetLtxMacro{\rmfamily}{\LWR@origrmfamily}%
6503 \LetLtxMacro{\sffamily}{\LWR@origsffamily}%
6504 \LetLtxMacro{\ttfamily}{\LWR@origttfamily}%
6505 \LetLtxMacro{\bfseries}{\LWR@origbfseries}%
6506 \LetLtxMacro{\mdseries}{\LWR@origmdseries}%
6507 \LetLtxMacro{\upshape}{\LWR@origupshape}%
6508 \LetLtxMacro{\slshape}{\LWR@origslshape}%
6509 \LetLtxMacro{\scshape}{\LWR@origscshape}%
6510 \LetLtxMacro{\itshape}{\LWR@origitshape}%
6511 \LetLtxMacro{\em}{\LWR@origem}%
6512 \LetLtxMacro{\normalfont}{\LWR@orignormalfont}%
6513 \let\sp\LWR@origsp%
6514 \let\sb\LWR@origsb%
6515 \LetLtxMacro\textsuperscript\LWR@origtextsuperscript%
6516 \LetLtxMacro@textsuperscript\LWR@orig@textsuperscript%
6517 \LetLtxMacro\textsubscript\LWR@origtextsubscript%
6518 \LetLtxMacro@textsubscript\LWR@orig@textsubscript%
6519 \LetLtxMacro\underline\LWR@origunderline%
6520 \let~\LWR@origtilde%
6521 \let\enskip\LWR@origenskip%
6522 \let\quad\LWR@origquad%
6523 \let\qqquad\LWR@origqqquad%
6524 \LetLtxMacro{\tabular}{\LWR@origtabular}%
6525 \LetLtxMacro{\endtabular}{\LWR@origendtabular}%
6526 \LetLtxMacro\toprule\LWR@origtoprule%
6527 \LetLtxMacro\midrule\LWR@origmidrule%
6528 \LetLtxMacro\cmidrule\LWR@origcmidrule%
6529 \LetLtxMacro\bottomrule\LWR@origbottomrule%
6530 \LetLtxMacro\addlinespace\LWR@origaddlinespace%
6531 \LetLtxMacro\morecmidrules\LWR@origmorecmidrules%
6532 \LetLtxMacro\specialrule\LWR@origspecialrule%
6533 \let\newline\LWR@orignewline%
6534 \LetLtxMacro{\raisebox}{\LWR@origraisebox}%
6535 \LetLtxMacro\includegraphics\LWR@origincludegraphics%
6536 \LetLtxMacro{\scalebox}{\LWR@origscalebox}%
6537 \LetLtxMacro{\rotatebox}{\LWR@origrotatebox}%
6538 \let\reflectbox\LWR@origreflectbox%
6539 \LetLtxMacro\resizebox\LWR@origresizebox%
6540 \let\framebox\LWR@origframebox%
6541 \let\makebox\LWR@origmakebox%
6542 \let\fbbox\LWRprint@fbbox%
6543 \let\fbboxBlock\LWRprint@fbbox%
6544 \LetLtxMacro{\fminipage}{\LWRprint@fminipage}%
6545 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}%
```

```

6546 \LetLtxMacro{\minipage}{\LWR@origminipage}%
6547 \let\endminipage\LWR@endminipage%
6548 \LetLtxMacro{\parbox}{\LWR@origparbox}%
6549 \let\TeX\LWR@origTeX%
6550 \let\LaTeX\LWR@origLaTeX%
6551 \let\LaTeXe\LWR@origLaTeXe%
6552 \renewcommand*{\Xe}{X\textsubscript{E}}%

6553 \LetLtxMacro\@ensuredmath\LWR@origensuredmath%

6554 \csletcs{equation*}{LWR@origequationstar}%
6555 \csletcs{endequation*}{LWR@origendequationstar}%
6556 %
6557 \LWR@restoreorigaccents%
6558 %
6559 \LWR@FBCancel%
6560 }

```

## 65 Math

### 65.1 Limitations

#### 65.1.1 Rendering tradeoffs

- Math rendering** Math may be rendered as SVG graphics or using the MATHJAX JavaScript display engine.
- SVG files** In its current implementation, rendering math as images creates a new svg file for each expression. 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 check-summing may be used to remove the need for duplicate files.
- SVG inline** Another approach could be to in-line the svg files directly into the HTML. This may reduce the number of files and potentially speed loading the images, but slows the display of the rest of the document before the images are loaded.
- PNG files** Others converters have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but svg files are the preferred approach for scalable graphics.
- MathML** Conversion to MathML might be a better approach, among other things allowing a

more compact representation of math than svg drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 9 regarding EPUB output with MathJax.

### 65.1.2 SVG option

**SVG math option** For svg math, math is rendered as usual by  $\text{\LaTeX}$  into the initial PDF file using the current font<sup>13</sup>, then is captured from the PDF and converted to svg graphics via a number of utility programs. The svg format is a scalable-vector web format, so math may be typeset by  $\text{\LaTeX}$  with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML alt attribute carries the  $\text{\LaTeX}$  code which generated the math, allowing copy/paste of the  $\text{\LaTeX}$  math expression into other documents.

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

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

### 65.1.3 MathJax option

**MathJax math option** The popular MathJax alternative ([mathjax.org](http://mathjax.org)) may be used to display math.

Prog MathJax

When MathJax is enabled, math is rendered twice:

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

**MathJax limitations** Limitations when using MathJax include:

Prog MathJax

**chapter numbers**

- In document classes which have chapters, `\tagged` equations have the chapter

<sup>13</sup>See section 248 regarding fonts and fractions.



number prepended in HTML output, unlike  $\LaTeX$  `\tag*` equations (correctly) do not. This may be improved with future versions of the MathJax support script.

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

#### subequations

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

#### footnotes in math

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

#### lateximage

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

#### siunitx

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

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

Also see section 8.5.5.

#### siunitx inside an equation

#### other macros and packages

- Other math-related macros and packages are not supported by MathJax, including `\ensuremath`, `bigdelim`, `units`, and `nicefrac`, along with occasionally-used macros such as `\footnote` and `\relax`.

#### custom MathJax macros

- MathJax does not automatically support custom  $\LaTeX$  macros, but they may be created by the user inside a math expression:

```
\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}
```

## 65.2 Inline and display math

for HTML output: `6561 \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.

```
6562 \let\LWR@origtextdollar\$
6563
6564 \renewcommand*{\$}{%
6565 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
6566 {\LWR@origtextdollar}%
6567 {\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
6568 }
```

Ctr `LWR@externalfilecnt` Counter for the external files which are generated and then referenced from the HTML:

```
6569 \newcounter{LWR@externalfilecnt}

6570 \LetLtxMacro\LWR@origdollar$
6571 \LetLtxMacro\LWR@secondorigdollar$% balance for editor syntax highlighting

6572 \LetLtxMacro\LWR@origopenparen\ (
6573 \LetLtxMacro\LWR@origcloseparen\)
6574 \LetLtxMacro\LWR@origopenbracket\[
6575 \LetLtxMacro\LWR@origclosebracket\]
```

`$` Redefine the dollar sign to place math inside a `lateximage`, or use MathJax:  
`$$`

```
6576 \begingroup
6577 \catcode'\$=\active%
6578 \protected\gdef$ {\ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

`\LWR@doubledollar` Redefine the double dollar sign to place math inside a `lateximage`, or use MathJax:

```
6579 \protected\gdef\LWR@doubledollar$#1$${
```

If MathJax or formatting for a word processor, print the  $\LaTeX$  expression:

```
6580 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{%
```

For MathJax, print the math between `\[` and `\]`:

```
6581 {
6582
6583 \textbackslash[
6584 \LWR@HTMLsanitize{#1}%
6585 \textbackslash]
```

```
6586
6587 }% mathjax
```

For SVG, print the math inside a `lateximage`, with an `<alt>` tag of the  $\LaTeX$  code:

```
6588 {% not mathjax
6589
6590     \begin{lateximage}%
6591     [\textbackslash{[] \LWR@HTMLsanitize{#1} \textbackslash{[]}]%
6592     \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
6593     \end{lateximage}%
6594
6595 }% not mathjax
6596 }%
```

`\LWR@singledollar` Redefine the single dollar sign to place math inside a `lateximage`, or use MathJax:

```
6597 \protected\gdef\LWR@subsingledollar#1{%
6598 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

For MathJax, print the math between `\(` and `\)`:

```
6599 {%
6600     {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}%
6601 }% mathjax
```

For SVG, print the math inside a `lateximage`, with an `<alt>` tag of the  $\LaTeX$  code:

```
6602 {% not mathjax
6603     \begin{lateximage}%
6604     [\textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)]%
6605     \LWR@origdollar#1\LWR@origdollar%
6606     \end{lateximage}%
6607 }%
6608 }
6609
6610 \protected\gdef\LWR@singledollar#1${%
6611 \LWR@subsingledollar{#1}%
6612 }
```

`\(` Redefine to the above dollar macros.

`\)`

```
6613 \protected\gdef\(#1\){%
6614 $#1$}
6615 \protected\gdef\[#1\]{$$#1$$}
6616
6617 \endgroup
```

6618  
6619  
6620

`\@ensuredmath`  $\{ \langle expression \rangle \}$  Not yet adapted to lwarp.

```
6621 \LetLtxMacro\LWR@origensuredmath\@ensuredmath
6622
6623 \renewcommand{\@ensuredmath}[1]{%
6624   \ifmmode%
6625     \LWR@origensuredmath{#1}%
6626   \else%
6627     \LWR@subsingledollar{\relax#1}%
6628   \fi%
6629 }
```

Remove the old math and displaymath environments:

```
6630 \let\math\relax
6631 \let\endmath\relax
6632 \let\displaymath\relax
6633 \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`.

```
6634 \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`.

```
6635 \NewEnviron{displaymath}{\expandafter\[\BODY\]\@ignoretrue}
```

### 65.3 MathJax support

Ctrl `LWR@nextequation` Used to add one to compute the next equation number.

```
6636 \newcounter{LWR@nextequation}
```

`\LWR@syncmathjax` Sets the MathJax equation format and number for the following equations.

These MathJax commands are printed inside “\( $\)” and “\( $\)” characters. They are printed to HTML output, not interpreted by  $\text{\LaTeX}$ .$$

```
6637 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
6638 \ifcsdef{thechapter}{
6639 \InlineClass{hidden}{
6640 \textbackslash(
6641 \textbackslash{seteqsection \{\thechapter\}
6642 \textbackslash)
6643 }
6644 }
6645 }}% not using chapters
```

MathJax doesn't allow setting the equation number to 1:

```
6646 \ifthenelse{\cnttest{\value{equation}}>0}
6647 {
```

Tell MathJax that the next set of equations begins with the current  $\TeX$  equation number, plus one.

```
6648 \setcounter{LWR@nextequation}{\value{equation}}
6649 \addtocounter{LWR@nextequation}{1}
```

Place the MathJax command inside “\ (“ and “\)” characters, to be printed to HTML, not interpreted by  $\TeX$ .

```
6650 \InlineClass{hidden}{
6651 \textbackslash(
6652 \textbackslash{seteqnumber \{\arabic{LWR@nextequation}\}
6653 \textbackslash)
6654 }
6655 }}% not eq > 0
6656 }
```

```
\LWR@hidelatexequation {\langle environment \rangle} {\langle contents \rangle}
```

Creates the  $\TeX$  version of the equation inside an HTML comment.

```
6657 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```
6658 \LWR@stoppars
6659 \LWR@htmlopencomment
6660
```

Start the  $\text{\LaTeX}$  math environment inside the HTML comment:

```
6661 \begingroup
6662 \csuse{LWR@orig#1}
```

While in the math environment, restore various commands to their  $\text{\LaTeX}$  meanings.

```
6663 \LWR@restoreorigformatting
```

See `\LWR@htmlmathlabel` in section 65.5.1.

Print the contents of the equation:

```
6664 #2
```

End the  $\text{\LaTeX}$  math environment inside the HTML comment:

```
6665 \csuse{LWR@origend#1}
6666 \endgroup
6667
```

Close the HTML comment and resume HTML paragraph handling:

```
6668 \LWR@htmlclosecomment
6669 \LWR@startpars
6670 }
```

```
\LWR@addmathjax {<environment>}{<contents>}
```

Given the name of a math environment and its contents, create a MathJax instance. The contents are printed to HTML output, not interpreted by  $\text{\LaTeX}$ .

```
6671 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
```

Enclose the MathJax environment inside printed “\ (“ and “\)” characters.

```
6672 \LWR@origtilde\LWR@orignewline
6673 \textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
6674 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
6675 \textbackslash{}end\{#1\}
6676 \LWR@orignewline
6677 }
```

## 65.4 Equation environment

Remember existing equation environment:

```
6678 \let\LWR@origequation\equation
6679 \let\LWR@origendequation\endequation
```

Remove existing equation environment:

```
6680 \let\equation\relax
6681 \let\endequation\relax
```

Env `equation` The new equation environment is created with `\NewEnviron` (from the `environ` package), which stores the contents of its environment in a macro called `\BODY`.

For SVG math output, the contents are typeset using the original equation inside a `lateximage`, along with an `<alt>` tag containing a detokenized copy of the  $\LaTeX$  source for the math.

For MathJax output, the contents are typeset in an original equation environment placed inside a HTML comment, with special processing for `\labels`. The contents are also printed to the HTML output for processing by the MathJax script.

```
6682 \NewEnviron{equation}{%
6683
```

If `mathjax` or `FormatWP`, print the  $\LaTeX$  expression:

```
6684 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{%
```

MathJax output:

```
6685 {
```

Print commands to synchronize MathJax's equation number and format to the current  $\LaTeX$  chapter/section and equation number:

```
6686 \LWR@syncmathjax
```

Print the  $\LaTeX$  math inside an HTML comment:

```
6687 \LWR@hidelatexequation{equation}{\BODY}
6688 }
```

SVG output: Create the `lateximage` along with an HTML `<alt>` tag having an equation number, the  $\LaTeX$  equation environment commands, and the contents of the environment's `\BODY`.

```
6689 {% not mathjax
```

Begin the `lateximage` with an `<alt>` tag containing the math source:

```
6690 \begin{lateximage}[({\LWR@equationtag} \textbackslash{begin}\{equation\})] %
6691 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}} %
6692 \textbackslash{end}\{equation\}}]% alt tag
```

Create the actual  $\LaTeX$ -formatted equation inside the `lateximage` using the contents of the environment.

```
6693 \LWR@origequation
6694 \BODY% contents collected by NewEnviron
6695 \LWR@origendequation
6696 \end{lateximage}%
6697}% not mathjax
6698
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
6699 }[%
6700 \ifbool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) ]%
6701 {%
6702 \LWR@addmathjax{equation}{\BODY}%
6703 }{}%
6704
6705 ]
```

Env `equation*`

```
6706 \csletcs{\LWR@origequationstar}{equation*}
6707 \csletcs{\LWR@origendequationstar}{endequation*}
6708 \renewenvironment{equation*}
6709 {\displaymath}
6710 {\enddisplaymath}
```



## 65.5 AMS Math environments

### 65.5.1 Support macros

Bool LWR@amsmultline True if processing a multiline environment.

To compensate for multiline-specific code, LWR@amsmultline is used to add extra horizontal space in \LWR@htmlmathlabel if is used in an amsmath environment which is not a multiline environment and not an equation.

```
6711 \newbool{LWR@amsmultline}
6712 \boolfalse{LWR@amsmultline}
```

\LWR@htmlmathlabel {<label>}

lwarp points \ltx@label here. This is used by \label when inside a  $\TeX$  AMS math environment's math display environment.

\LWR@origltx@label points to the  $\TeX$  original, modified by lwarp, then by amsmath, then by cleveref.

```
6713 \newcommand*{\LWR@htmlmathlabel}[1]{%
6714 \LWR@traceinfo{LWR@htmlmathlabel #1}%
```

If mathjax or FormatWP, print the  $\TeX$  expression:

```
6715 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{%
6716 {%
```

The combined  $\TeX$  & HTML label is printed in a \text field:

```
6717 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
6718 \ifbool{LWR@amsmultline}{\hspace*{\totwidth@}}
```

Temporarily end the HTML comment, insert the  $\TeX$  & HTML label, then resume the HTML comment. \@firstofone is required to remove extra braces introduced by the amsmath package.)

```
6719 \LWR@htmlclosecomment%
6720 \LWR@origltx@label{#1}%
6721 \LWR@htmlopencomment%
6722 }% text
```

```

6723 }% mathjax
6724 {%
6725     \LWR@origltx@label{#1}%
6726 }%
6727 }

```

`\LWR@beginhideamsmath` Starts hiding  $\text{\LaTeX}$  math inside an HTML comment.

```

6728 \newcommand*{\LWR@beginhideamsmath}{
6729 \LWR@stoppars
6730 \LWR@origtilde\LWR@orignewline
6731 \LWR@htmlopencomment
6732
6733 \begingroup
6734 \LWR@restoreorigformatting
6735 }

```

`\LWR@endhideamsmath` Ends hiding  $\text{\LaTeX}$  math inside an HTML comment.

```

6736 \newcommand*{\LWR@endhideamsmath}{
6737 \endgroup
6738
6739 \LWR@htmlclosecomment
6740 \LWR@orignewline
6741 \LWR@startpars
6742 }

```

### 65.5.2 Environment patches

The following amsmath environments already collect their contents in `\@envbody` for further processing. `eqnarray` is not an  $\text{\LaTeX}$  package, and thus requires special handling.

For `svg math`: Each environment is encapsulated inside a `lateximage` environment, along with a special optional argument of `\LWR@amsmathbody` or `\LWR@amsmathbodynumbered` telling `lateximage` to use as the HTML `<alt>` tag the environment's contents which were automatically captured by the  $\text{\LaTeX}$  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 `eqnarray` This environmnet is not an  $\text{\LaTeX}$  environment and thus its body is not automatically captured, so the `environ` package is used to capture the environment into `\BODY`.

```
6743 \let\LWR@origeqnarray\eqnarray
6744 \let\LWR@origendeqnarray\endeqnarray
```

To remember whether the starred environment was used, and thus whether to number the equations:

```
6745 \newbool{LWR@numbereqnarray}
6746 \booltrue{LWR@numbereqnarray}
```

Common code used by eqnarray and Beqnarray (from fancybox):

```
6747 \newcommand{\LWR@eqnarrayfactor}{%
```

If mathjax or FormatWP, print the  $\TeX$  expression:

```
6748 \ifbool{expr}{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6749 {%
```

If MathJax, the environment contents (the `\BODY`) are executed in a HTML comment to trigger the correct equation number increment (if not starred), then are included verbatim in the output for MathJax to interpret:

```
6750 \LWR@syncmathjax
6751 \boolfalse{LWR@amsmultline}
6752 \ifbool{LWR@numbereqnarray}
6753 {
```

If numbering the equations, execute a copy inside an HTML comment block:

```
6754 \LWR@beginhideamsmath
6755 \LWR@origeqnarray
6756 \BODY
6757 \LWR@origendeqnarray
6758 \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MathJax to interpret:

```
6759 \LWR@addmathjax{eqnarray}{\BODY}
6760 }%
6761 {% not LWR@numbereqnarray
```

If not numbering equations, just create the contents for MathJax:

```
6762 \LWR@addmathjax{eqnarray*}{\BODY}
6763 }% LWR@numbereqnarray
6764 }% mathjax
6765 {% not mathjax
6766 \ifbool{LWR@numbereqnarray}
```

6767     {

For numbered svg equations, first create a lateximage with an alt attribute containing sanitized copy of the source code:

```
6768             \begin{lateximage}[(\LWR@startingequationtag--\LWR@equationtag)
6769                 \LWR@addmathjax{eqnarray}{\BODY}]
```

Then create the image contents using an actual eqnarray:

```
6770             \LWR@origeqnarray
6771             \BODY
6772             \LWR@origendeqnarray
6773             \end{lateximage}
6774     }%
6775     {% not LWR@numbereqnarray
```

If not numbered, do the same, but an extra \nonumber seems to be required:

```
6776             \begin{lateximage}[(\LWR@addmathjax{eqnarray*}{\BODY}]
6777             \LWR@origeqnarray
6778             \BODY
6779             \nonumber
6780             \LWR@origendeqnarray
6781             \end{lateximage}
6782     }% LWR@numbereqnarray
6783 }% not mathjax
```

Default to number equations in the future:

```
6784 \booltrue{LWR@numbereqnarray}
6785 }
```

eqnarray itself is made with a blank line before and after to force it to be on its own line:

```
6786 \RenewEnviron{eqnarray}
6787 {%
6788
6789 \LWR@eqnarrayfactor
6790
6791 }
```

The starred version is patched to turn off the numbering:

```
6792 \csgpretoto{eqnarray*}{\boolfalse{LWR@numbereqnarray}}
```

The following  $\mathcal{MS}$  environments are more easily patched in-place:

Env `multline`

```

6793 \BeforeBeginEnvironment{multline}{
6794
6795 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6796 {
6797     \LWR@syncmathjax
6798     \booltrue{LWR@amsmultline}
6799     \LWR@beginhideamsmath
6800 }
6801 {
6802     \lateximage[\LWR@amsmathbodynumbered{multline}]
6803 }
6804 }
6805
6806 \AfterEndEnvironment{multline}{
6807
6808 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6809 {
6810     \LWR@endhideamsmath
6811     \boolfalse{LWR@amsmultline}
6812     \LWR@addmathjax{multline}{\the\@envbody}
6813 }
6814 {\endlateximage}
6815
6816 }
```

Env `multline*`

```

6817 \BeforeBeginEnvironment{multline*}{
6818
6819 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6820 {
6821     \LWR@syncmathjax
6822     \booltrue{LWR@amsmultline}
6823     \LWR@beginhideamsmath
6824 }
6825 {
6826     \lateximage[\LWR@amsmathbody{multline*}]
6827 }
6828 }
6829
6830 \AfterEndEnvironment{multline*}{
6831
6832 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6833 {
6834     \LWR@endhideamsmath
6835     \boolfalse{LWR@amsmultline}
6836     \LWR@addmathjax{multline*}{\the\@envbody}

```

```

6837 }
6838 {\endlateximage}
6839
6840 }
6841

```

Env **gather**

```

6842 \BeforeBeginEnvironment{gather}{
6843
6844 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6845 {
6846   \LWR@syncmathjax
6847   \boolfalse{LWR@amsmultline}
6848   \LWR@beginhideamsmath
6849 }
6850 {
6851   \lateximage[\LWR@amsmathbodynumbered{gather}]
6852 }
6853 }
6854
6855 \AfterEndEnvironment{gather}{
6856
6857 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6858 {
6859   \LWR@endhideamsmath
6860   \LWR@addmathjax{gather}{\the\@envbody}
6861 }
6862 {\endlateximage}
6863
6864 }

```

Env **gather\***

```

6865 \BeforeBeginEnvironment{gather*}{
6866
6867 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6868 {
6869   \LWR@syncmathjax
6870   \boolfalse{LWR@amsmultline}
6871   \LWR@beginhideamsmath
6872 }
6873 {
6874   \lateximage[\LWR@amsmathbody{gather*}]
6875 }
6876 }
6877
6878 \AfterEndEnvironment{gather*}{

```

```

6879
6880 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6881 {
6882   \LWR@endhideamsmath
6883   \LWR@addmathjax{gather*}{\the\@envbody}
6884 }
6885 {\endlateximage}
6886
6887 }

```

Env align

```

6888 \BeforeBeginEnvironment{align}{
6889
6890 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6891 {
6892   \LWR@syncmathjax
6893   \boolfalse{LWR@amsmultline}
6894   \LWR@beginhideamsmath
6895 }
6896 {
6897   \lateximage[\LWR@amsmathbodynumbered{align}]
6898 }
6899 }
6900
6901 \AfterEndEnvironment{align}{
6902
6903 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6904 {
6905   \LWR@endhideamsmath
6906   \LWR@addmathjax{align}{\the\@envbody}
6907 }
6908 {\endlateximage}
6909
6910 }

```

Env align\*

```

6911 \BeforeBeginEnvironment{align*}{
6912
6913 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6914 {
6915   \LWR@syncmathjax
6916   \boolfalse{LWR@amsmultline}
6917   \LWR@beginhideamsmath
6918 }
6919 {
6920   \lateximage[\LWR@amsmathbody{align*}]

```

```

6921 }
6922 }
6923
6924 \AfterEndEnvironment{align*}{
6925
6926 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6927 {
6928     \LWR@endhideamsmath
6929     \LWR@addmathjax{align*}{\the\@envbody}
6930 }
6931 {\endlateximage}
6932
6933 }

```

Env **flalign**

```

6934 \BeforeBeginEnvironment{flalign}{
6935
6936 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6937 {
6938     \LWR@syncmathjax
6939     \boolfalse{LWR@amsmultline}
6940     \LWR@beginhideamsmath
6941 }
6942 {
6943     \lateximage[\LWR@amsmathbodynumbered{flalign}]
6944 }
6945 }
6946
6947 \AfterEndEnvironment{flalign}{
6948
6949 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6950 {
6951     \LWR@endhideamsmath
6952     \LWR@addmathjax{flalign}{\the\@envbody}
6953 }
6954 {\endlateximage}
6955
6956 }

```

Env **flalign\***

```

6957 \BeforeBeginEnvironment{flalign*}{
6958
6959 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6960 {
6961     \LWR@syncmathjax
6962     \boolfalse{LWR@amsmultline}

```



```

6963     \LWR@beginhideamsmath
6964 }
6965 {
6966     \lateximage[\LWR@amsmathbody{flalign*}]
6967 }
6968 }
6969
6970 \AfterEndEnvironment{flalign*}{
6971
6972 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6973 {
6974     \LWR@endhideamsmath
6975     \LWR@addmathjax{flalign*}{\the\@envbody}
6976 }
6977 {\endlateximage}
6978
6979 }

6980 \end{warpHTML}

```

## 66 Lateximages

### 66.1 Description

**Env lateximage** A lateximage is a piece of the document which is typeset in  $\text{\LaTeX}$  then included in the HTML output as an image. This is used for math if svg math is chosen, and also for the picture, tikzpicture, and other environments.

Before typesetting the lateximage a large number of formatting, graphics, and symbols-related macros are temporarily restored to their print-mode meaning by `\LWR@restoreorigformatting`. (See section 64.)

A lateximage is typeset on its own PDF page inside an HTML comment which starts on the preceeding page and ends on following page, and instructions are written to lateximage.txt for lwarpmk to extract the `\lateximage` from the page of the PDF file then generate an accompanying .svg file image file. Meanwhile, instructions to show this image are placed into the HTML file after the comment.

An HTML `<span>` is created to hold both the HTML comment, which will have the pdf tototext conversion, and also the link to the final .svg image.

A  $\text{\LaTeX}$  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}`

## 66.2 Support counters and macros

**for HTML output:** 6981 `\begin{warpHTML}`

Ctr `LWR@lateximagenumber` Sequence the images.

```
6982 \newcounter{LWR@lateximagenumber}
6983 \setcounter{LWR@lateximagenumber}{0}
```

Ctr `LWR@lateximagedepth` Do not create `\lateximage` inside of `\lateximage`.

```
6984 \newcounter{LWR@lateximagedepth}
6985 \setcounter{LWR@lateximagedepth}{0}
```

A few utility macros to write special characters:

```
6986 \edef\LWR@hashmark{\string#} % for use in \write
6987 \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`.


```
6988 \newcounter{LWR@LIpage}

6989 \end{warpHTML}
```

## 66.3 Font size

**for HTML & PRINT:** 6990 `\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.

 **no backslash** Do not include the leading backslash in the name.

```
6991 \newcommand*{\LateximageFontSizeName}{large}

6992 \end{warpall}
```

## 66.4 Sanitizing math expressions for HTML

for HTML output: 6993 `\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...

```
6994 \newcommand{\LWR@HTMLsanitize}[1]{%
6995 \begingroup%
6996 \LWR@FBcancel%
6997 \protect\StrSubstitute{\detokenize{#1}}%
6998 {\detokenize{&}}%
6999 {\detokenize{&}}[\LWR@strresult]%
7000 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
7001 {\detokenize{<}}%
7002 {\detokenize{<}}[\LWR@strresult]%
7003 [\LWR@strresult]%
7004 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
7005 {\detokenize{>}}%
7006 {\detokenize{>}}[\LWR@strresult]%
7007 [\LWR@strresult]%
7008 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
7009 {\detokenize{##}}%
7010 {\detokenize{##}}[\LWR@strresult]%
7011 [\LWR@strresult]%
7012 \LWR@strresult%
7013 \endgroup%
7014 }
```

`\LWR@HTMLsanitizeexpand`  $\{\langle text \rangle\}$

This version expands the argument before sanitizing it.

```
7015 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
7016 \begingroup%
7017 \LWR@FBcancel%
7018 \protect\StrSubstitute{\detokenize\expandafter{#1}}%
7019 {\detokenize{&}}%
7020 {\detokenize{&}}[\LWR@strresult]%
7021 [\LWR@strresult]%
7022 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
7023 {\detokenize{<}}%
7024 {\detokenize{<}}[\LWR@strresult]%
7025 [\LWR@strresult]%
7026 \LWR@strresult%
7027 \endgroup%
7028 }
```

```

7025 [\LWR@strresult]%
7026 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}}%
7027 {\detokenize{>}}}%
7028 {\detokenize{&gt;}}}%
7029 [\LWR@strresult]%
7030 \LWR@strresult%
7031 \endgroup%
7032 }

```

## 66.5 Equation numbers

**Ctrl** **LWR@startingequation** For use with lateximage and multi-line numbered equations. Remembers the next equation number so that it may be printed in the alt tag.

```

7033 \newcounter{LWR@startingequation}
7034
7035 \@ifundefined{chapter}
7036 {
7037 \renewcommand{\theLWR@startingequation}{%
7038 \arabic{LWR@startingequation}}%
7039 }
7040 }
7041 {% chapter defined
7042 \renewcommand{\theLWR@startingequation}{%
7043 \ifnumcomp{\value{chapter}}{>}{0}{\arabic{chapter}.}{}}%
7044 \arabic{LWR@startingequation}}%
7045 }
7046 }

```

**Bool** **LWR@isstartingequation** True for the first equation tag, false for later tags in the same environment.

```

7047 \newbool{LWR@isstartingequation}

```

**\LWR@startingequationtag** Prints the starting equation number or tag.

```

7048 \let\LWR@startingequationtag\theLWR@startingequation

```

**\LWR@equationtag** Prints the ending equation number or tag.

```

7049 \let\LWR@equationtag\theequation

```

Only if svg math, patch \tag after packages have loaded, in case someone else modified \tag.

```

7050 \AtBeginDocument{
7051
7052 \ifbool{mathjax}{}{% not mathjax

```

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

For use inside the math environments while using svg math. Sets `\theLWR@startingequation` and `\theequation` to the given tag.

```

7053 \NewDocumentCommand{\LWR@remembertag}{m}{%
7054 \ifbool{LWR@isstartingequation}%
7055 {%
7056   \global\boolfalse{LWR@isstartingequation}%
7057   \xdef\LWR@startingequationtag{#1}%
7058 }%
7059 }%
7060 \xdef\LWR@equationtag{#1}%
7061 }%

```

Patches for  $\mathcal{AMS}$  math `\tag` macro to remember the first tag:

```

7062 \LetLtxMacro\LWR@origmake@df@tag@@\make@df@tag@@
7063 \LetLtxMacro\LWR@origmake@df@tag@@@\make@df@tag@@@
7064
7065 \renewcommand*\make@df@tag@@[1]{%
7066 \LWR@remembertag{#1}%
7067 \LWR@origmake@df@tag@@{#1}%
7068 }
7069
7070 \renewcommand*\make@df@tag@@@[1]{%
7071 \LWR@remembertag{#1}%
7072 \LWR@origmake@df@tag@@@{#1}%
7073 }
7074
7075 }% not mathjax
7076 }% AtBeginDocument

```

## 66.6 HTML `<alt>` tags

`\LWR@amsmathbody`  $\{\langle envname \rangle\}$  For use inside the optional argument to a `lateximage` to add the contents of a AMS math environment to the `<alt>` tag.

```

7077 \newcommand*\LWR@amsmathbody[1]
7078 {%
7079 \textbackslash\begin\{\{#1\} %

```

```

7080 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}}%
7081 \textbackslash\end\}\{#1\}%
7082 }

```

`\LWR@amsmathbodynumbered` `{\envname}` For use inside the optional argument to a `lateximage` to add the contents of a AMS math environment to the `<alt>` tag, prefixed by the equation numbers.

```

7083 \newcommand*{\LWR@amsmathbodynumbered}[1]
7084 {%
7085 \ifnumcomp{\value{LWR@startingequation}}{=}{\value{equation}}}%
7086 {(\LWR@equationtag)}%
7087 {(\LWR@startingequationtag--\LWR@equationtag)} %
7088 \LWR@amsmathbody{#1} %
7089 }

```

## 66.7 lateximage

Env `lateximage` [`<alt> tag`]

```

7090 \catcode'\$=\active%
7091
7092
7093 \NewDocumentEnvironment{lateximage}{0{image}}
7094 {%
7095 \LWR@traceinfo{lateximage: starting on page \arabic{page}}}%

```

Nested `lateximages` remain one large `lateximage`:

```

7096 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%

```

If nesting inside an already-existing `lateximage`, simply record one more level:

```

7097 {%
7098   \addtocounter{LWR@lateximagedepth}{1}%
7099 }%

```

Otherwise, this is the outer-most `lateximage`:

```

7100 {% start of outer-most lateximage

```

Remember the next equation number to be allocated, in case it must be printed in a multi-equation environment:

```

7101 \setcounter{LWR@startingequation}{\value{equation}}%
7102 \addtocounter{LWR@startingequation}{1}%
7103 \booltrue{LWR@isstartingequation}%
7104 \let\LWR@startingequationtag\theLWR@startingequation%
7105 \let\LWR@equationtag\theequation%

```

Starting a new lateximage:

```

7106 \addtocounter{LWR@lateximagenumber}{1}%
7107 \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%

```

While inside a lateximage, locally do not use mathjax:

```

7108 \boolfalse{mathjax}

```

Be sure that are doing a paragraph:

```

7109 \LWR@ensuredoingapar%

```

Next file:

```

7110 \addtocounter{LWR@externalfilecnt}{1}%
7111 \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%

```

Figure out what the next page number will be:

```

7112 \setcounterpageref{LWR@LIPage}{LWR@lateximage\arabic{LWR@lateximagenumber}}%
7113 \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:

```

7114 \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
7115 class="lateximagesource"{} } \LWR@orignewline

```

Write instructions to the lateximages.txt file:

```

7116 \LWR@traceinfo{lateximage: about to write to lateximages.txt}%
7117 \immediate\write\LWR@lateximagesfile{|\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.

```

7118 \LWR@traceinfo{lateximage: about to create open comment}%
7119 \LWR@htmlopencomment%

```

One level deeper:

```
7120 \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
7121 \LWR@traceinfo{lateximage: about to create a new page}%
7122 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
7123 \LWR@traceinfo{lateximage: about to create minipage}%
7124 \LWR@origminipage{6in}%
7125 \csuse{LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for `\hspace`, etc. inside a `lateximage`.

```
7126 \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
7127 \LWR@restoreorigformatting%
```

Use full-page footnotes instead of minipage footnotes. These become HTML footnotes.

```
7128 \def\@mpfn{footnote}%
7129 \def\thempfn{\thefootnote}%
7130 \let\@footnotetext\LWR@footnotetext%
```

Create the `LWRlateximage<number>` label:

```
7131 \LWR@traceinfo{lateximage: about to create label}%
7132 \LWR@origlabel{LWRlateximage\arabic{LWR@lateximagenumber}}%
7133 \LWR@traceinfo{lateximage: finished creating the label}%
```

Enable print-mode math functions:

```
7134 \LetLtxMacro$\LWR@origdollar%
7135 \catcode'\$=3% math shift
7136 \LetLtxMacro\(\LWR@origopenparen%
7137 \LetLtxMacro\)\LWR@origcloseparen%
7138 \LetLtxMacro\[ \LWR@origopenbracket%
7139 \LetLtxMacro\] \LWR@origclosebracket%
7140}% end of outer-most lateximage
7141 \LWR@traceinfo{lateximage: finished start of environment}%
7142}% end of \begin{lateximage}
```

`\endlateximage` When the environment closes:



```

7143 {% start of \end{lateximage}
7144 \LWR@traceinfo{lateximage: starting end of environment}%

```

Nested more than one deep?

```

7145 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%

```

If nesting inside an already-existing lateximage, simply record one more level:

```

7146 {\addtocounter{LWR@lateximagedepth}{-1}}%

```

If this is the outer-most lateximage:

```

7147 {% end of outer-most lateximage

```

Finish the lateximage minipage and start a new PDF page:

```

7148 \LWR@origendminipage%
7149 \LWR@orignewpage%
7150 \LWR@origscriptsize%

```

Close the HTML comment which encapsulated any traces of the lateximage picked up by pdftotext:

```

7151 \LWR@htmlclosecomment{}\LWR@orignewline%
7152 \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}%

```

Create a link to the lateximage, allowing its natural height:

```

7153 \LWR@subinlineimage[#1]{lateximage}%
7154 {lateximages\OSPathSymbol{}{lateximage-\theLWR@externalfilecnt}{svg}}%

```

Be sure that are doing a paragraph:

```

7155 \LWR@ensuredoingapar%

```

Close the HTML span which has the pdftotext comment and also the link to the .svg image:

```

7156 \LWR@htmltag{/span}%
7157 \ifbool{HTMLDebugComments}{%
7158 \LWR@htmlcomment{End of lateximage}%
7159 }{}%
7160 % \LWR@orignewline% Removed to prevent extra space.

```

Undo one lateximage level:

```

7161 \addtocounter{LWR@lateximagedepth}{-1}%
7162 }% end of outer-most lateximage
7163 \LWR@traceinfo{lateximage: done}%
7164 }%
7165 \catcode'\$=3% math shift
7166 \end{warpHTML}

```

**for PRINT output:**

```

7167 \begin{warpprint}
7168 % \newenvironment{lateximage}[1][\minipage{\linewidth}]{\endminipage}
7169 \newenvironment{lateximage}[1][\{}{\}}
7170 \end{warpprint}

```

## 67 center, flushleft, flushright

**for HTML output:**

```

7171 \begin{warpHTML}

```

Env **center** Replace center functionality with CSS tags:

```

7172 \renewenvironment*{center}
7173 {
7174 \LWR@forcenewpage
7175 \ifbool{FormatWP}
7176 {\BlockClass[text-align:center]{center}}
7177 {\BlockClass{center}}
7178 }
7179 {\endBlockClass}

```

Env **flushright**

```

7180 \renewenvironment*{flushright}
7181 {
7182 \LWR@forcenewpage
7183 \ifbool{FormatWP}
7184 {\BlockClass[text-align:right]{flushright}}
7185 {\BlockClass{flushright}}
7186 }
7187 {\endBlockClass}

```

Env **flushleft**

```

7188 \renewenvironment*{flushleft}
7189 {
7190 \LWR@forcenewpage
7191 \ifbool{FormatWP}

```

```

7192 {\BlockClass[text-align:left]{flushleft}}
7193 {\BlockClass{flushleft}}
7194 }
7195 {\endBlockClass}

7196 \end{warpHTML}

```

## 68 Pre-loaded packages

**for HTML output:** 7197 \begin{warpHTML}

If textcomp was loaded before lwarp, perhaps as part of the font-related packages, explicitly load the lwarp patches now:

```

7198 \@ifpackageloaded{textcomp}
7199 {
7200 \LWR@origRequirePackage{lwarp-textcomp}
7201 }
7202 {}

```

If graphics or graphicx were loaded before lwarp, perhaps by xunicode, explicitly load the lwarp patches now:

```

7203 \@ifpackageloaded{graphics}
7204 {
7205 \LWR@origRequirePackage{lwarp-graphics}
7206 }
7207 {}

```


```


7208 \end{warpHTML}

```

## 69 Siunitx

**Pkg** siunitx The lwarp core passes a few options to siunitx.

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

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

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

for HTML output: 7209 \begin{warphTML}

Options for siunitx:

```
7210 \PassOptionsToPackage{
7211     detect-mode=true,
7212     per-mode=symbol,% fraction is not seen by pdftotext
7213 %     text-celsius = {\protect\LWRsiunitx@degree{}}C},
7214 %     text-degree = {\protect\LWRsiunitx@degree},
7215 }{siunitx}
7216

7217 \end{warphTML}
```

## 70 Graphics print-mode modifications

### 70.1 General limitations

- ⚠ .pdf image files For \includegraphics with .pdf files, the user should provide a .pdf image file, and also a .svg, .png, or .jpg version of the same image. **These should be referred to without a file extension:**
- ⚠ no file extension

```
\includegraphics{filename} % print:.pdf, HTML:.svg or other
```

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

If a .pdf file is explicitly referred to with its file extension, a link to the .pdf file will appear in the HTML output.

```
\includegraphics{filename.pdf} % creates a link in HTML
```

- other image files For .png, .jpg, or .gif image files, the same file may be used in both print or HTML versions, and may be used with a file extension, but will also be used without the file extension if it is the only file of its base name.

- ⚠ graphics vs. graphicx If using the older graphics syntax, use both optional arguments for \includegraphics. A single optional parameter is interpreted as the newer graphicx syntax. Note that viewports are not supported by warp; the entire image will be shown.


⚠ viewports

- units For \includegraphics, avoid px and % units for width and height, or enclose them inside warphTML environments. For font-proportional image sizes, use ex or em. For fixed-sized images, use cm, mm, in, pt, or pc. Use the keys width=.5\linewidth, or similar for \textwidth or \textheight to give fixed-sized images proportional to a 6 by 9 inch text area.

**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.

**`\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  $\TeX$ , so expect some ugly results for scaling and rotating.

## 70.2 Print-mode modifications

**for PRINT output:** For print output, accept and then discard the new class key:

```
7218 \begin{warpprint}
7219 \define@key{Gin}{class}{}

```

Print-mode additions for the overpic package. See section 190 for the HTML version.

```
7220 \AtBeginDocument{
7221 \ifpackageloaded{overpic}{
7222 \newcommand*\overpicfontsize{12}
7223 \newcommand*\overpicfontskip{14}
7224 }{}
7225 }
7226 \end{warpprint}

```

## 71 Xcolor boxes

**Pkg xcolor** A few new definitions are provided for enhanced HTML colored boxes, and `\fcolorbox` is slightly modified. Print-mode version are also provided.

Print-mode versions of new xcolor defintions. These are defined inside `warppall` because they are also used for HTML while inside a `lateximage`. They are defined `\AtBeginDocument` so that the xcolor originals may first be loaded and saved for reuse.

The framed versions are modified to allow a background color of none, in which case only the frame is drawn, allowing the background page color to show.

**for HTML & PRINT:** 7227 `\begin{warppall}`

After xparse may have been loaded ...

```
7228 \AtBeginDocument{
```

... and *only* if xcolor was loaded:

```
7229 \@ifpackageloaded{xcolor}{
7230 \LWR@traceinfo{patching xcolor}
```

`\colorboxBlock` `\colorboxBlock` is the same as `\colorbox`:

```
7231 \LetLtxMacro\colorboxBlock\colorbox
```

In HTML mode, the following is done when xcolor is loaded. Following is the print-mode action:

```
7232 \warpprintonly{
7233 \LetLtxMacro\LWRprint@colorboxBlock\colorbox
7234 \LetLtxMacro\LWRorigprint@fcolorbox\colorbox
7235 \LetLtxMacro\LWRorigprint@fcolorboxBlock\colorbox
7236 }
```

`\fcolorbox` [*framemodel*] [*framecolor*] [*boxmodel*] [*boxcolor*] [*text*]

In print mode, `\fcolorbox` is modified to accept a background color of none.

(`\fcolorbox` is particular about its optional arguments, thus the elaborate combinations of `\ifthenelse`.)

```
7237 \newsavebox{\LWR@colorminipagebox}
7238
7239 \DeclareDocumentCommand{\LWRprint@fcolorbox}{o m o m +m}{%
7240 \LWR@traceinfo{\LWRprint@fcolorbox #2 #4}%
```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
7241 \begin{lrbox}{\LWR@colorminipagebox}%
7242 #5%
7243 \end{lrbox}%
```

Sort out the various optional arguments and the background color of none. In each case, the LRbox is placed inside a `\fcolorbox`.

```
7244 \ifthenelse{\equal{#4}{none}}{%
7245 {% #4 none
7246 \LWR@traceinfo{background is none}%
7247 {% scope the \colorlet
```

```

7248      \colorlet{LWR@currentcolor}{.}%
7249      \color{#2}%
7250      \fbox{%
7251          \color{LWR@currentcolor}%
7252          \usebox{\LWR@colorminipagebox}%
7253      }% fbox
7254  }% colorlet
7255 }% #4 none
7256 {% #4 not none
7257 \LWR@traceinfo{background not none}%
7258 \IfValueTF{#1}%
7259 {%
7260     \IfValueTF{#3}%
7261     {\LWRorigprint@fcolorbox[#1]{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
7262     {\LWRorigprint@fcolorbox[#1]{#2}[#4]{\usebox{\LWR@colorminipagebox}}}%
7263 }%
7264 {% no value #1
7265     \IfValueTF{#3}%
7266     {\LWRorigprint@fcolorbox[#2][#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
7267     {\LWRorigprint@fcolorbox[#2][#4]{\usebox{\LWR@colorminipagebox}}}%
7268 }% no value #1
7269 }% #4 not none
7270 \LWR@traceinfo{LWRprint@fcolorbox done}%
7271 }

```

`\fcolorboxBlock` [*⟨framemodel⟩*] [*⟨framecolor⟩*] [*⟨boxmodel⟩*] [*⟨boxcolor⟩*] [*⟨text⟩*]

In print mode, `\fcolorboxBlock` is the same as `\fcolorbox`.

```
7272 \LetLtxMacro\LWRprint@fcolorboxBlock\LWRprint@fcolorbox
```

Env `fcolorminipage` [*⟨1:framemodel⟩*] [*⟨2:framecolor⟩*] [*⟨3:boxmodel⟩*] [*⟨4:boxcolor⟩*] [*⟨5:align⟩*] [*⟨6:height⟩*] [*⟨7:inner-align⟩*] [*⟨8:width⟩*]

In print mode, becomes a `\fcolorbox` containing a minipage:

```

7273 \NewDocumentEnvironment{LWRprint@fcolorminipage}{o m o m O{c} O{ } o m}
7274 {%
7275 \LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%

```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
7276 \begin{lrbox}{\LWR@colorminipagebox}%
```

If inner alignment is not given, use the outer alignment instead:

```

7277 \IfValueTF{#7}%
7278 {\begin{minipage}[#5][#6][#7]{#8}}%

```

```

7279 {\begin{minipage}[#5][#6][#5]{#8}}%
7280 }%
7281 {%
7282 \end{minipage}%
7283 \end{lrbox}%
7284 \LWR@traceinfo{*** starting end fcolorminipage #1 #2 #3 #4 #8}%

```

Sort out the various optional arguments and the background color of none. In each case, the LRbox is placed inside a `\fcolorbox`.

```

7285 \ifthenelse{\equal{#4}{none}}%
7286 {% #4 none
7287   {% scope the \colorlet
7288     \colorlet{\LWR@currentcolor}{.}%
7289     \color{#2}%
7290     \fbox{%
7291       \color{\LWR@currentcolor}%
7292       \usebox{\LWR@colorminipagebox}%
7293     }% fbox
7294   }% colorlet
7295 }% #4 none
7296 {% #4 not none
7297   \IfValueTF{#1}%
7298   {%
7299     \IfValueTF{#3}%
7300     {\LWR@origprint@fcolorbox[#1]{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
7301     {\LWR@origprint@fcolorbox[#1]{#2}[#4]{\usebox{\LWR@colorminipagebox}}}%
7302   }%
7303   {% no value #1
7304     \IfValueTF{#3}%
7305     {\LWR@origprint@fcolorbox{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
7306     {\LWR@origprint@fcolorbox{#2}[#4]{\usebox{\LWR@colorminipagebox}}}%
7307   }% no value #1
7308 }% #4 not none
7309 \LWR@traceinfo{*** finished end fcolorminipage}%
7310 }

```

`\LWR@restoreorigprintxcolor` Used to activate print-mode additions for xcolor. In print mode, this is used immediately following. In HTML mode, this is used inside a `lateximage`.

```

7311 \newcommand*{\LWR@restoreorigprintxcolor}{%
7312 \LWR@traceinfo{\LWR@restoreorigprintxcolor}%
7313 \LetLtxMacro\colorboxBlock\LWRprint@colorboxBlock%
7314 \LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
7315 \LetLtxMacro\fcolorboxBlock\LWRprint@fcolorboxBlock%
7316 \LetLtxMacro\fcolorminipage\LWRprint@fcolorminipage%
7317 \LetLtxMacro\endfcolorminipage\endLWRprint@fcolorminipage%
7318 }
7319

```



```

7320 \appto{\LWR@restoreorigformatting}{%
7321 \LWR@restoreorigprintxcolor%
7322 }

```

If print mode, immediately activate the print-mode enhancements for xcolor:

```


7323 \warpprintonly{\LWR@restoreorigprintxcolor}
7324
7325 \LWR@traceinfo{xcolor patches done}
7326 }{}% xcolor loaded
7327 }% AtBeginDocument

7328 \end{warppall}

```

## 72 Cleveref

Pkg `cleveref` `cleveref` package is used as-is with minor patches.

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

Ex:

```

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

```

See `\cpagerefFor` at page 394 to redefine the message which is printed for page number references.

**loading order** `cleveref` and the following associated macro patches are automatically preloaded at the end of the preamble via `\AtEndPreamble` and `\AfterEndPreamble`. This is done because the HTML conversion requires `cleveref`. The user’s document may not require `cleveref`, thus the user may never explicitly load it, so during HTML output `lwarp` loads it last. If the user’s document preamble uses `cleveref` options, or functions such as `\crefname`, then `cleveref` may be loaded in the user’s preamble near the end, and `lwarp`’s additional loading of `cleveref` will have no effect.

Table 10 on page 332 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:** `7329 \begin{warppHTML}`

`\AtEndPreamble` forces `cleveref` to be loaded last:

```
7330 \AtEndPreamble{
7331 \RequirePackage{cleveref}
7332 }
```

The following patches are applied after `cleveref` has loaded, and after `\AtBeginDocument`:

```
7333 \AfterEndPreamble{
7334 \LWR@traceinfo{Patching cleveref.}
```

`\@@setcref`  $\{\langle kindofref \rangle\} \{\langle label \rangle\}$

```
7335 \renewcommand*{\@@setcref}[2]{#1{\ref{#2}}\{\}\{\}}
```

`\@@setcrefrange`  $\{\langle text \rangle\} \{\langle label \rangle\} \{\langle label \rangle\}$

```
7336 \renewcommand{\@@setcrefrange}[3]{%
7337 #1{\ref{#2}}{\ref{#3}}\{\}\{\}\{\}}
```

`\cpagerefFor` Redefinable word between “page(s)” and the page numbers.

```
7338 \newcommand*{\cpagerefFor}{for}
```

`\@@setcpageref`  $\{\langle typeofref \rangle\} \{\langle label \rangle\}$ , where `typeofref` is “page” or “pages”

```
7339 \renewcommand*{\@@setcpageref}[2]{%
7340 #1{\cpagerefFor\ \cref{#2}}\{\}\{\}%
7341 }
```

```
7342 \renewcommand{\@@setcpagerefrange}[3]{%
7343 #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}\{\}\{\}\{\}%
7344 }% AfterEndPreamble
```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

```
7345 \LetLtxMacro\LWR@origlabel\label
7346 % \LetLtxMacro\label\LWR@newlabel
7347 \RenewDocumentCommand{\label}\{\}\{\LWR@newlabel\}
7348
7349 \LetLtxMacro\LWR@origref\ref
```

```

7350 % \LetLtxMacro\ref\LWR@newref
7351 \RenewDocumentCommand{\ref}{-}{\LWR@newref}%
7352
7353 \LetLtxMacro\LWR@origpageref\pageref
7354 % \LetLtxMacro\pageref\LWR@newpageref
7355 \RenewDocumentCommand{\pageref}{-}{\LWR@newpageref}
7356
7357 \end{warpHTML}

```

## 73 Picture

Env `picture` The `picture` environment is enclosed inside a `\lateximage`.

for HTML output: 7358 `\begin{warpHTML}`

Env `picture`

```


7359 \BeforeBeginEnvironment{picture}{\lateximage}
7360
7361 \AfterEndEnvironment{picture}{\endlateximage}

7362 \end{warpHTML}

```

## 74 Boxes and Minipages

A CSS flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

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

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

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


**in a span** There is limited support for minipages inside an HTML `<span>`. An HTML `<div>` cannot appear inside a `<span>`. While in a `<span>`, minipages, and parboxes, and

any enclosed lists have limited HTML tags, resulting in an “inline” format, without markup except for HTML breaks. Use `\newline` or `\par` for an HTML break.

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

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

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

 **text alignment** Nested minipages adopt their parent’s text alignment in HTML, whereas in regular  $\TeX$  PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

**for HTML output:** 7363 `\begin{warpHTML}`

## 74.1 Counters and lengths

**Ctrl** `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
7364 \newcounter{LWR@minipagedepth}
7365 \setcounter{LWR@minipagedepth}{0}
```

**Len** `\WR@minipagewidth` Used to convert the width into printable units.

```
7366 \newlength{\LWR@minipagewidth}
```

**Len** `\WR@minipageheight` Used to convert the height into printable units.

```
7367 \newlength{\LWR@minipageheight}
```

## 74.2 Footnote handling

Also see section 46 for other forms of footnotes. Minipage footnotes are gathered in section 46.5, and then placed into the document in section 74.3.

## 74.3 Minipage handling

`\LWR@endminipage` Used to close a minipage.

Copied the  $\text{\LaTeX}$  definition and modified to create a `<div>` of class `mpfootnotes`:

```

7368 \def\LWR@endminipage{%
7369   \par
7370   \unskip
7371   \ifvoid\@mpfootins\else
7372     \vskip\skip\@mpfootins
7373     \normalcolor
7374     \LWR@htmldivclass{mpfootnotes}
7375     \LWR@origmedskip
7376     \unvbox\@mpfootins
7377     \LWR@htmldivclassend{mpfootnotes}
7378   \fi
7379   \@minipagefalse
7380   \color@endgroup
7381   \egroup
7382   \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.

```

7383 \newcommand*{\LWR@subminipage}{%
7384 \LWR@stoppars
7385 \LWR@origminipage{6in}

```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```

7386 \LWR@origraggedright%

```

Resume paragraph tag handling for the contents of the minipage:

```

7387 \LWR@startpars%
7388 }

```

`\LWR@endsubminipage` Closes the subminipage.

```

7389 \newcommand*{\LWR@endsubminipage}{%
7390 \LWR@stoppars%
7391 \LWR@endminipage% The following empty line is required:
7392
7393 }

```

Bool `LWR@minipagefullwidth` Should the next minipage have no HTML width?

```

7394 \newbool{LWR@minipagefullwidth}
7395 \boolfalse{LWR@minipagefullwidth}

```

`\minipagefullwidth` Requests that the next minipage have no width tag in HTML:

**for HTML output:** 7396 `\newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}`  
 7397 `\end{warpHTML}`

**for PRINT output:** 7398 `\begin{warpprint}`  
 7399 `\newcommand*{\minipagefullwidth}{}`  
 7400 `\end{warpprint}`

**for HTML output:** 7401 `\begin{warpHTML}`

Bool `LWR@minipagethispar` Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

7402 `\newbool{LWR@minipagethispar}`  
 7403 `\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.

7404 `\RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}`  
 7405 `{%`

Units for printing dimensions to HTML:

7406 `\uselengthunit{PT}%`

Compute width, adjusted for frames:

7407 `\setlength{\LWR@minipagewidth}{#4}%`  
 7408 `\ifthenelse{\cinttest{\value{LWR@minipagedepth}}{=}{0}}{%`  
 7409 `\addtolength{\LWR@minipagewidth}{3em}% room for frames`  
 7410 `\setlength{\linewidth}{6in}%`  
 7411 `\setlength{\textwidth}{6in}%`  
 7412 `\setlength{\textheight}{9in}%`  
 7413 `}%}`  
 7414 `\LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}`

Compute height:

7415 `\setlength{\LWR@minipageheight}{\textheight}% default unless specified`  
 7416 `\IfValueT{#2}{\setlength{\LWR@minipageheight}{#2}}%`

Track nesting depth:

```
7417 \addtocounter{LWR@minipagedepth}{1}%
```

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.

```
7418 \ifbool{FormatWP}{\newline}{}%
```

```
7419 \LWR@stoppars%
```

If FormatWP, add a text frame:

```
7420 \ifbool{FormatWP}{%
```

```
7421
```

```
7422 \addtocounter{LWR@thisfloat}{1}%
```

```
7423 \booltrue{LWR@freezethisfloat}%
```

```
7424 \LWR@htmltag{div id="autofloat-\arabic{LWR@thisfloat}" class="wpminipage"}%
```

```
7425
```

```
7426 }{}%
```

Create the <div> tag with optional alignment style:

```
7427 \LWR@traceinfo{minipage: creating div class}%
```

```
7428 \LWR@orignewpage%
```

```
7429 \LWR@htmltag{div class="minipage" style="%
```

```
7430 \ifthenelse{\equal{#1}{t}}{vertical-align:bottom ; }{}%
```

```
7431 \ifthenelse{\equal{#1}{c}}{vertical-align:middle ; }{}%
```

```
7432 \ifthenelse{\equal{#1}{b}}{vertical-align:top ; }{}%
```

```
7433 \ifthenelse{\equal{#3}{t}}{justify-content:flex-start ; }{}%
```

```
7434 \ifthenelse{\equal{#3}{c}}{justify-content:center ; }{}%
```

```
7435 \ifthenelse{\equal{#3}{b}}{justify-content:flex-end ; }{}%
```

```
7436 \ifthenelse{\equal{#3}{s}}{justify-content:space-between ; }{}%
```

Print the width and optional height styles:

```
7437 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
```

```
7438 \uselengthunit{PT}%
```

```
7439 \ifbool{LWR@minipagefullwidth}%
```

```
7440 {\boolfalse{LWR@minipagefullwidth}}%
```

```
7441 {%
```

```
7442   \ifthenelse{\lengthtest{#4}=\linewidth}%
```

```
7443   }{}%
```

```
7444   {width:\rndprintlength{\LWR@minipagewidth} ; }%
```

```
7445 }%
```

```
7446 \LWR@traceinfo{minipage: about to print the height}%
```

```
7447 \IfValueT{#2}{height:\rndprintlength{\LWR@minipageheight} ; }%
```

```
7448 "{}%
```

Finish with an empty line to start  $\LaTeX$  minipage processing on a new line. Use a large minipage area to avoid the unnecessary wrapping of tags.

7449

```
7450 \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.

```
7451 \setlength{\linewidth}{#4}% the original width
```

```
7452 \setlength{\textwidth}{6in}%
```

```
7453 \setlength{\textheight}{9in}%
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
7454 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
7455 \LWR@startpars%
```

```
7456 \ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%
```

```
7457
```

```
7458 === begin minipage ===
```

```
7459
```

```
7460 }{}%
```

```
7461 \LWR@traceinfo{minipage: finished starting the minipage}%
```

```
7462 }
```

End the environment with  $\LaTeX$  processing and closing tag:

```
7463 {%
```

```
7464 \ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%
```

```
7465
```

```
7466 === end minipage ===
```

```
7467
```

```
7468 }{}%
```

```
7469 \LWR@stoppars%
```

```
7470 \LWR@endminipage% The following empty line is required:
```

```
7471
```

```
7472 \ifbool{FormatWP}{%
```

```
7473
```

```
7474 \LWR@htmlelementend{div}%
```

```
7475 \boolfalse{LWR@freezethisfloat}%
```

```
7476
```

```
7477 }{}%
```

```
7478 \LWR@htmldivclassend{minipage}%
```

```
7479
```



```

7480 \LWR@origvspace{1\baselineskip}% required for subcaption
7481 \addtocounter{LWR@minipagedepth}{-1}%
7482 \LWR@startpars%
7483 \ifbool{FormatWP}{\newline}{}%

```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```

7484 \global\booltrue{LWR@minipagethispar}%
7485 }

```

## 74.4 Parbox, makebox, framebox, fbox, raisebox

for HTML output:

```
\parbox [<pos>] [<height>] [<inner-pos>] {<width>}{<text>}
```

A parbox uses the minipage code:

```

7486 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}
7487 {
7488 \LWR@traceinfo{parbox of width #4}%
7489 \begin{minipage}[#1][#2][#3]{#4}
7490 #5
7491 \end{minipage}
7492 }

```

```
\makebox (<(>)posn) [<width>] [<pos>] {<text>}
```

```

7493 \LetLtxMacro\LWR@origmakebox\makebox
7494
7495 \RenewDocumentCommand{\makebox}{d() o o m}{%

```

Check for the optional width:

```

7496 \IfValueTF{#2}%
7497 {%

```

Check for the horizontal text alignment. For stretched, the best HTML can do is justified alignment.

```

7498   {% scope
7499   \def\LWR@align{center}%
7500   \ifstrequal{#3}{l}{\def\LWR@align{left}}{}%
7501   \ifstrequal{#3}{r}{\def\LWR@align{right}}{}%
7502   \ifstrequal{#3}{s}{\def\LWR@align{justify}}{}%

```

To print the width argument:

```
7503 \setlength{\LWR@tempwidth}{#2}%
```

inline-block allows width and text-alignment to be used in a <span>.

```
7504 \uselengthunit{PT}%
7505 \InlineClass[%
7506     display:inline-block ; %
7507     text-align:\LWR@align\ ; %
7508     width:\rndprintlength{\LWR@tempwidth}%
7509 ]%
7510 {makebox}%
```

Without a width argument, the text is simply used inline:

```
7511 {\mbox{#4}}%
7512 }% scope
7513 }%
7514 {\mbox{#4}}%
7515 }
```

`\framebox` [*<width>*] [*<pos>*] {*<text>*}

```
7516 \LetLtxMacro\LWR@origframebox\framebox
7517
7518 \RenewDocumentCommand{\framebox}{o o m}{%
7519 \fbox{\makebox[#1][#2]{#3}}%
7520 }
```

`\LWR@forceminwidth` {*<legth>*}

Sets `\LWR@atleastonept` to be at least 1pt.

```
7521 \newlength{\LWR@atleastonept}
7522
7523 \newcommand*{\LWR@forceminwidth}[1]{%
7524 \setlength{\LWR@atleastonept}{#1}%
7525 \ifthenelse{%
7526     \lengthtest{\LWR@atleastonept>0pt}\AND%
7527     \lengthtest{\LWR@atleastonept<1pt}%
7528 }%
7529 {\setlength{\LWR@atleastonept}{1pt}}%
7530 {}%
7531 }
```

`\LWR@blackborderpadding` Prints the HTML attributes for a black border and padding.

`\LWR@forceminwidth` must be used first in order to set the border width.

```
7532 \newcommand*{\LWR@blackborderpadding}{%
7533 \uselengthunit{PT}}%
7534 border:\rndprintlength{\LWR@atleastonept} solid black ; %
7535 padding:\rndprintlength{\fboxsep}%
7536 }
```

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

Creates a framed inline span enclosing the text.

Remember the print-mode version:

```
7537 \let\LWRprint@fbox\fbox
```

Create a new HTML version, but don't use it until after `xcolor` may have loaded:

```
7538 \newcommand{\LWRhtml@fbox}[1]{%
7539 \LWR@traceinfo{HTML fbox}
7540 \LWR@forceminwidth{\fboxrule}%
7541 \InlineClass[%
7542 \LWR@blackborderpadding%
7543 ]{\fbox}{#1}
7544 }
```

`xcolor` \lets things to `\fbox` when it is loaded, and this must remain even for HTML output while in a `lateximage`, so `\fbox` is not modified until `\AtBeginDocument`:

```
7545 \AtBeginDocument{\let\fbox\LWRhtml@fbox}
```

`\fboxBlock`  $\{\langle text \rangle\}$  Creates a framed HTML `<div>` of the text.

A print-output version is also supplied below.

```
7546 \newcommand{\fboxBlock}[1]{%
7547 \LWR@forceminwidth{\fboxrule}%
7548 \begin{BlockClass}%
7549 \LWR@blackborderpadding%
7550 ]{\fboxBlock}
7551 #1
7552 \end{BlockClass}
7553 }
```

Env `fminipage`  $[\langle align \rangle] [\langle height \rangle] [\langle align \rangle] \{\langle width \rangle\}$

Creates a framed HTML `<div>` around its contents.

A print-output version is also supplied below.

```

7554 \NewDocumentEnvironment{fminipage}{0{t} o 0{t} m}
7555 {%
7556 \LWR@traceinfo{fminipage #1 #2 #3 #4}%
7557 \LWR@forceminwidth{\fboxrule}%
7558 \setlength{\LWR@tempwidth}{#4}%
7559 \IfValueT{#2}{\setlength{\LWR@tempheight}{#2}}%
7560 \begin{BlockClass}%
7561 \LWR@blackborderpadding ; %
7562 \uselengthunit{PT}%
7563 \IfValueT{#2}{height:\rndprintlength{\LWR@tempheight} ; }%
7564 width:\rndprintlength{\LWR@tempwidth}%
7565 ]{fminipage}%
7566 }
7567 {%
7568 \end{BlockClass}%
7569 \LWR@traceinfo{fminipage done}%
7570 }

```

`\raisebox`  $\{ \langle \textit{raiselen} \rangle \} [ \langle \textit{height} \rangle ] [ \langle \textit{depth} \rangle ] \{ \langle \textit{text} \rangle \}$

```

7571 \LetLtxMacro{\LWR@origraisebox}{\raisebox}
7572
7573 \RenewDocumentCommand{\raisebox}{m o o m}{%
7574 #4%
7575 }

7576 \end{warpHTML}

```

**for HTML & PRINT:** 7577 \begin{warpall}

`LWRprint@fminipage` is defined inside `warpall`. For print output, it is `\let` to `fminipage`. For HTML output, the HTML version of `fminipage` is used instead, but the print version is still available for use inside a `lateximage`.

Env `LWRprint@fminipage`  $[ \langle 1:\textit{align} \rangle ] [ \langle 2:\textit{height} \rangle ] [ \langle 3:\textit{inner-align} \rangle ] \{ \langle 4:\textit{width} \rangle \}$

Creates a frame around its contents.

```

7578 \newsavebox{\LWR@fminipagebox}
7579
7580 \NewDocumentEnvironment{LWRprint@fminipage}{0{t} o 0{t} m}
7581 {%

```

An outer minipage will be used for vertical alignment. An inner minipage will be framed with `\fbox`.

If the optional inner alignment is not given, use the outer instead:

```
7582 \IfValueTF{#3}%
7583 {\def\LWR@thisalign{#3}}
7584 {\def\LWR@thisalign{#1}}%
```

Form the outer minipage depending on whether a height was given. Make the outer minipage larger to compensate for the frame.

```
7585 \IfValueTF{#2}%
7586 {\minipage[#1][#2+2\fbboxsep+2\fbboxrule][\LWR@thisalign]{#4+2\fbboxsep+2\fbboxrule}}%
7587 {\minipage[#1]{#4+2\fbboxsep+2\fbboxrule}}%
```

Capture the contents of the environment:

```
7588 \begin{lrbox}{\LWR@fminipagebox}%
```

Nest the contents inside an inner minipage of the desired size:

```
7589 \IfValueTF{#2}%
7590 {\minipage[#1][#2][\LWR@thisalign]{#4}}%
7591 {\minipage[#1]{#4}}%
7592 }
7593 {%
```

Close the inner minipage and the LR box with the contents:

```
7594 \endminipage%
7595 \end{lrbox}%
```

Create a frame around the contents of the environment:

```
7596 \fbbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```
7597 \endminipage%
7598 }
```

```
7599 \end{warpall}
```

**for PRINT output:** 7600 \begin{warpprint}

For print output, the following are \let to become active.

```
\fbboxBlock {\langle text \rangle}
```

Creates a framed HTML <div> around the text.

```
7601 \let\fbboxBlock\fbbox
```

Env `fminipage` [*<align>*] [*<height>*] [*<align>*] {*<width>*}

Creates a frame around its contents.

```
7602 \LetLtxMacro{fminipage}{\LWRprint@fminipage}
7603 \LetLtxMacro{endfminipage}{\endLWRprint@fminipage}

7604 \end{warpprint}
```

## 75 Direct formatting

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

△ **HTML special chars** `&`, `<`, and `>` have special meanings in HTML. If `\&`, `\textless`, and `\textgreater` are used, the proper result should occur in HTML, but there may be HTML parsing problems if these special characters occur unescaped in program listings or other verbatim text.

For high-level block and inline custom CSS classes, see section 40.8.

for HTML output: 7605 `\begin{warppHTML}`

`\LWR@HTMLtextstyle` {*<FormatWP style>*} {*<class>*} {*<text>*}

If `FormatWP`, adds an explicit style to the text span class. This is used by LibreOffice to mark its imported text using the given style.

```
7606 \DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{%
7607 \ifbool{FormatWP}%
7608 {\LWR@htmlspanclass[#1]{#2}{#3}}%
7609 {\LWR@htmlspanclass{#2}{#3}}%
7610 }
```

`\emph` {*<text>*}

```
7611 \DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{#1}}
7612 \DeclareRobustCommand{\LWR@nullemph}[1]{#1}
7613 \LetLtxMacro{\emph}{\LWR@HTMLemph}
```

`\textmd {<text>}`

```
7614 \DeclareRobustCommand{\LWR@HTMLtextmd}[1]{%
7615 \LWR@HTMLtextstyle{font-weight:normal}{textmd}{#1}%
7616 }
7617 \DeclareRobustCommand{\LWR@nulltextmd}[1]{#1}
7618
7619 \LetLtxMacro{\textmd}{\LWR@HTMLtextmd}
```

`\textbf {<text>}`

```
7620 \DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{#1}}
7621 \DeclareRobustCommand{\LWR@nulltextbf}[1]{#1}
7622 \LetLtxMacro{\textbf}{\LWR@HTMLtextbf}
```

`\textrm {<text>}`

```
7623 \DeclareRobustCommand{\LWR@HTMLtextrm}[1]{%
7624 \LWR@HTMLtextstyle{font-family:serif}{textrm}{#1}%
7625 }
7626
7627 \DeclareRobustCommand{\LWR@nulltextrm}[1]{#1}
7628
7629 \LetLtxMacro{\textrm}{\LWR@HTMLtextrm}
```

`\textsf {<text>}`

```
7630 \DeclareRobustCommand{\LWR@HTMLtextsf}[1]{%
7631 \LWR@HTMLtextstyle{font-family:sans}{textsf}{#1}%
7632 }
7633 \DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
7634 \LetLtxMacro{\textsf}{\LWR@HTMLtextsf}
```

`\texttt {<text>}`

```
7635 \DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
7636 \DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
7637 \LetLtxMacro{\texttt}{\LWR@HTMLtexttt}
```

`\textup {<text>}`

```
7638 \DeclareRobustCommand{\LWR@HTMLtextup}[1]{%
7639 \LWR@HTMLtextstyle{font-variant:normal}{textup}{#1}%
7640 }
7641
7642 \DeclareRobustCommand{\LWR@nulltextup}[1]{#1}
```

```
7643
7644 \LetLtxMacro{\textup}{\LWR@HTMLtextup}
```

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

```
7645 \DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}}
7646 \DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
7647 \LetLtxMacro{\textit}{\LWR@HTMLtextit}
```

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

```
7648 \DeclareRobustCommand{\LWR@HTMLtextsc}[1]{%
7649 \LWR@HTMLtextstyle{font-variant:small-caps}{textsc}{#1}%
7650 }
7651
7652 \DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}
7653
7654 \LetLtxMacro{\textsc}{\LWR@HTMLtextsc}
```

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

```
7655 \DeclareRobustCommand{\LWR@HTMLtextsl}[1]{%
7656 \LWR@HTMLtextstyle{font-style:oblique}{textsl}{#1}%
7657 }
7658
7659 \DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}
7660
7661 \LetLtxMacro{\textsl}{\LWR@HTMLtextsl}
```

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

```
7662 \DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
7663 \DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
7664 \LetLtxMacro{\textnormal}{\LWR@HTMLtextnormal}

7665 \DeclareRobustCommand{\LWR@nullrmfamily}{}
7666 \DeclareRobustCommand{\LWR@nullsfamily}{}
7667 \DeclareRobustCommand{\LWR@nullttfamily}{}
7668 \DeclareRobustCommand{\LWR@nullbfseries}{}
7669 \DeclareRobustCommand{\LWR@nullmdseries}{}
7670 \DeclareRobustCommand{\LWR@nullupshape}{}
7671 \DeclareRobustCommand{\LWR@nullslshape}{}
7672 \DeclareRobustCommand{\LWR@nullscshape}{}
7673 \DeclareRobustCommand{\LWR@nullitshape}{}
7674 \DeclareRobustCommand{\LWR@nullem}[1]{}
7675 \DeclareRobustCommand{\LWR@nullnormalfont}{}

```



`\LWR@nullfonts` Removes formatting during filename operations.

```

7676 \newcommand*{\LWR@nullfonts}{%
7677 \LetLtxMacro{\emph}{\LWR@nullemph}%
7678 \LetLtxMacro{\textmd}{\LWR@nulltextmd}%
7679 \LetLtxMacro{\textbf}{\LWR@nulltextbf}%
7680 \LetLtxMacro{\textrm}{\LWR@nulltextrm}%
7681 \LetLtxMacro{\textsf}{\LWR@nulltextsf}%
7682 \LetLtxMacro{\texttt}{\LWR@nulltexttt}%
7683 \LetLtxMacro{\textup}{\LWR@nulltextup}%
7684 \LetLtxMacro{\textit}{\LWR@nulltextit}%
7685 \LetLtxMacro{\textsc}{\LWR@nulltextsc}%
7686 \LetLtxMacro{\textsl}{\LWR@nulltextsl}%
7687 \LetLtxMacro{\textnormal}{\LWR@nulltextnormal}%
7688 \LetLtxMacro{\rmfamily}{\LWR@nullrmfamily}%
7689 \LetLtxMacro{\sffamily}{\LWR@nullsffamily}%
7690 \LetLtxMacro{\ttfamily}{\LWR@nullttfamily}%
7691 \LetLtxMacro{\bfseries}{\LWR@nullbfseries}%
7692 \LetLtxMacro{\mdseries}{\LWR@nullmdseries}%
7693 \LetLtxMacro{\upshape}{\LWR@nullupshape}%
7694 \LetLtxMacro{\slshape}{\LWR@nullslshape}%
7695 \LetLtxMacro{\scshape}{\LWR@nullscshape}%
7696 \LetLtxMacro{\itshape}{\LWR@nullitshape}%
7697 \LetLtxMacro{\em}{\LWR@nullem}%
7698 \LetLtxMacro{\normalfont}{\LWR@nullnormalfont}%
7699 \renewcommand*{\HTMLUnicode}[1]{}%
7700 \renewcommand*{\HTMLentity}[1]{}%

```

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

```

7701 \renewcommand*{\&}{and}%
7702 \renewcommand{\textsuperscript}[1]{##1}%
7703 \renewcommand{\textsubscript}[1]{##1}%
7704 \LetLtxMacro\underline\LWR@origunderline%
7705 \RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
7706 \DeclareExpandableDocumentCommand{\InlineClass}{+o +m +m}{##3}%
7707 \DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{##3}%
7708 \DeclareRobustCommand{\LWR@subsingledollar}[1]{}%
7709 }

```

`\mdseries`

```

7710 \renewcommand*{\mdseries}{}

```

`\bfseries`

```

7711 \renewcommand*{\bfseries}{}

```

`\rmfamily`

7712 `\renewcommand*\rmfamily{}\}`

`\sffamily`

7713 `\renewcommand*\sffamily{}\}`

`\ttfamily`

7714 `\renewcommand*\ttfamily{}\}`

`\upshape`

7715 `\renewcommand*\upshape{}\}`

`\itshape`

7716 `\renewcommand*\itshape{}\}`

`\scshape`

7717 `\renewcommand*\scshape{}\}`

`\normalfont`

7718 `\renewcommand*\normalfont{}\}`

`\sp`  $\langle text \rangle$

For siunitx. Must work in math mode.

7719 `\renewcommand{\sp}[1]{\text{<sup>#1</sup>}}\}`

`\sb`  $\langle text \rangle$

For siunitx. Must work in math mode.

7720 `\renewcommand{\sb}[1]{\text{<sub>#1</sub>}}\}`

`\textsuperscript`  $\langle text \rangle$

7721 `\renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}`

`\@textsuperscript`  $\{\langle text \rangle\}$

```
7722 \renewcommand{\@textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

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

```
7723 \AtBeginDocument{
7724 \renewcommand{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
7725 }
```

`\@textsubscript`  $\{\langle text \rangle\}$

```
7726 \AtBeginDocument{
7727 \renewcommand{\@textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
7728 }
```

`\up`  $\{\langle text \rangle\}$  Prints superscript.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
7729 \AtBeginDocument{\let\up\textsuperscript}
```

`\fup`  $\{\langle text \rangle\}$  Prints superscript.

Supports `fntcount` package.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
7730 \AtBeginDocument{\let\fup\textsuperscript}
```

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

```
7731 \renewcommand{\underline}[1]{%
7732 \LWR@HTMLtextstyle%
7733 {text-decoration:underline;text-decoration-skip}%
7734 {underline}{#1}%
7735 }
```

`\hfill`

```
7736 \renewcommand*\hfill{\quad}
```

`\hrulefill`

```
7737 \renewcommand*{\hrulefill}{\rule{1in}{1pt}}
```

`\dotfill`

```
7738 \renewcommand*{\dotfill}{\dots}
```

```
7739 \end{warpHTML}
```

## 76 Skips, spaces, font sizes

for HTML output: 7740 `\begin{warpHTML}`

`\`, must be redefined after `\RequirePackage{printlen}`

```
7741 \let\LWR@origcomma\,
7742 \let\LWR@origtilde~
7743 \let\LWR@origenskip\enskip
7744 \let\LWR@origquad\quad
7745 \let\LWR@origqqquad\qqquad
7746 \let\LWR@orighspace\hspace
7747 \let\LWR@origvspace\vspace
7748 \let\LWR@origrule\rule
7749 \let\LWR@origmedskip\medskip
7750 \let\LWR@origtextellipsis\textellipsis
```

Direct-formatting space commands become HTML entities:

```
7751 \renewcommand*{\,}{\HTMLUnicode{202f}} % HTML thin non-breakable space
```

```
7752
```

```
7753 \renewcommand*{~}{\HTMLentity{nbsp}}
```

```
7754
```

```
7755 \renewcommand*{\textellipsis}{\HTMLUnicode{2026}}
```

Direct-formatting font sizes are ignored:

```
7756 \let\LWR@orignormalsize\normalsize
7757 \let\LWR@origsmall\small
7758 \let\LWR@origfootnotesize\footnotesize
7759 \let\LWR@origscriptsize\scriptsize
7760 \let\LWR@origtiny\tiny
7761 \let\LWR@origlarge\large
7762 \let\LWR@origLarge\Large
```

```

7763 \let\LWR@origLARGE\LARGE
7764 \let\LWR@orighuge\huge
7765 \let\LWR@origHuge\Huge
7766 \renewcommand*{\normalsize}{ }
7767 \renewcommand*{\small}{ }
7768 \renewcommand*{\footnotesize}{ }
7769 \renewcommand*{\scriptsize}{ }
7770 \renewcommand*{\tiny}{ }
7771 \renewcommand*{\large}{ }
7772 \renewcommand*{\Large}{ }
7773 \renewcommand*{\LARGE}{ }
7774 \renewcommand*{\huge}{ }
7775 \renewcommand*{\Huge}{ }
7776
7777 \renewcommand*{\onecolumn}{ }
7778
7779 \renewcommand{\twocolumn}[1][ ]{
7780
7781 #1
7782
7783 }

```

`\newline` Uses the HTML `<br />` element.

```

7784 \newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
7785 \let\newline\LWR@newlinebr

```

`\\` Redefined to `\LWR@endoffline` or `\LWR@tabularendoffline`.

`\LWR@endoffline` \* [*len*]

`\\` is assigned to `\LWR@endoffline` at `\LWR@LwarpStart`.

Inside `tabular`, `\\` is temporarily changed to `\LWR@tabularendoffline`.

```

7786 \LetLtxMacro\LWR@origendoffline\\
7787 \NewDocumentCommand{\LWR@endoffline}{s o}
7788 {%
7789 \newline%
7790 }

```

`\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

`\hspace` after a minipage to be on its own line. Since people often place small horizontal space  
`\enskip` between minipages, it is desirable to maintain this space if possible. lwarp tries to  
`\quad` do this by remembering that a minipage has been seen, in which case paragraph  
`\qquad` tags are suppressed around `\hspace`, `\enskip`, `\quad`, and `\qquad` until the end of  
the paragraph, when the closing `p` tag is created.

When a minipage is seen, the boolean `LWR@minipagethispar` is set, telling the following horizontal whitespace commands to try to suppress their surrounding paragraph tags. `LWR@minipagethispar` is cleared at the next end of paragraph, when the HTML paragraph closing tag is generated.

Placed just before `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
7791 \newcommand*{\LWR@minipagestartpars}{%
7792 \ifbool{LWR@minipagethispar}{\LWR@startpars}{}%
7793 }
```

`\LWR@minipagestoppars` Placed just after `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
7794 \newcommand*{\LWR@minipagestoppars}{%
7795 \ifbool{LWR@minipagethispar}{\LWR@stoppars}{}%
7796 }
```

`\quad` Handles special minipage & horizontal space interactions.

```
7797 \renewcommand*{\quad}{%
7798 \LWR@minipagestoppars%
7799 \HTMLUnicode{2001}%
7800 \LWR@minipagestartpars%
7801 }
```

`\qquad` Handles special minipage & horizontal space interactions.

```
7802 \renewcommand*{\qquad}{\quad\quad}
```

`\enskip` Handles special minipage & horizontal space interactions.

```
7803 \renewcommand*{\enskip}{%
7804 \LWR@minipagestoppars%
7805 \HTMLUnicode{2000}%
7806 \LWR@minipagestartpars%
7807 }
```

Len `\WR@tempwidth` Used to compute span width, height, raise for `\hspace` and `\rule`:

Len `\WR@tempheight`

Len `\WR@tempraise`

```

7808 \newlength{\LWR@tempwidth}
7809 \newlength{\LWR@tempheight}
7810 \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`.

```

7811 \NewDocumentCommand{\LWR@hspace}{s m}{%
7812 \setlength{\LWR@tempwidth}{#2}%

```

If `\fill`, change to `\qquad`:

```

7813 \ifnum\gluestretchorder\LWR@tempwidth>0%
7814 \setlength{\LWR@tempwidth}{2em}%
7815 \fi%

```

Only if the width is not zero:

```

7816 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{0pt}}{ }{%

```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

7817 \LWR@minipagestoppars%

```

Support the HTML thin wrappable space:

```

7818 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{.16667em}}
7819 {%
7820 \HTMLUnicode{2009}% thin breakable space
7821 }%

```

Print the span with the converted width. Not rounded.

```

7822 {%
7823 \uselengthunit{PT}%
7824 \LWR@htmltagc{%
7825 span style="width:\printlength{\LWR@tempwidth}; %
7826 display:inline-block"%
7827 }%

```

If formatting for a word processor, approximate with a number of \quads, in case a span of a given width is not supported:

```

7828      \ifbool{FormatWP}{%
7829          \setlength{\LWR@templengthone}{\LWR@tempwidth}%
7830          \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
7831              \quad%
7832              \addtolength{\LWR@templengthone}{-1em}%
7833          }%
7834      }{}%
```

Close the span:

```

7835      \LWR@htmltagc{/span}%
7836  }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

7837      \LWR@minipagestartpars%
7838  }% width not 0
7839 }
```

`\LWR@nohspace` \*  $\{\langle length \rangle\}$

Used to disable \hspace while creating description \items.

```
7840 \NewDocumentCommand{\LWR@nohspace}{s m}{}
```

`\hspace` \*  $\{\langle length \rangle\}$

Handles special minipage & horizontal space interactions.

```
7841 \LetLtxMacro{\hspace}{\LWR@hspace}
```

`\LWR@vspace` \*  $\{\langle length \rangle\}$  Nullified vspace.

```
7842 \NewDocumentCommand{\LWR@vspace}{s m}{}
```

`\vspace` \*  $\{\langle length \rangle\}$  Nullified.

```
7843 \let\vspace\LWR@vspace
```

`\linebreak` [ $\langle num \rangle$ ] Inserts an HTML br tag.

```
7844 \renewcommand*{\linebreak}[1][]{\newline}
```



`\nolinebreak` [ $\langle num \rangle$ ]

```
7845 \renewcommand*{\nolinebreak}[1] [] {}
```

`\pagebreak` [ $\langle num \rangle$ ] Starts a new paragraph.

```
7846 \renewcommand*{\pagebreak}[1] [] {}
7847
7848 }
```

`\nopagebreak` [ $\langle num \rangle$ ]

```
7849 \renewcommand*{\nopagebreak}[1] [] {}
```

`\enlargethispage` \*  $\{\langle len \rangle\}$

```
7850 \RenewDocumentCommand{\enlargethispage}{s m}{}%
```

`\LWR@currenttextcolor` The color to use for text and `\rule`, defaulting to black:

```
7851 \newcommand*{\LWR@currenttextcolor}{black}
```

`\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.

```
7852 \NewDocumentCommand{\LWR@rule}{o m m}{%%
```

The width is copied into a temporary  $\text{\L\TeX}$  length, from which comparisons and conversions may be made:

```
7853 \setlength{\LWR@tempwidth}{#2}%%
```

If it's zero-width then skip the entire rule:

```
7854 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}{
7855 {}}% zero- width
7856 {}% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```

7857 \ifthenelse{%
7858     \lengthtest{\LWR@tempwidth>0pt}\AND%
7859     \lengthtest{\LWR@tempwidth<1pt}%
7860 }%
7861 {\setlength{\LWR@tempwidth}{1pt}}{}%

```

Likewise with height:

```

7862 \setlength{\LWR@tempheight}{#3}%
7863 \ifthenelse{%
7864     \lengthtest{\LWR@tempheight>0pt}\AND%
7865     \lengthtest{\LWR@tempheight<1pt}%
7866 }%
7867 {\setlength{\LWR@tempheight}{1pt}}{}%

```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```

7868 \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  $\text{\LaTeX}$  code.

```

7869 \uselengthunit{PT}%
7870 \LWR@htmltagc{%
7871 span
7872 style="%

```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```

7873 \ifbool{FormatWP}{\background:\LWR@currenttextcolor; }%

```

The width and height are printed, converted to PT:

```

7874 width:\printlength{\LWR@tempwidth}; %
7875 height:\printlength{\LWR@tempheight}; %

```

The raise height is converted to a css transform. The \*2 raise multiplier is to approximately match HTML output's X height. Conversion to a  $\text{\LaTeX}$  length allows a typical  $\text{\LaTeX}$  expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a  $\text{\LaTeX}$  length limits the allowable syntax. To do: A superior method would compute a ratio of  $\text{\LaTeX}$  ex height, then print that to HTML with an ex unit.

```

7876 \IfValueT{#1}%
7877 {%
7878     \setlength{\LWR@tempraise}{Opt-#1}%

```

```

7879      \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
7880      \LWR@orignewline%
7881      -ms-transform: translate(0pt,\printlength{\LWR@tempraise}); %
7882      \LWR@orignewline%
7883      -webkit-transform: translate(0pt,\printlength{\LWR@tempraise}); %
7884      \LWR@orignewline%
7885      transform: translate(0pt,\printlength{\LWR@tempraise}); %
7886      \LWR@orignewline%
7887  }%

```

Display inline-block to place the span inline with the text:

```

7888      display:inline-block;"%
7889  }%

```

If formatting for a word processor, approximate with a number of underscores, in case a span of a given width is not supported:

```

7890      \ifbool{FormatWP}{%
7891          \setlength{\LWR@templengthone}{\LWR@tempwidth}%
7892          \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
7893              \_{}%
7894              \addtolength{\LWR@templengthone}{-1em}%
7895          }%
7896      }{}%

```

Close the span:

```

7897      \LWR@htmltagc{/span}%

```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

7898      \LWR@minipagestartpars%
7899  }% non-zero width
7900 }

```

`\rule` [*<raise>*] {*<width>*} {*<height>*}

Handles special minipage & horizontal space interactions.

```

7901 \renewcommand{\rule}{\LWR@rule}

```

```

7902 \end{warpHTML}

```

## 77 `\phantomsection`

for HTML output: 7903 `\begin{warpHTML}`

`\phantomsection` Emulate the `hyperref` `\phantomsection` command, often used to insert the bibliography into table of contents:

```
7904 \newcommand*{\phantomsection}{%
7905 \section*{}%
7906 }

7907 \end{warpHTML}
```

## 78 `\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>

### 78.1 HTML logos

for HTML output: 7908 `\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  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{LuaT}_{\mathrm{E}}\mathrm{X}$ ,  $\mathrm{ConT}_{\mathrm{E}}\mathrm{Xt}$ , etc.

```
7909 \let\LWR@origTeX\TeX
7910
7911 \newcommand*{\LWR@TeX}
7912 {%
7913     \InlineClass{latexlogofont}%
7914     {%
```

```

7915      \LWR@HTMLtextstyle%
7916          {text-transform:uppercase}%
7917          {latexlogo}%
7918      {T\textsubscript{e}X}%
7919  }%
7920 }

```

`\LaTeX`  $\mathbb{L}\mathbb{E}\mathbb{X}$ ,  $\mathbb{L}\mathbb{E}\mathbb{X}2_{\epsilon}$

`\LaTeXe`

```

7921 \let\LWR@origLaTeX\LaTeX
7922
7923 \newcommand*{\LWR@LaTeX}
7924 {%
7925     \InlineClass{latexlogofont}%
7926     {%
7927         \LWR@HTMLtextstyle%
7928         {text-transform:uppercase}%
7929         {latexlogo}%
7930         {L\textsuperscript{a}T\textsubscript{e}X}%
7931     }%
7932 }
7933
7934 \let\LWR@origLaTeXe\LaTeXe
7935
7936 \renewcommand*{\LaTeXe}
7937 {\LaTeX\InlineClass{latexlogofont}%
7938 {\,2\textsubscript{\textit{\HTMLUnicode{3B5}}}}}

```

`\LuaTeX`  $\mathbb{L}\mathbb{u}\mathbb{A}\mathbb{T}\mathbb{E}\mathbb{X}$ ,  $\mathbb{L}\mathbb{u}\mathbb{A}\mathbb{E}\mathbb{X}$

`\LuaLaTeX`

```

7939 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}\Lua}\TeX}
7940 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}\Lua}\LaTeX}

```

`\XeTeX`  $\mathbb{X}\mathbb{E}\mathbb{T}\mathbb{E}\mathbb{X}$ ,  $\mathbb{X}\mathbb{E}\mathbb{L}\mathbb{A}\mathbb{T}\mathbb{E}\mathbb{X}$

`\XeLaTeX`

`xetexlogo` is a css class which aligns the backwards E in  $\mathbb{X}\mathbb{E}\mathbb{T}\mathbb{E}\mathbb{X}$  and spaces  $\mathbb{T}\mathbb{E}\mathbb{X}$  appropriately.

`xelatexlogo` is a css class which aligns the backwards E in  $\mathbb{X}\mathbb{E}\mathbb{L}\mathbb{A}\mathbb{T}\mathbb{E}\mathbb{X}$  and spaces  $\mathbb{L}\mathbb{A}\mathbb{T}\mathbb{E}\mathbb{X}$  appropriately.

```

7941 \newcommand*{\Xe}
7942   {X\textsubscript{\HTMLUnicode{18e}}}
7943 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}\Xe}\TeX}
7944 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}\Xe}\LaTeX}

```

`\ConTeXt`   `ConTEXt`

```
7945 \newcommand*{\LWR@ConTeXt}
7946 {\InlineClass{latexlogofont}{Con}\TeX{}}%
7947 \InlineClass{latexlogofont}{t}}
```

`\BibTeX`   `BiBTEX`, *MakeIndex*  
`\MakeIndex`

```
7948 \providecommand*{\BibTeX}
7949 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}
7950
7951 \newcommand*{\MakeIndex}
7952 {\InlineClass{latexlogofont}{\textit{MakeIndex}}}
```

`\AmS`    $\mathcal{MS}$

`amslogo` is a CSS class used for the  $\mathcal{MS}$ logo.

```
7953 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}}
7954 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}
```

`\MiKTeX`   `MiKTEX`

```
7955 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}
```

`\LyX`   `LyX`

`lyxlogo` is a CSS class used for the `LyX`logo.

```
7956 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
```

```
7957 \end{warpHTML}
```

## 78.2 Print logos

**for PRINT output:**

```
7958 \begin{warpprint}
7959 \newcommand*{\XeTeXrevE}
7960 {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
7961 \providecommand*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
7962 \providecommand*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
7963 \providecommand*{\AmS}{%
7964 \leavevmode\hbox{$\mathcal{A}\kern-.2em\lower.376ex%
7965 \hbox{$\mathcal{M}$}\kern-.2em\mathcal{S}$}}
7966 \newcommand*{\LyX}{\textsf{LyX}}
```

```

7967 \providecommand*\LuaTeX{\mbox{Lua\TeX}}
7968 \providecommand*\LuaLaTeX{\mbox{Lua\LaTeX}}
7969 \providecommand*\BibTeX{\mbox{B\textsc{ib}\TeX}}
7970 \providecommand*\MakeIndex{\mbox{\textit{MakeIndex}}}
7971 \providecommand*\ConTeXt{\mbox{Con\TeX{t}}}
7972 \providecommand*\MiKTeX{\mbox{MiK\TeX}}
7973 \end{warpprint}

```

## 79 \AtBeginDocument, \AtEndDocument

**for HTML output:** 7974 \begin{warppHTML}

**\LWR@LwarpStart** Automatically sets up the HTML-related actions for the start and end of the document.

**\LWR@LwarpEnd**

```

7975 \AfterEndPreamble{\LWR@LwarpStart}
7976 \AtEndDocument{\LWR@LwarpEnd}

```

```

7977 \end{warppHTML}

```

## 80 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.

The following adjustments apply to the lwarp-\* package listings:



---

File 2 **lwarp-a4.sty**

§ 81 Package **a4**

Pkg a4 a4 is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{a4}  
2 \newcommand\*{\WideMargins}{}

---

File 3 **lwarp-a4wide.sty**

§ 82 Package **a4wide**

Pkg a4wide a4wide is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{a4wide}

---

File 4 **lwarp-a5comb.sty**

§ 83 Package **a5comb**

Pkg a5comb a5comb is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{a5comb}


---

File 5 **lwarp-abstract.sty**

§ 84 Package **abstract**

*(Based on original code by PETER WILSON.)*

Pkg abstract abstract is supported and patched by lwarp.

 **missing TOC** If using the number option with file splits, be sure to place the table of contents before the abstract. The number option causes a section break which may cause a

file split, which would put a table of contents out of the home page if it is after the abstract.

for HTML output: Accept all options for lwarp-abstract:

```

1 \LWR@ProvidesPackagePass{abstract}

2 \AtBeginDocument{
3 \BeforeBeginEnvironment{abstract}{
4 \LWR@forcenewpage
5 \BlockClass{abstract}
6 }
7 \AfterEndEnvironment{abstract}{\endBlockClass}
8 }
9
10 \renewcommand{\@bsrunintitle}{%
11 \hspace*{\abstitlestitle}%
12 {\abstractnamefont%
13 \InlineClass{abstractrunintitle}{\abstractname}%
14 \@bslabeldelim}%
15 }
16
17 \if@titlepage
18 \renewenvironment{abstract}{%
19 % \titlepage
20 \null\vfil
21 \@beginparpenalty\@lowpenalty
22 \if@bsrunin
23 \else
24 \if@bsstyle
25 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
26 \else
27 \ifnumber@bs
28 \num@bs
29 \else
30 \begin{\absnamepos}%
31 \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
32 \endparpenalty\@M
33 \end{\absnamepos}%
34 %% \vspace{\abstitlestitle}%
35 \fi
36 \fi
37 \vspace{\abstitlestitle}%
38 \fi
39 \put@bsintoc%
40 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
41 {\par\end{@bstr@ctlist}}\vfil\null%\endtitlepage
42 }
43 \else
44 \renewenvironment{abstract}{%

```

---

```

45 \if@bsrunin
46 \else
47 \if@bsstyle
48 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
49 \else
50 \ifnumber@bs
51 \num@bs
52 \else
53 \begin{\absnamepos}%
54 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
55 \end\absnamepos%
56 %% \vspace{\abstitleskip}%
57 \fi
58 \fi
59 \vspace{\abstitleskip}%
60 \fi
61 \put@bsintoc%
62 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
63 {\par\end{@bstr@ctlist}}
64 \fi
65

```

---

File 6 **lwarp-adjmulticol.sty**

## § 85 Package **adjmulticol**

Pkg `adjmulticol` `adjmulticol` is emulated.

Emulation similar to `multicols` is used, with adjusted margins. If the number of columns is specified as 1, it is set so, but if two or greater are used, `lwarp` allows a variable number of columns up to three.

**for HTML output:** `1 \LWR@ProvidesPackageDrop{adjmulticol}`

`2 \RequirePackage{multicol}`

`adjmulticols * {\<numcols>} {\<left margi>} {\<right margin>}`

`3 \NewDocumentEnvironment{adjmulticols}{s m m m}`  
`4 {%`

Compute the margins, and limit to positive only:

```

5 \setlength{\LWR@templengthone}{#3}%
6 \ifdimcomp{\LWR@templengthone}{<}{0pt}{\setlength{\LWR@templengthone}{0pt}}{}%
7 \setlength{\LWR@templengthtwo}{#4}
8 \ifdimcomp{\LWR@templengthtwo}{<}{0pt}{\setlength{\LWR@templengthtwo}{0pt}}{}%

```

If one column is specified, use a <div> of class singlecolumn, else use multicols:

```
9 \newcommand*{\LWR@mc colstype}{multicols}%
10 \ifnumcomp{#2}{=}{1}{\renewcommand*{\LWR@mc colstype}{singlecolumn}}{}}%
```

Help avoid page overflow:

```
11 \LWR@forcenewpage%
```

Print margins in points:

```
12 \uselengthunit{PT}%
```

Create the <div> with the given margin and class:

```
13 \BlockClass[%
14 margin-left:\rndprintlength{\LWR@templengthone} ; %
15 margin-right:\rndprintlength{\LWR@templengthtwo}%
16 ]{\LWR@mc colstype}%
17 }
18 {\endBlockClass}
```

---

File 7 **lwarp-addlines.sty**

\$ 86    Package    **addlines**

Pkg    addlines    addlines is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{addlines}

2 \newcommand*{\addlines[1][1]}{}
3 \let\addline\addlines
4 \newcommand*{\removelines[1][1]}{}
5 \let\removeline\removelines
```

---

File 8 **lwarp-afterpage.sty**

\$ 87    Package    **afterpage**

Pkg    afterpage    Emulated.

for HTML output:    Discard all options for lwarp-afterpage:

```
1 \LWR@ProvidesPackageDrop{afterpage}
```

---

```
2 \newcommand{\afterpage}[1]{#1}
```

---

## File 9 lwarp-algorithmicx.sty

### § 88 Package **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 240.1.

**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 \ifbool{FormatWP}{%
14 \setlength{\LWR@templengthone}{\the\ALG@thistlm}%
15 \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
16 \quad%
17 \addtolength{\LWR@templengthone}{-1em}%
18 }%
19 }{}%
20 \LWR@htmltagc{/span}%
21 }%
22 %
23 \let\origComment\Comment%
24 %
25 \renewcommand{\Comment}[1]{\InlineClass{floatright}{\origComment{#1}}}%
26 }

27 \end{warpHTML}
```

File 10 `lwarp-alltt.sty`

§ 89      Package **alltt**

Pkg    alltt    alltt is patched for use by lwarp.

```
for HTML output: 1 \LWR@ProvidesPackagePass{alltt}

2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching alltt.}
4 \AtBeginEnvironment{alltt}{%
5 \LWR@forcenewpage
6 \LWR@atbeginverbatim{alltt}\unskip\LWR@origvspace*{-\baselineskip}%
7 }
8 \AfterEndEnvironment{alltt}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
9 }
```

File 11 `lwarp-amsthm.sty`

§ 90      Package **amsthm**

*(Based on original code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)*

Pkg    amsthm    amsthm is patched for use by lwarp.

Table 12: AMSthm package — CSS styling of theorems and proofs

**Theorem:** <div> of class amsthmbody<theoremstyle>  
**Theorem Name:** <span> of class amsthmname<theoremstyle>  
**Theorem Number:** <span> of class amsthmnumber<theoremstyle>  
**Theorem Note:** <span> of class amsthmnote<theoremstyle>  
**Proof:** <div> of class amsthmproof  
**Proof Name:** <span> of class amsthmproofname  
where <theoremstyle> is plain, definition, etc.

```
for HTML output: 1 \LWR@ProvidesPackagePass{amsthm}
```

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
3 \renewcommand{\theoremstyle}[1]{%
4   \ifundefined{th@#1}{%
5     \PackageWarning{amsthm}{Unknown theoremstyle '#1'}%
6     \thm@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% new
8   }{%
9     \thm@style{#1}%
10    \renewcommand{\LWR@newtheoremstyle}{#1}% new
11  }%
12 }
```

Patched to remember the style for this theorem type:

```
13 \def\xnthm#1#2{%
14   \csedef{\LWR@thmstyle#2}{\LWR@newtheoremstyle}% new
15   \let\@tempa\relax
16   \exp\@ifdefinable\csname #2\endcsname{%
17     \global\exp\let\csname end#2\endcsname\@endtheorem
18     \ifx *#1% unnumbered, need to get one more mandatory arg
19       \edef\@tempa##1{%
20         \gdef\exp\@nx\csname#2\endcsname{%
21           \@nx\@thm{\exp\@nx\csname th@\the\thm@style\endcsname}%
22           {}{##1}}}%
23       \else % numbered theorem, need to check for optional arg
24         \def\@tempa{\@oparg{\ynthm{#2}}{}}%
25       \fi
26       \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{\LWR@thmstyle#2}}}% new
27   }%
28   \@tempa
29 }
```

Patched to enclose with css:

```
30 \newcommand{\LWR@haveamsthmname}{
31 \renewcommand{\thmname}[1]{\InlineClass{amsthmname\LWR@thisthmstyle}{##1}}
32 }
33
34 \newcommand{\LWR@haveamsthmnumber}{
35 \renewcommand{\thmnumber}[1]{\InlineClass{amsthmnumber\LWR@thisthmstyle}{##1}}
36 }
37
38 \newcommand{\LWR@haveamsthmnote}{
39 \renewcommand{\thmnote}[1]{\InlineClass{amsthmnote\LWR@thisthmstyle}{##1}}
```

```

40 }
41
42 \LWR@haveamsthmname
43 \LWR@haveamsthmnumber
44 \LWR@haveamsthmnote

```

Patches for CSS:

```

45 \def\@begintheorem#1#2[#3]{%
46 \LWR@forcenewpage% new
47   \BlockClass{amsthmbody\LWR@thisthmstyle}% new
48   \deferred@thm@head{
49     \the\thm@headfont \thm@indent
50     \@ifempty{#1}{\let\thmname\@gobble}{\LWR@haveamsthmname}% new
51     \@ifempty{#2}{\let\thmnumber\@gobble}{\LWR@haveamsthmnumber}% new
52     \@ifempty{#3}{\let\thmnote\@gobble}{\LWR@haveamsthmnote}% new
53     \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
54     \the\thm@headpunct~
55     \thmheadnl % possibly a newline.
56     \hskip\thm@headsep
57   }%
58   \ignorespaces}

```

Patched for CSS:

```

59 \def\@endtheorem{\endBlockClass\endtrivlist\@endpefalse }

```

Proof QED symbol:

```

60 \AtBeginDocument{
61 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
62 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
63 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
64 }

```

Patched for CSS:

```

65 \renewenvironment{proof}[1][\proofname]{\par
66 \LWR@forcenewpage% new
67   \BlockClass{amsthmproof}% new
68   \pushQED{\qed}%
69   \normalfont \topsep6\p@\@plus6\p@\relax
70   \trivlist
71   \item[\hskip\labelsep
72     \InlineClass{amsthmproofname}{#1\@addpunct{.}}]\ignorespaces% changes
73 }{%
74   \InlineClass{theoremendmark}{\popQED}\endtrivlist%
75   \endBlockClass% new
76   \@endpefalse

```




---

77 }

---

File 12 **lwarp-anonchap.sty**§ 91    Package **anonchap**

Pkg    anonchap    anonchap is emulated.

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

The code is shared by tocbibind.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{anonchap}

2 \newcommand{\simplechapter}[1][\@empty]{%
3   \def\@chapcntformat##1{%
4     #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
5   }%
6 }
7
8 \newcommand{\restorechapter}{%
9 \let\@chapcntformat\@secntformat%
10 }
```

---

File 13 **lwarp-anysize.sty**§ 92    Package **anysize**

Pkg    anysize    anysize is emulated.

**for HTML output:**

```


1 \LWR@ProvidesPackageDrop{anysize}

2 \def\papersize#1#2{}
3 \def\marginsize#1#2#3#4{}
```

---

File 14 **lwarp-appendix.sty**§ 93    Package **appendix**

Pkg    appendix    appendix is patched for use by lwarp.

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

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{appendix}

2 \renewcommand*{\@chap@pppage}{%
3 \part*{\appendixpagename}
4 \if@dotoc@pp
5 \addappheadtotoc
6 \fi
7 }
8
9 \renewcommand*{\@sec@pppage}{%
10 \part*{\appendixpagename}
11 \if@dotoc@pp
12 \addappheadtotoc
13 \fi
14 }
```

---

File 15 **lwarp-arabicfront.sty**

\$ 94     Package **arabicfront**

Pkg    arabicfront    arabicfront is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{arabicfront}

---

File 16 **lwarp-authblk.sty**

\$ 95     Package **authblk**

Pkg    authblk    authblk is patched for HTML.

**package support**    lwarp supports the native  $\LaTeX$  titling commands, and also supports the packages **load order** `authblk` and `titling`. If both are used, `authblk` should be loaded before `titling`.

**\published and \subtitle**    If using the `titling` package, additional `titlepage` fields for `\published` and `\subtitle` may be added by using `\AddSubtitlePublished` in the preamble. See section 53.7.

*(Based on original code by PATRICK W. DALY.)*

**for HTML output:**    Require that `authblk` be loaded before `titling`:

```

1 \@ifpackageloaded{titling}{
2 \PackageError{lwarp-authblk}
3 {Package authblk must be loaded before titling}
4 {Titling appends authblk's author macro, so authblk must be loaded first.}
5 }
6 {}

```

Load authblk:

```
7 \LWR@ProvidesPackagePass{authblk}
```

Patch to add a class for the affiliation:

```

8 \LetLtxMacro\LWRAB@affil\affil
9
10 \renewcommand{\affil}[2] [] {%
11 \LWRAB@affil[#1]{\protect\InlineClass{affiliation}{#2}}
12 }

```

Create an HTML break for an \authorcr:

```
13 \renewcommand*{\authorcr}{\protect\LWR@newlinebr}
```

---

#### File 17 **lwarp-balance.sty**

§ 96    Package **balance**

    Pkg    balance    Emulated.

for HTML output:    Discard all options for lwarp-balance:

```

1 \LWR@ProvidesPackageDrop{balance}

2 \newcommand*{\balance}{}
3 \newcommand*{\nobalance}{}

```

---

#### File 18 **lwarp-bigdelim.sty**

§ 97    Package **bigdelim**

    Pkg    bigdelim    bigdelim is used as-is for print or lateximage, and patched for HTML.

The delimiters are displayed in HTML by printing the delimiter, the text, and a thick border across the side of the \multirow which indicates the actual height of the

delimiter. The delimiter character is given a `<span>` class of `ldelim` or `rdelim`, and the default CSS sets this to `font-size:200%`

⚠ use `\mrowcell` `\ldelim` and `\rdelim` use `\multirow`, so `\mrowcell` must be used in the proper number of empty cells in the same column below `\ldelim` or `\rdelim`, but not in cells which are above or below the delimiter:

---

```
\begin{tabular}{lll}
<empty> & a & b \\
\ldelim{\{}{2}{.25in}[left ] & c & d \\
\mrowcell & e & f \\
<empty> & g & h \\
\end{tabular}
```

---

```

      a   b
left { c   d
     e   f
      g   h

```

---

for HTML output:

First, remove the temporary definitions of `\ldelim` and `\rdelim`, which were previously defined for tabular scanning in case `bigdelim` was not loaded:

```
1 \let\ldelim\relax
2 \let\rdelim\relax
```

Next, load the package's new definitions:

```
3 \LWR@ProvidesPackagePass{bigdelim}
```

Remember the print-mode versions:

```
4 \LetLtxMacro\LWR@origldelim\ldelim
5 \LetLtxMacro\LWR@origrdelim\rdelim
```

```
\ldelim {\langle 1:delimiter \rangle} {\langle 2:#rows \rangle} {\langle 3:width \rangle} [\langle 4:text \rangle]
\rdelim

6 \RenewDocumentCommand{\ldelim}{m m m O{}}{%
7 \renewcommand{\LWR@multirowborder}{right}%
8 \multirow{#2}{#3}{#4 \InlineClass{ldelim}{#1}}%
9 }
10
11 \RenewDocumentCommand{\rdelim}{m m m O{}}{%
12 \renewcommand{\LWR@multirowborder}{left}%
13 \multirow{#2}{#3}{\InlineClass{rdelim}{#1} #4}%
14 }
```

When entering a `lateximage`, restore the print-mode versions:

```
15 \appto{\LWR@restoreorigformatting}{%
16 \LetLtxMacro{\ldelim}{\LWR@origldelim}%
17 \LetLtxMacro{\rdelim}{\LWR@origrdelim}%
18 }
```

---

File 19 **lwarp-bigstrut.sty**

§ 98     Package **bigstrut**

Pkg   bigstrut   bigstrut is used as-is for print or `lateximage`, and patched for HTML.

for HTML output:

```
1 \LWR@ProvidesPackagePass{bigstrut}

2 \LetLtxMacro\LWR@origbigstrut\bigstrut
3
4 \renewcommand\bigstrut[1][x]{}
5
6 \appto{\LWR@restoreorigformatting}{%
7 \LetLtxMacro{\bigstrut}{\LWR@origbigstrut}%
8 }
```

---

File 20 **lwarp-bookmark.sty**

§ 99     Package **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]{}

```

---

File 21 **lwarp-booktabs.sty**

§ 100      Package **booktabs**

Pkg    booktabs    booktabs is emulated during HTML output, and used as-is during print output and inside an HTML lateximage.

**for HTML output:**    1 \LWR@ProvidesPackagePass{booktabs}

Booktabs emulation is spread among the tabular code. The original definitions are saved here for use in HTML lateximages. The HTML versions temporarily overwrite these print versions when tabular is started.

```
2 \LetLtxMacro\LWR@origtoprule\toprule
3 \LetLtxMacro\LWR@origmidrule\midrule
4 \LetLtxMacro\LWR@origcmidrule\cmidrule
5 \LetLtxMacro\LWR@origbottomrule\bottomrule
6 \LetLtxMacro\LWR@origaddlinespace\addlinespace
7 \LetLtxMacro\LWR@origmorecmidrules\morecmidrules
8 \LetLtxMacro\LWR@origspecialrule\specialrule
```

---

File 22 **lwarp-boxedminipage.sty**

§ 101      Package **boxedminipage**

Pkg    boxedminipage    boxedminipage is superseded by boxedminipage2e.

**for HTML output:**    1 \LWR@loadnever{boxedminipage}{boxedminipage2e}

---

File 23 **lwarp-boxedminipage2e.sty**

§ 102      Package **boxedminipage2e**

Pkg    boxedminipage2e    boxedminipage2e is emulated.

**for HTML output:**    Discard all options for lwarp-boxedminipage2e:

```
1 \LWR@ProvidesPackageDrop{boxedminipage2e}
```

---

```

2 \newenvironment{boxedminipage}{%
3 \begin{BlockClass}{framebox}%
4 \minipage%
5 }
6 {
7 \endminipage%
8 \end{BlockClass}
9 }

```

---

File 24 **lwarp-breakurl.sty**

§ 103      Package **breakurl**

Pkg    breakurl    breakurl is emulated.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{breakurl}

2 \LetLtxMacro\burl\url
3
4 \NewDocumentCommand{\LWR@burlaltb}{0}{ +m m}{%
5 \LWR@subhyperref{#3}{#2}% order reversed from \href
6 \endgroup%
7 \LWR@ensuredoingapar%
8 }
9
10 \DeclareRobustCommand*\burlalt{%
11 \LWR@ensuredoingapar%
12 \begingroup%
13 \catcode'\_ =12
14 \LWR@burlaltb%
15 }
16 \LetLtxMacro\urlalt\burlalt

```

---

File 25 **lwarp-caption2.sty**

§ 104      Package **caption2**

Pkg    caption2    caption2 is not used. The user is recommended to use caption instead.

**for HTML output:**

```

1 \LWR@loadnever{caption2}{caption}

```

---

File 26 **lwarp-ccaption.sty**

§ 105      Package **ccaption**

Pkg    ccaption    ccaption is not used. The user is recommended to use caption instead.

for HTML output:    1 \LWR@loadnever{ccaption}{caption}

---

File 27 **lwarp-changepage.sty**

§ 106      Package **changepage**

Pkg    changepage    changepage is ignored.

for HTML output:    Discard all options for lwarp-changepage:

```
1 \LWR@ProvidesPackageDrop{changepage}

2 \newif\ifoddpge
3 \DeclareRobustCommand{\checkoddpge}{\oddpgetrue}
4 \DeclareRobustCommand{\changetext}[5]{}
5 \DeclareRobustCommand{\changepage}[9]{}
6 \newenvironment{adjustwidth}[2]{}{}
7 \newenvironment{adjustwidth*}[2]{}{}
```

---

File 28 **lwarp-chngpage.sty**

§ 107      Package **chngpage**

Pkg    chngpage    chngpage is supersceded by changepage.

for HTML output:    1 \LWR@loadnever{chngpage}{changepage}

---

File 29 **lwarp-chappg.sty**

§ 108      Package **chappg**

Pkg    chappg    chappg is ignored.



**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{chappg}

2 \renewcommand{\pagenumbering}[2] [] {}
3 \providecommand{\chappgsep}{--}

```


---

### File 30 **lwarp-color.sty**

#### \$ 109 Package **color**

Pkg color Allowed but ignored. xcolor is then required as well.

color is superceded by xcolor, and lwarp requires several of the features of xcolor.

 **missing colors** It should be sufficient for the user's document to load color then load xcolor as well.

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{color}
2 \RequirePackage{xcolor}

```

---

### File 31 **lwarp-crop.sty**

#### \$ 110 Package **crop**

Pkg crop Emulated.

**for HTML output:** Discard all options for lwarp-crop:

```

1 \LWR@ProvidesPackageDrop{crop}

2 \newcommand*{\crop}[1] [] {}
3 \newcommand*{\cropdef}[6] [] {}

```

---

### File 32 **lwarp-cuted.sty**

#### \$ 111 Package **cuted**

Pkg cuted cuted is emulated.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{cuted}

2 \newenvironment{strip}{}{}
3 \newskip\stripsep
4 \def\oldcolsbreak#1{}

```

---

File 33 **lwarp-cutwin.sty**

§ 112      Package **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}}
```

---

File 34 **lwarp-dblfnote.sty**

§ 113      Package **dblfnote**

Pkg    dblfnote    dblfnote is emulated.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{dblfnote}

2 \newcounter{DFNsloppiness}
3 \newdimen\DFNcolumnsep
4 \newdimen\DFNcolumnwidth
5 \def\DFNallowcbreak{}
6 \def\DFNinhibitcbreak{}
7 \def\DFNtrysingle{}
8 \def\DFNalwaysdouble{}
9 \def\DFNruleboth{}
10 \def\DFNruleleft{}

```

---

File 35 **lwarp-dcolumn.sty**

§ 114    Package **dcolumn**

Pkg    dcolumn    dcolumn is emulated during HTML output, and the dcolumn package is ignored.

```

1 \LWR@ProvidesPackageDrop{dcolumn}

```

---

File 36 **lwarp-draftwatermark.sty**

§ 115    Package **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]{}

```

---

**File 37 lwarp-ebook.sty****§ 116 Package ebook**

Pkg ebook ebook is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{ebook}

2 \setcounter{secnumdepth}{0}
3 \setcounter{tocdepth}{2}
4
5 \providecommand{\pagefill}[1][0.001mm]{\noindent}
6
7 \providecommand{\ebook}{
8 \setcounter{secnumdepth}{0}
9 \setcounter{tocdepth}{2}
10 }
```

---

**File 38 lwarp-ellipsis.sty****§ 117 Package 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}
```

---

**File 39 lwarp-emptypage.sty****§ 118 Package emptypage**

Pkg emptypage emptypage is ignored.

for HTML output: Discard all options for lwarp-emptypage:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

---

File 40 **lwarp-endnotes.sty**

§ 119      Package **endnotes**

*(Based on original code by JOHN LAVAGNINO.)*

Pkg **endnotes**      Used as-is.

[table of contents](#)      To place the endnotes in the TOC, use:

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

[HTML page](#)      To additionally have the endnotes on their own HTML page, if FileDepth allows:

```
\ForceHTMLPage
\theendnotes
```

**for HTML output:**      1 \LWR@ProvidesPackagePass{endnotes}

```
2 \def\enoteformat{%
3 % \rightskip\z@ \leftskip\z@ \parindent=1.8em
4 \leavevmode
5 % \llap{
6 \makeenmark
7 % }
8 }
9
10 \def\@makeenmark{\hbox{\LWR@htmlspan{sup}{\normalfont\theenmark}}}
11 \def\makeenmark{\@makeenmark}
```

---

File 41 **lwarp-enumerate.sty**

§ 120      Package **enumerate**

Pkg **enumerate**      enumerate is supported with no changes.

This package is only required because it was used in the past to drop and then emulate the package. It cannot be removed because an older version which dropped the package may still remain, for example in a local vs. distribution directory, but it is now supported directly by lwarp and thus must no longer be dropped.

---

**for HTML output:** 1 \LWR@ProvidesPackagePass{enumerate}

---

File 42 **lwarp-enumitem.sty**

§ 121 Package **enumitem**

Pkg **enumitem** enumitem is supported with minor adjustments.

**for HTML output:** 1 \LWR@ProvidesPackagePass{enumitem}

**for HTML output:** 2 \begin{warpHTML}

\newlist {<name>} {<type>} {<maxdepth>}  
 \renewlist {<name>} {<type>} {<maxdepth>}

For enumitem lists, new lists must have the start and end actions assigned to the new environment. Renewed lists already have their actions assigned, and thus need no changes.

```
3 \let\LWR@orignewlist\newlist
4
5 \renewcommand*\newlist{[3]{%
6 \LWR@orignewlist{#1}{#2}{#3}%
7 \AtBeginEnvironment{#1}{\csuse{LWR@#2start}}}%
8 \AtEndEnvironment{#1}{\csuse{LWR@#2end}}}%
9 }

10 \end{warpHTML}
```

---

File 43 **lwarp-epigraph.sty**

§ 122 Package **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 \ifbool{FormatWP}
7 {\begin{BlockClass}[border-top:1px solid gray]{epigraphsource}}
```

---

```

8 {\begin{BlockClass}{epigraphsource}}
9 #2
10 \end{BlockClass}
11 \end{BlockClass}
12 }

13 \newcommand{\epigraph}[2]
14 {
15 \begin{LWR@BlockClassWP}{text-align:right}{\epigraph}
16 \qitem{#1}{#2}
17 \end{LWR@BlockClassWP}
18 }
19
20 \newenvironment*{epigraphs}
21 {\LWR@BlockClassWP{text-align:right}{\epigraph}}
22 {\endLWR@BlockClassWP}

```

Use CSS to format epigraphs.

The following are null commands for source compatibility:

```

23 \newlength{\epigraphwidth}
24 \setlength{\epigraphwidth}{.5\linewidth}
25 \newenvironment*{flushepinormal}{}{}
26 \newcommand{\textflush}[1]{flushepinormal}
27 \newcommand{\epigraphflush}[1]{flushright}
28 \newcommand{\sourceflush}[1]{flushright}
29 \newcommand*{\epigraphsize}{\small}
30 \newlength{\epigraphrule}
31 \newlength{\beforeepigraphskip}
32 \newlength{\afterepigraphskip}
33 \newcommand{\epigraphhead}[2][0]{#2}
34 \newcommand{\dropchapter}[1]{}
35 \newcommand*{\undodrop}{}
36 \newcommand{\cleartoevenpage}[1][{}]{ }

```

---

File 44 **lwarp-eso-pic.sty**

## § 123 Package **eso-pic**

Pkg eso-pic eso-pic is emulated during HTML output, and the eso-pic package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{eso-pic}

2 \newcommand*{\LenToUnit}{}
3 \newcommand{\AtPageUpperLeft}[1]{}

```

---

```

4 \newcommand{\AtPageLowerLeft}[1]{}
5 \newcommand{\AtPageCenter}[1]{}
6 \newcommand{\AtStockLowerLeft}[1]{}
7 \newcommand{\AtStockUpperLeft}[1]{}
8 \newcommand{\AtStockCenter}[1]{}
9 \newcommand{\AtTextUpperLeft}[1]{}
10 \newcommand{\AtTextLowerLeft}[1]{}
11 \newcommand{\AtTextCenter}[1]{}
12 \NewDocumentCommand{\AddToShipoutPictureBG}{s +m}{}
13 \renewcommand{\AddToShipoutPicture}{\AddToShipoutPictureBG}
14 \NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
15 \newcommand*{\ClearShipoutPictureBG}{}
16 \newcommand*{\ClearShipoutPicture}{}
17 \newcommand*{\ClearShipoutPictureFG}{}
18 \newcommand{\gridSetup}[6]{}

```

---

File 45 **lwarp-everypage.sty**

§ 124     Package **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]{}

```

---

File 46 **lwarp-everyshi.sty**

§ 125     Package **everyshi**

Pkg    everyshi    Emulated.

**for HTML output:**    Discard all options for lwarp-everyshi:

```

1 \LWR@ProvidesPackageDrop{everyshi}

2 \newcommand*{\EveryShipout}[1]{}
3 \newcommand*{\AtNextShipout}[1]{}

```



---

File 47 **lwarp-extramarks.sty**

§ 126      Package **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}{}

```

---

File 48 **lwarp-fancybox.sty**

§ 127      Package **fancybox**

*(Based on original code by TIMOTHY VAN ZANDT.)*

Pkg    fancybox    fancybox is supported with some patches.

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

```

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

```

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

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

```

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

```

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

```

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

```

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

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

**for HTML output:** `1 \begin{warpHTML}`

```

2 \LWR@ProvidesPackagePass{fancybox}

3 \renewcommand*{\@shadowbox}[1]{%
4 \ifbool{FormatWP}%
5 {\InlineClass[border:1px solid black]{shadowbox}{#1}}%
6 {\InlineClass{shadowbox}{#1}}%
7 }
8
9 \renewcommand*{\@doublebox}[1]{%
10 \ifbool{FormatWP}%
11 {\InlineClass[border:1px double black]{doublebox}{#1}}%
12 {\InlineClass{doublebox}{#1}}%
13 }
14
15 \renewcommand*{\@ovalbox}[2]{%
16 \ifbool{FormatWP}%
17 {\InlineClass[border:1px solid black; border-radius:1ex]{ovalbox}{#2}}%
18 {%
19 \ifthenelse{\isequivalentto{#1}{\thinlines}}%
20 {\InlineClass{ovalbox}{#2}}%
21 {\InlineClass{Ovalbox}{#2}}%
22 }%
23 }

```

Convert minipages, parboxes, and lists into linear text using the LWR@nestspan environment:

```

24 \let\LWR@origSbox\Sbox
25
26 \def\Sbox{\LWR@origSbox\LWR@nestspan}
27
28
29 \let\LWR@origendSbox\endSbox
30
31 \def\endSbox{\endLWR@nestspan\LWR@origendSbox}

```

Beqnarray is adapted for MathJax or enclosed inside a lateximage:

```

32 \RenewEnviron{Beqnarray}
33 {\LWR@eqnarrayfactor}
34
35 \csgpreto{Beqnarray*}{\boolfalse{LWR@numbereqnarray}}

```

\GenericCaption is enclosed in an HTML block:

```

36 \renewcommand{\GenericCaption}[1]{%
37 \LWR@figcaption%
38 #1%

```

```
39 \endLWR@figcaption%
40 }
```

Btrivlist is enclosed in an HTML block:

```
41 \RenewDocumentEnvironment{Btrivlist}{m o}
42 {\begin{BlockClass}{Btrivlist}\tabular{#1}}
43 {\endtabular\end{BlockClass}}
```

Btrivlist is also neutralized when used inside a span:

```
44 \AtBeginEnvironment{LWR@nestspan}{%
45 \RenewDocumentEnvironment{Btrivlist}{m o}{-}{-}%
46 \RenewDocumentCommand{\LWR@origitem}{d()}{\LWRFB@origitem}%
47 }
```

lwarp's handling of \item is patched to accept fancybox's optional arguments:

```
48 \let\LWRFB@origitemizeitem\LWR@itemizeitem
49 \let\LWRFB@origdescitem\LWR@descitem
50 \LetLtxMacro{\LWRFB@origitem}{\LWR@origitem}
51
52 \RenewDocumentCommand{\LWR@itemizeitem}{d()}{\LWRFB@origitemizeitem}
53 \RenewDocumentCommand{\LWR@descitem}{d()}{\LWRFB@origdescitem}
```

The various boxed lists become regular lists:

```
54 \renewenvironment{Bitemize}[1][\begin{itemize}]{\end{itemize}}
55 \renewenvironment{Benumerate}[1][\begin{enumerate}]{\end{enumerate}}
56 \renewenvironment{Bdescription}[1][\begin{description}]{\end{description}}
```

\boxput simply prints one then the other argument, side-by-side instead of above and behind:

```
57 \RenewDocumentCommand{\boxput}{s d() m m}{%
58 \IfBooleanTF{#1}{#3\quad#4}{#4\quad#3}%
59 }
```

Neutralized commands:

```
60 \RenewDocumentCommand{\fancyput}{s d() m}{-}
61 \RenewDocumentCommand{\thisfancyput}{s d() m}{-}
62
63 \RenewDocumentCommand{\fancypage}{m m}{-}
64 \RenewDocumentCommand{\thisfancypage}{m m}{-}
65
```

```

66 \def\LandScape#1{}
67 \def\endLandScape{}
68 \def\@Landscape#1#2#3{}
69 \def\endLandscape{}

```

Low-level patches for Verbatim, VerbatimInput, UseVerbatim:

```

70 \let\LWRFB@UseVerbatim\UseVerbatim
71 \renewcommand*\{UseVerbatim}[1]{%
72 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
73 \LWRFB@UseVerbatim{#1}%
74 \LWR@afterendverbatim%
75 }
76
77 \let\LWRFB@LUseVerbatim\LUseVerbatim
78
79 \renewcommand*\{LUseVerbatim}[1]{%
80 \LWR@atbeginverbatim{LVerbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
81 \noindent%
82 \LWRFB@LUseVerbatim{#1}%
83 \LWR@afterendverbatim%
84 }
85
86 \def\@BUseVerbatim[#1]#2{%
87 \LWR@atbeginverbatim{BVerbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
88 \LWRFB@UseVerbatim{#2}%
89 \LWR@afterendverbatim%
90 }

91 \end{warpHTML}

```

---

File 49 **lwarp-fancyhdr.sty**

§ 128     Package **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}[2] [] {}

```

---

```

7 \newcommand*\chead}[2] [] {}
8 \newcommand*\rhead}[2] [] {}
9 \newcommand*\lfoot}[2] [] {}
10 \newcommand*\cfoot}[2] [] {}
11 \newcommand*\rfoot}[2] [] {}
12 \newcommand*\headrulewidth[1] {}
13 \newcommand*\footrulewidth[1] {}
14 \newcommand*\fancyheadoffset}[2] [] {}
15 \newcommand*\fancyfootoffset}[2] [] {}
16 \newcommand*\fancyhfoffset}[2] [] {}
17 \newcommand*\iffloatpage}[2] {#2}
18 \newcommand*\ifftopfloat}[2] {#2}
19 \newcommand*\iffbotfloat}[2] {#2}

```

---

File 50 **lwarp-fancyvrb.sty**

§ 129     Package **fancyvrb**

*(Based on original code by TIMOTHY VAN ZANDT.)*

Pkg fancyvrb fancyvrb is supported with some patches.

for HTML output: 

```
1 \RequirePackage{xcolor}% for \convertcolorspec
2 \LWR@ProvidesPackagePass{fancyvrb}
```

for HTML output: 

```
3 \begin{warpHTML}
```

Initial default patch for fancyvrb:

```
4 \fvset{frame=none}%
```

For \VerbatimFootnotes:

```

5 \renewcommand{\VerbatimFootnotes}{
6 \PackageError{lwarp}
7 {Verbatim footnotes are not yet supported by lwarp.}
8 {This may be improved some day.}
9 }

```

After the preamble is loaded, after any patches to Verbatim:

```

10 \AfterEndPreamble{
11 \LWR@traceinfo{Patching Verbatim.}

```

Remember the original definition of Verbatim:

```
12 \let\LWRFV@origVerbatim\Verbatim
```

Env **Verbatim** Patched to place the environment in a `fancyvrb` div, and the label in a `fancyvrblabel` div. Also corrects the left margin for line numbers. Also uses `VerbatimHTMLWidth` to control placement of line numbers on the right. Aligning the right margin requires knowing the width.

```
13 \renewcommand*\Verbatim}{%
14 \LWR@forcenewpage
15 \LWRFV@origVerbatim%
16 }
```

`\LWR@FVstyle` Holds the style of the verbatim.

```
17 \newcommand*\LWR@FVstyle}{}
```

The following patches to `Verbatim` are executed at the start and end of the environment, depending on the choice of frame. Original code is from the `fancyvrb` package.

```
18 \newcommand*\LWR@fvstartnone}{%
19 \LWR@traceinfo{fvstartnone}%
20 \hbox to\z@{\LWR@atbeginverbatim[\LWR@FVstyle]{verbatim}}%
21 }
22
23 \newcommand*\LWR@fvendnone}{%
24 \LWR@traceinfo{fvendnone}%
25 \hbox to\z@{\LWR@afterendverbatim}%
26 }
27
28 \newcommand*\LWR@fvstartsingle}{%
29 \LWR@traceinfo{fvstartsingle}%
30 \LWR@fvstartnone%
31 \FV@BeginListFrame@Single%
32 }
33
34 \newcommand*\LWR@fvendsingle}{%
35 \LWR@traceinfo{fvendsingle}%
36 \FV@EndListFrame@Single%
37 \LWR@fvendnone%
38 }
39
40 \newcommand*\LWR@fvstartline}{%
41 \LWR@traceinfo{fvstartline}%
42 \LWR@fvstartnone%
43 \FV@BeginListFrame@Lines%
44 }
45
46 \newcommand*\LWR@fvendline}{%
47 \LWR@traceinfo{fvendline}%
```

```

48 \FV@EndListFrame@Lines%
49 \LWR@fvendnone%
50 }

```

The following patches select the start/left/right/end behaviors depending on frame. Original code is from the fancyvrb package.

```

51
52 \newcommand*{\LWR@FVfindcurrentcolor}{%
53 \protect\colorlet{\LWR@current@color}{.}%
54 \protect\convertcolourspec{named}{\LWR@current@color}{HTML}\LWR@tempcolor%
55 }
56
57 \newcommand*{\LWR@FVtextstyle}{%
58 \LWR@FVfindcurrentcolor
59 \ifdefstring{\LWR@tempcolor}{000000}%
60 {}%
61 {color: \#\LWR@tempcolor ; }%
62 }
63
64
65 \newcommand*{\LWR@FVfindbordercolor}{%
66 \FancyVerbRuleColor%
67 \LWR@FVfindcurrentcolor
68 \color{black}
69 }
70
71 % border width of \FV@FrameRule
72 \newcommand*{\LWR@FVborderstyle}[1]{%
73 padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ;
74 \LWR@FVfindbordercolor
75 border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt solid \#\LWR@tempcolor ;
76 }
77
78 \def\FV@Frame@none{%
79 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle}%
80 \let\FV@BeginListFrame\LWR@fvstartnone%
81 \let\FV@LeftListFrame\relax%
82 \let\FV@RightListFrame\relax%
83 \let\FV@EndListFrame\LWR@fvendnone}
84
85 \FV@Frame@none% default values
86
87 \def\FV@Frame@single{%
88 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{}}%
89 \let\FV@BeginListFrame\LWR@fvstartsingle%
90 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
91 \let\FV@RightListFrame\FV@RightListFrame@Single%
92 \let\FV@EndListFrame\LWR@fvendsingle}

```



```

93
94 \def\FV@Frame@lines{%
95 \renewcommand*{\LWR@FVstyle}{%
96   \LWR@FVtextstyle\LWR@FVborderstyle{-top}\LWR@FVborderstyle{-bottom}%
97 }%
98 \let\FV@BeginListFrame\LWR@fvstartline%
99 \let\FV@LeftListFrame\relax%
100 \let\FV@RightListFrame\relax%
101 \let\FV@EndListFrame\LWR@fvendline}
102
103 \def\FV@Frame@topline{%
104 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-top}}%
105 \let\FV@BeginListFrame\LWR@fvstartline%
106 \let\FV@LeftListFrame\relax%
107 \let\FV@RightListFrame\relax%
108 \let\FV@EndListFrame\LWR@fvendnone}
109
110 \def\FV@Frame@bottomline{%
111 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-bottom}}%
112 \let\FV@BeginListFrame\LWR@fvstartnone%
113 \let\FV@LeftListFrame\relax%
114 \let\FV@RightListFrame\relax%
115 \let\FV@EndListFrame\LWR@fvendline}
116
117 \def\FV@Frame@leftline{%
118 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-left}}%
119 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
120 \ifx\FancyVerbFillColor\relax%
121 \let\FV@FrameFillLine\relax%
122 \else%
123 \@tempdima\FV@FrameRule\relax%
124 \multiply\@tempdima-\tw@%
125 \edef\FV@FrameFillLine{%
126 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}}%
127 \kern-\number\@tempdima sp}}%
128 \fi%
129 \let\FV@BeginListFrame\LWR@fvstartnone%
130 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
131 \let\FV@RightListFrame\relax%
132 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the fancyvrb package.

```

133 \def\FV@SingleFrameLine#1{%
134   \hbox to\z@{%
135     \kern\leftmargin
136     \ifnum#1=\z@\relax
137     \let\FV@Label\FV@LabelBegin

```

```

138   \else
139     \let\FV@Label\FV@LabelEnd
140   \fi
141   \ifx\FV@Label\relax
142 %     \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
143   \else
144     \ifnum#1=\z@
145 %       \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
146       \ifx\FV@LabelPositionTopLine\relax
147         \else
148           \LWR@FVfindbordercolor
149           \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
150           \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
151           \LWR@htmltagc{/div}
152         \fi
153       \else
154 %       \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
155       \ifx\FV@LabelPositionBottomLine\relax
156         \else
157           \LWR@FVfindbordercolor
158           \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
159           \LWR@origtextrm{\FV@LabelEnd}
160           \LWR@htmltagc{/div}
161         \fi
162       \fi
163     \fi
164   \fi
165   \hss
166 }
167 }

```

Processes each line, adding optional line numbers. Original code is from the fancyvrb package.

```

168 \def\FV@ListProcessLine#1{%
169   \hbox to \hsize{%
170 %     \kern\leftmargin
171     \hbox to \VerbatimHTMLWidth {%
172       \ifcvoid\FV@LeftListNumber}{\kern 2.5em}%
173       \FV@LeftListNumber%
174 %     \FV@LeftListFrame
175     \FancyVerbFormatLine{#1}%
176     \hss%
177 %     \FV@RightListFrame
178     \FV@RightListNumber%
179   }%
180   \hss% required to avoid underfull hboxes
181 }
182 }

```

Env **BVerbatim**

```

183 \AtBeginEnvironment{BVerbatim}
184 {
185 \LWR@forcenewpage
186 \LWR@atbeginverbatim{bverbatim}
187
188 }
189
190 \AfterEndEnvironment{BVerbatim}
191 {
192 \leavevmode\par\LWR@origvspace{-\baselineskip}
193 \LWR@afterendverbatim
194 }

```

End of the modifications to make at the end of the preamble:

```

195 } % \AfterEndPreamble

196 \end{warpHTML}

```

---

File 51 **lwarp-figcaps.sty**

§ 130     Package **figcaps**

Pkg **figcaps**     Emulated.

**for HTML output:**     Discard all options for lwarp-figcaps:

```

1 \LWR@ProvidesPackageDrop{figcaps}

2 \newcommand*{\figcapson}{}
3 \newcommand*{\figcapsoff}{}
4 \newcommand*{\printfigures}{}
5 \newcommand*{\figmarkon}{}
6 \newcommand*{\figmarkoff}{}
7 \def\figurecapname{Figure Captions}
8 \def\tablepagename{Tables}
9 \def\figurepagename{Figures}

```

---

File 52 **lwarp-fix2col.sty**

§ 131 Package **fix2col**

Pkg fix2col fix2col is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{fix2col}

---

File 53 **lwarp-float.sty**

§ 132 Package **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 62.2 for the \listof command.

**\newfloat**  $\{\langle 1: type \rangle\} \{\langle 2: placement \rangle\} \{\langle 3: ext \rangle\} [\langle 4: within \rangle]$

Emulates the \newfloat command from the float package.

“placement” is ignored.

```
2 \NewDocumentCommand{\newfloat}{m m m o}{%
3 \IfValueTF{#4}
4 {\DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}}
5 {\DeclareFloatingEnvironment[fileext=#3]{#1}}
```

newfloat package automatically creates the \listof command for new floats, but float does not, so remove \listof here in case it is manually created later.

```
6 \cslet{listof#1s}\relax
7 \cslet{listof#1es}\relax
8 }
```

**\floatname**  $\{\langle type \rangle\} \{\langle name \rangle\}$

Sets the text name of the float, such as “Figure”.

```
9 \NewDocumentCommand{\floatname}{m +m}{%
```

```
10 \SetupFloatingEnvironment{#1}{name=#2}%
11 }
```

`\floatplacement`  $\{\langle type \rangle\} \{\langle placement \rangle\}$

Float placement is ignored.

```
12 \newcommand*\floatplacement}[2]{%
13 \SetupFloatingEnvironment{#1}{placement=#2}%
14 }
```

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

Float styles are ignored.

```
15 \newcommand*\floatstyle}[1]{%
16 }
```

`\restylefloat`  $* \{\langle style \rangle\}$

Float styles are ignored.

```
17 \NewDocumentCommand{\restylefloat}{s m}{%
18 }
```

---

File 54 **lwarp-floatflt.sty**

§ 133 Package **floatflt**

Pkg floatflt Emulated.

for HTML output: Discard all options for lwarp-floatflt:

```
1 \LWR@ProvidesPackageDrop{floatflt}
```

Env [ $\langle \rangle$ ] `offset`  $\{\langle type \rangle\} \{\langle width \rangle\}$  Borrowed from the lwarp version of keyfloat:

```
2 \NewDocumentEnvironment{KFLTfloatflt@marginfloat}{0{-1.2ex} m m}
3 {%
4 \setlength{\LWR@templengthone}{#3}%
5 \uselengthunit{PT}%
6 \LWR@BlockClassWP{%
7   float:right; %
8   width:\rndprintlength{\LWR@templengthone}; %
```

---

```

9     margin:10pt%
10 }{%
11     width:\rndprintlength{\LWR@templengthone}%
12 }%
13 {marginblock}%
14 \captionsetup{type=#2}%
15 }
16 {%
17 \endLWR@BlockClassWP%
18 }

```

```

Env floatingfigure  [⟨placement⟩] {⟨width⟩}

19 \DeclareDocumentEnvironment{floatingfigure}{o m}
20   {\begin{KFLTfloatflt@marginfloat}{figure}{#2}}
21   {\end{KFLTfloatflt@marginfloat}}

```

```

Env floatingtable  [⟨placement⟩]

22 \DeclareDocumentEnvironment{floatingtable}{o}
23   {\begin{KFLTfloatflt@marginfloat}{table}{1.5in}}
24   {\end{KFLTfloatflt@marginfloat}}

```

---

File 55 **lwarp-floatpag.sty**

§ 134    Package **floatpag**

Pkg floatpag    Emulated.

for HTML output: Discard all options for lwarp-floatpag:

```

1 \LWR@ProvidesPackageDrop{floatpag}

2 \newcommand*{\floatpagestyle}[1]{%
3 \newcommand*{\rotfloatpagestyle}[1]{%
4 \newcommand*{\thisfloatpagestyle}[1]{%

```

---

File 56 **lwarp-floatrow.sty**

§ 135    Package **floatrow**

Pkg floatrow    floatrow is emulated during HTML output, and the floatrow package is ignored.

for HTML output:    1 \LWR@ProvidesPackageDrop{floatrow}

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

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

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

After everything has loaded, remember whether subcaption was loaded. If not, it is assumed that subfig is used instead:

```
2 \newbool{LWR@subcaptionloaded}
3
4 \AtBeginDocument{
5 \ifpackageloaded{subcaption}
6 {\booltrue{LWR@subcaptionloaded}}
7 {\boolfalse{LWR@subcaptionloaded}}
8 }
```

**\floatbox** [*1 preamble*] [*2 captype*] [*3 width*] [*4 height*] [*5 vert pos*] [*6 caption*] [*7 object*]

Only parameters for captype, width, caption, and object are used.

LWR@insubfloatrow is true if inside a subfloatrow environment.

There are two actions, depending on the use of subcaption or subfig.

```
9 \NewDocumentCommand{\floatbox}{o m o o o +m +m}{%
10 \ifbool{LWR@subcaptionloaded}%
11 {% subcaption
```

For subcaption:

```
12 \ifbool{LWR@insubfloatrow}%
13 {% subcaption in a subfloatrow
```

subfigure and subtable environments take width as an argument.

```
14 \IfValueTF{#3}%
15 {\@nameuse{sub#2}{#3}}%
16 {\@nameuse{sub#2}{\linewidth}}%
```

```

17    }% subcaption in a subfloatrow
18    {% subcaption not in subfloatrow

```

figure and table environments do not take a width argument.

```

19        \@nameuse{#2}%
20    }% subcaption not in subfloatrow
21    #6
22
23    #7

```

End the environments:

```

24    \ifbool{LWR@insubfloatrow}%
25    {\@nameuse{endsub#2}}%
26    {\@nameuse{end#2}}%
27}% subcaption
28{% assume subfig

```

For subfig:

```

29\ifbool{LWR@insubfloatrow}%
30{% subfig in a subfloatrow

```

\subfloat is a macro, not an environment.

Package subfig's \subfloat command takes an optional argument which is the caption, but \floatbox argument #6 contains commands to create the caption and label, not the caption itself. Thus, \caption is temporarily disabled to return its own argument without braces.

```

31    \begingroup
32    \let\caption\@firstofone
33    \subfloat[#6]{#7}
34    \endgroup
35}% subfig in a subfloatrow
36{% subfig package, but not a subfig

```

figure and table are environments:

```

37 \@nameuse{#2}
38 #6
39
40 #7
41 \@nameuse{end#2}
42}% subfig package, but not a subfig
43}% assume subfig
44}

```

Not used:

```

45 \newcommand*{\nocapbeside}{}
46 \newcommand*{\capbeside}{}
47 \newcommand*{\captop}{}

```



```

48 \newlength{\FBwidth}
49 \setlength{\FBwidth}{.3\linewidth}
50 \newlength{\FBheight}
51 \setlength{\FBheight}{2in}
52 \newcommand*{\useFCwidth}{}
53 \newcommand{\floatsetup}[2][{}]{
54 \newcommand{\thisfloatsetup}[1]{
55 \newcommand{\clearfloatsetup}[1]{
56 \newcommand*{\killfloatstyle}{}

```

`\newfloatcommand`     $\langle 1 \text{ command} \rangle$   $\langle 2 \text{ captype} \rangle$   $\langle 3 \text{ preamble} \rangle$   $\langle 4 \text{ default width} \rangle$

Preamble and default width are ignored.

```

57 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
58 \@namedef{#1}{
59 \floatbox{#2}
60 }
61 }

```

`\renewfloatcommand`     $\langle 1 \text{ command} \rangle$   $\langle 2 \text{ captype} \rangle$   $\langle 3 \text{ preamble} \rangle$   $\langle 4 \text{ default width} \rangle$

Preamble and default width are ignored.

```

62 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
63 \@namedef{#1}{%
64 \floatbox{#2}
65 }
66 }

```

`\ffigbox`     $\langle width \rangle$   $\langle height \rangle$   $\langle vposn \rangle$   $\langle caption commands \rangle$   $\langle contents \rangle$

```

67 \newfloatcommand{ffigbox}{figure}[\nocapbeside][

```

`\ttabbox`     $\langle width \rangle$   $\langle height \rangle$   $\langle vposn \rangle$   $\langle caption commands \rangle$   $\langle contents \rangle$

```

68 \newfloatcommand{ttabbox}{table}[\capttop][\FBwidth]

```

`\fcapside`     $\langle width \rangle$   $\langle height \rangle$   $\langle vposn \rangle$   $\langle caption commands \rangle$   $\langle contents \rangle$

```

69 \newfloatcommand{fcapside}{figure}[\capbeside][

```

Env    `floatrow`     $\langle numfloats \rangle$

The row of floats is placed into a `<div>` of class `floatrow`.

```

70 \newenvironment*{floatrow}[1][2]
71 {
72 \LWR@forcenewpage
73 \BlockClass{floatrow}

```

While inside the floatrow, divide the `\linewidth` by the number of floats.

```
74 \booltrue{LWR@infloatrow}
75 \setlength{\linewidth}{6in/#1}
76 }
77 {
78 \boolfalse{LWR@infloatrow}
79 \endBlockClass
80 }
```

Keys for `\DeclareNewFloatType`:

```
81 \newcommand*{\LWR@frowkeyplacement}{}
82 \newcommand*{\LWR@frowkeyname}{}
83 \newcommand*{\LWR@frowkeyfileext}{}
84 \newcommand*{\LWR@frowkeywithin}{}
85 \newcommand*{\LWR@frowkeycapstyle}{}
86
87 \define@key{frowkeys}{placement}{}%
88 \define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}%
89 \define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}%
90 \define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}%
91 \define@key{frowkeys}{relatedcapstyle}{}%
```

`\DeclareNewFloatType`    *{<type>}{<options>}*

Use `\listof{type}{Title}` to print a list of the floats.

```
92 \newcommand*{\DeclareNewFloatType}[2]{%
```

Reset key values:

```
93 \renewcommand*{\LWR@frowkeyplacement}{}
94 \renewcommand*{\LWR@frowkeyname}{}
95 \renewcommand*{\LWR@frowkeyfileext}{}
96 \renewcommand*{\LWR@frowkeywithin}{}
97 \renewcommand*{\LWR@frowkeycapstyle}{}%
```

Read new key values:

```
98 \LWR@traceinfo{about to setkeys frowkeys}%
99 \setkeys{frowkeys}{#2}%
100 \LWR@traceinfo{finished setkeys frowkeys}%
```

Create a new float with optional [within]:

```
101 \ifthenelse{\equal{\LWR@frowkeywithin}{}}{
102 {
103   \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\
104     \LWR@frowkeyfileext}%
105   \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}
106 }%
107 {%
```

```

108 \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\
109 \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
110 \newfloat{#1}{\LWR@frowkeyplacement}%
111 {\LWR@frowkeyfileext}[\LWR@frowkeywithin]%
112 \LWR@traceinfo{finished newfloat #1}
113 }%

```

Rename the float if a name was given:

```

114 \ifthenelse{\equal{\LWR@frowkeyname}{}}{
115 {}
116 {\floatname{#1}{\LWR@frowkeyname}}}%
117 }

```

Not used:

```

118 \newcommand{\buildFBBBOX}[2]{}
119 \newcommand*{\CenterFloatBoxes}{}
120 \newcommand*{\TopFloatBoxes}{}
121 \newcommand*{\BottomFloatBoxes}{}
122 \newcommand*{\PlainFloatBoxes}{}
123
124 \newcommand{\capsubrowsettings}{}
125
126 \NewDocumentCommand{\RawFloats}{o o}{}

```

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

To be used inside a minipage or parbox.

```

127 \newcommand{\RawCaption}[1]{#1}

```

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

Places additional text inside a float, inside a CSS `<div>` of class `floatfoot`.

```

128 \NewDocumentCommand{\floatfoot}{s +m}{%
129 \begin{BlockClass}{floatfoot}
130 #2
131 \end{BlockClass}
132 }

```

Used to compute `\linewidth`.

```

133 \newbool{\LWR@insubfloatrow}
134 \boolfalse{\LWR@insubfloatrow}

```

Env    `subfloatrow`     $[\langle num\_floats \rangle]$

```

135 \newenvironment*{subfloatrow}[1][2]
136 {

```

The row of floats is placed into a <div> of class floatrow:

```
137 \LWR@forcenewpage
138 \BlockClass{floatrow}
```

While inside the floatrow, LWR@insubfloatrow is set true, which tells \floatbox to use \subfigure or \subtable.

```
139 \begingroup
140 \booltrue{LWR@insubfloatrow}
141 }
142 {
143 \endgroup
144 \endBlockClass
145 \boolfalse{LWR@insubfloatrow}
146 }
```

---

File 57 **lwarp-flushend.sty**

§ 136    Package **flushend**

Pkg    flushend    Emulated.

for HTML output:    Discard all options for lwarp-flushend:

```
1 \LWR@ProvidesPackageDrop{flushend}
2 %    \end{macrocode}
3 %
4 %    \begin{macrocode}
5 \newcommand*{\flushend}{}
6 \newcommand*{\raggedend}{}
7 \newcommand*{\flushcolsend}{}
8 \newcommand*{\raggedcolsend}{}
9 \newcommand*{\atColsBreak}[1]{}
10 \newcommand*{\atColsEnd}[1]{}
11 \newcommand*{\showcolsendrule}{}

```

---

File 58 **lwarp-fncychap.sty**

§ 137    Package **fncychap**

Pkg    fncychap    fncychap is ignored.

for HTML output:    Discard all options for lwarp-fncychap:

```
1 \LWR@ProvidesPackageDrop{fncychap}

```

---

```

2 \def\mghrulefill#1{}
3 \def\ChNameLowerCase{}
4 \def\ChNameUpperCase{}
5 \def\ChNameAsIs{}
6 \def\ChTitleLowerCase{}
7 \def\ChTitleUpperCase{}
8 \def\ChTitleAsIs{}
9 \newcommand{\ChRuleWidth}[1]{}
10 \newcommand{\ChNameVar}[1]{}
11 \newcommand{\ChNumVar}[1]{}
12 \newcommand{\ChTitleVar}[1]{}
13 \newcommand{\TheAlphaChapter}{}
14 \newcommand{\DOCH}{}
15 \newcommand{\DOTI}[1]{}
16 \newcommand{\DOTIS}[1]{}
17 \newlength{\mylen}
18 \newlength{\myhi}
19 \newlength{\px}
20 \newlength{\py}
21 \newlength{\pyy}
22 \newlength{\pxx}
23 \newlength{\RW}
24 \newcommand{\FmN}[1]{#1}
25 \newcommand{\FmTi}[1]{#1}

```

---

File 59 **lwarp-fnpos.sty**

§ 138     Package **fnpos**

Pkg   fnpos   fnpos is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{fnpos}

2 \newcommand*{\makeFNbottom}{}
3 \newcommand*{\makeFNmid}{}
4 \newcommand*{\makeFNbelow}{}
5 \newcommand*{\makeFNabove}{}

```

---

File 60 **lwarp-fontenc.sty**

§ 139     Package **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}
```

---

File 61 **lwarp-fontspec.sty**

§ 140     Package **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}
```

---

File 62 **lwarp-footmisc.sty**

§ 141     Package **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 }

```

---

File 63 **lwarp-footnote.sty**

## § 142 Package **footnote**

Pkg footnote footnote is used with minor patches.

for HTML output: 1 \LWR@ProvidesPackagePass{footnote}

Removed print-version formatting:

```

2 \def\fn@startnote{%
3   \parboxrestore%
4   \protected@edef\@currentlabel{\csname p@\@mpfn\endcsname\@thefnmark}%
5   \color@begingroup% *** conflicts with lwarp
6 }
7
8 \let\fn@endnote\color@endgroup% *** conflicts with lwarp
9 \newcommand*\fn@endnote}{%
10 \LWR@htmltagc{/LWR@tagregularparagraph}%
11 \LWR@orignewline%
12 }

```

Removed print-version formatting:

```

13 \def\fn@startfntext{%
14   \setbox\z@\vbox\bgroup%
15     \fn@startnote%
16     \fn@prefntext%
17     \ignorespaces%
18 }

```

Removed print-version formatting, added closing paragraph tag:

```

19 \def\fn@endfntext{%
20   \LWR@htmltagc{/\LWR@tagregularparagraph}%
21   \LWR@orignewline%
22   \fn@postfntext%
23   \egroup%
24   \beginngroup%
25     \let\@makefntext\@empty%
26     \let\@finalstrut\@gobble%
27     \LetLtxMacro\rule\@gobbletwo% *8* also the optional argument?
28     \@footnotetext{\unvbox\z@}%
29   \endgroup%
30 }

```

These have been redefined, so re-\let them again:

```

31 \let\endfootnote\fn@endfntext
32 \let\endfootnotetext\endfootnote

```

---

File 64 **lwarp-footnotehyper.sty**

§ 143      Package **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}

```



---

File 65 **lwarp-framed.sty**

§ 144     Package **framed**

*(Based on original code by DONALD ARSENEAU.)*

Pkg    framed    framed is supported and patched by lwarp.

for HTML output:    Accept all options for lwarp-framed:

```

1 \LWR@ProvidesPackagePass{framed}
2 \RequirePackage{xcolor}% for \convertcolorspec

3
4 \renewenvironment{framed}{%
5 \LWR@forcenewpage
6 \BlockClass{framed}%
7 }
8 {\endBlockClass}
9
10 \renewenvironment{oframed}{%
11 \LWR@forcenewpage
12 \BlockClass{framed}%
13 }
14 {\endBlockClass}
15
16
17 \renewenvironment{shaded}{%
18 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
19 \LWR@forcenewpage
20 \BlockClass[background: \#\LWR@tempcolor]{framed}%
21 }
22 {\endBlockClass}
23
24 \renewenvironment{shaded*}{%
25 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
26 \LWR@forcenewpage
27 \BlockClass[background: \#\LWR@tempcolor]{framed}%
28 }
29 {\endBlockClass}
30
31
32 \renewenvironment{leftbar}{%
33 \LWR@forcenewpage
34     \BlockClass{framedleftbar}
35     \def\FrameCommand{}}%
```

```

36 \MakeFramed {}
37 }%
38 {\endMakeFramed\endBlockClass}
39
40
41 \renewenvironment{snugshade}{%
42 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
43 \LWR@forcenewpage
44 \BlockClass[background: \#\LWR@tempcolor]{snugframed}%
45 }
46 {\endBlockClass}
47
48 \renewenvironment{snugshade*}{%
49 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
50 \LWR@forcenewpage
51 \BlockClass[background: \#\LWR@tempcolor]{snugframed}%
52 }
53 {\endBlockClass}
54
55 \let\oframed\framed
56 \let\endoframed\endframed
57
58
59 \RenewEnviron{titled-frame}[1]{%
60 \CustomFBox{#1}{0pt}{0pt}{0pt}{0pt}{\BODY}
61 }

\CustomFBox {<toptitle>} {<bottitle>} {<thicknesstop>} {<bottom>} {<left>} {<right>}
{<text contents>}

62 \renewcommand{\CustomFBox}[7]{%
63 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
64 \LWR@forcenewpage
65 \begin{BlockClass}[border: 3px solid \#\LWR@tempcolor]{framed}%
66 \ifthenelse{\isempty{#1}}{}{% not empty
67   \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
68     \textcolor{TFTitleColor}{\textbf{#1}}%
69   \end{BlockClass}
70 }% not empty
71
72 #7
73
74 \ifthenelse{\isempty{#2}}{}{% not empty
75   \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
76   \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
77     \textcolor{TFTitleColor}{\textbf{#2}}%
78   \end{BlockClass}
79 }% not empty
80 \end{BlockClass}

```

---

```

81 }

\TitleBarFrame [\<marker>] {\<title>} {\<contents>}

82 \renewcommand\TitleBarFrame[3] [] {
83 \CustomFBox
84   {\#2}{\}%
85   \fboxrule\fboxrule\fboxrule\fboxrule
86   {\#3}\%
87 }

88 \renewcommand{\TF@Title}[1]{\#1}

MakeFramed {\<settings>}

89 \let\MakeFramed\relax
90 \let\endMakeFramed\relax
91
92 \NewEnviron{MakeFramed}[1]{%
93 \FrameCommand{\begin{minipage}{\linewidth}\BODY\end{minipage}}%
94 }

\fb@put@frame {\<frame cmd no split>} {\<frame cmd split>}

95 \renewcommand*\fb@put@frame[2]{%
96 \relax%
97 \@tempboxa%
98 }

```

---

File 66 **lwarp-ftnright.sty**

\$ 145    Package **ftnright**

Pkg    ftnright    ftnright is ignored.

**for HTML output:**    Discard all options for lwarp-ftnright:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

---

File 67 **lwarp-fullpage.sty**

\$ 146    Package **fullpage**

Pkg    fullpage    Emulated.

**for HTML output:** Discard all options for lwarp-fullpage:

```
1 \LWR@ProvidesPackageDrop{fullpage}
```

---

File 68 **lwarp-fullwidth.sty**

§ 147     Package **fullwidth**

Pkg    fullwidth    fullwidth is emulated.

A minipage is used, of no HTML width.

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{fullwidth}

2 \newenvironment*{fullwidth}[1][{}]{%
3 \minipagefullwidth%
4 \minipage{\linewidth}%
5 }
6 {%
7 \endminipage%
8 }
```

---

File 69 **lwarp-geometry.sty**

§ 148     Package **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]{%}
```

---

File 70 **lwarp-glossaries.sty**

§ 149      Package **glossaries**

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

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

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

[placement and toc options](#)    The glossaries may be placed in a numbered or unnumbered section, given a toc entry, and placed inline or on their own HTML page:

**Numbered section, on its own HTML page:**

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

**Unnumbered section, inline with the current HTML page:**

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

**Unnumbered section, on its own HTML page:**

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

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

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

Opt lwarpmk    **htmlglossary**

**for HTML output:**

```
1 \PassOptionsToPackage{xindy}{glossaries}
2 \LWR@ProvidesPackagePass{glossaries}
3 \setupglossaries{nonumberlist}
4 \setglossarystyle{index}
```

Patched to fix TOC pointing to the previous page:

```

5 \renewcommand*{\@p@glossarysection}[2]{%
6   \glsclearpage
7   \phantomsection
8   \ifdefempty\@@glossarysecstar
9   {%
10    \csname\@@glossarysec\endcsname{#2}%
11  }%
12  {%

```

In the original, the TOC entry was made before the section, thus linking to the phantomsection in the printed version, but for HTML this caused the link to point to the page before the glossaries. Here, the TOC entry is made after the section is created:

```

13    \csname\@@glossarysec\endcsname*{#2}%
14    \@gls@toc{#1}{\@@glossarysec}% Moved after the previous line.
15  }%
16  \@@glossaryseclabel
17 }

```

---

## File 71 lwarp-graphics.sty

### § 150 Package **graphics**

Pkg graphics graphics is emulated.

for HTML output: 1 \LWR@ProvidesPackagePass{graphics}

#### § 150.1 Graphics extensions

\DeclareGraphicsExtensions {\langle list \rangle}

\AtBeginDocument allow SVG files instead of PDF:

```

2 \AtBeginDocument{
3 \DeclareGraphicsExtensions{.svg,.SVG,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}
4 \DeclareGraphicsRule{.svg}{svg}{.svg}{}
5 \DeclareGraphicsRule{.SVG}{svg}{.SVG}{}
6 }

```

Inside a lateximage, allow PDF instead of SVG:

```

7 \appto\LWR@restoreorigformatting{%
8 \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
9 }

```

## § 150.2 Length conversions and graphics options



whitespace

A scaled image in  $\TeX$  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.

```
10 \AtBeginDocument{
11 \renewcommand*\unitspace{}}
12 }
```

Used to store the user's selected dimensions and HTML class.

The class defaults to “inlineimage” unless changed by a class=xyx option.

```
13 \newlength{\LWR@igwidth}
14 \newlength{\LWR@igheight}
15 \newcommand*\LWR@igwidthstyle{}
16 \newcommand*\LWR@igheightstyle{}
17 \newcommand*\LWR@igorigin{}
18 \newcommand*\LWR@igangle{}
19 \newcommand*\LWR@igxscale{1}
20 \newcommand*\LWR@igyscale{1}
21 \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:

```
22 \define@key{igraph}{width}{%
23 \setlength{\LWR@igwidth}{#1}%
24 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}{%
25 {%
```

Default to use the converted fixed length given:

```
26 \uselengthunit{PT}%
27 \renewcommand*\LWR@igwidthstyle{width:\rndprintlength{\LWR@igwidth}}%
```

If ex or em dimensions were given, use those instead:

```
28 \IfEndWith{#1}{ex}%
29 {\renewcommand*\LWR@igwidthstyle{width:#1}}% yes ex
30 {}% not ex
31 \IfEndWith{#1}{em}%
32 {\renewcommand*\LWR@igwidthstyle{width:#1}}% yes em
33 {}% not em
34 \IfEndWith{#1}{\}%
35 {\renewcommand*\LWR@igwidthstyle{width:#1}}% yes percent
```

```

36    {}% not percent
37    \IfEndWith{#1}{px}%
38    {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
39    {}% not px
40 }{}% end of length > Opt
41 }

```

If an optional height was given, set an HTML style:

```

42 \define@key{igraph}{height}{%
43 \setlength{\LWR@igheight}{#1}%
44 \ifthenelse{\lengthtest{\LWR@igheight > Opt}}%
45 {%

```

Default to use the converted fixed length given:

```

46    \uselengthunit{PT}%
47    \renewcommand*{\LWR@igheightstyle}{%
48    height:\rndprintlength{\LWR@igheight} %
49    }%

```

If ex or em dimensions were given, use those instead:

```

50    \IfEndWith{#1}{ex}%
51    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
52    {}% not ex
53    \IfEndWith{#1}{em}%
54    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
55    {}% not em
56    \IfEndWith{#1}{\}%
57    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
58    {}% not percent
59    \IfEndWith{#1}{px}%
60    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
61    {}% not px
62 }{}% end of length > Opt
63 }

```

Handle origin key:

```

64 \define@key{igraph}{origin}{%
65 \renewcommand*{\LWR@igorigin}{#1}%
66 }

```

Handle angle key:

```

67 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}

```



Handle class key:

```
68 \define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}
69
```

It appears that graphicx does not have separate keys for xscale and yscale. scale adjusts both at the same time.

```
70 \define@key{igraph}{scale}{%
71 \renewcommand*{\LWR@igxscale}{#1}%
72 \renewcommand*{\LWR@igyyscale}{#1}}
```

Numerous ignored keys:

```
73 \define@key{igraph}{bb}{}
74 \define@key{igraph}{bblx}{}
75 \define@key{igraph}{bbly}{}
76 \define@key{igraph}{bburx}{}
77 \define@key{igraph}{bbury}{}
78 \define@key{igraph}{natwidth}{}
79 \define@key{igraph}{natheight}{}
80 \define@key{igraph}{hiresbb}{}
81 \define@key{igraph}{viewport}{}
82 \define@key{igraph}{trim}{}
83 \define@key{igraph}{totalheight}{}
84 \define@key{igraph}{keepaspectratio}{}
85 \define@key{igraph}{clip}{}
86 \define@key{igraph}{draft}{}
87 \define@key{igraph}{type}{}
88 \define@key{igraph}{ext}{}
89 \define@key{igraph}{read}{}
90 \define@key{igraph}{command}{}

```

### § 150.3 Printing HTML styles

`\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.

```
91 \newcommand*{\LWR@rotstyle}[2]{%
92 #1transform:rotate(-#2deg);
93 }
```

`\LWR@scaletyle`  $\{\langle prefix \rangle\} \{\langle xscale \rangle\} \{\langle yscale \rangle\}$

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.

```
94 \newcommand*\LWR@scalestyle}[3]{%
95   #1transform:scale(#2,#3);
96 }
```

## § 150.4 `\includegraphics`

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```
97 \newbool{LWR@infloatrow}
98 \boolfalse{LWR@infloatrow}
```

`\LWR@opacity` may be set by the transparent package. For HTML it is only used for `\includegraphics`.

```
99 \def\LWR@opacity{1}
```

Used to determine the actual image size if needed:

```
100 \newsavebox{\LWR@imagesizebox}

101 \let\LWR@origGin@setfile\Gin@setfile
```

Define the new class key for the print-mode version of `\includegraphics`, which is enabled inside a `lateximage`.

```
102 \AtBeginDocument{
103   \define@key{Gin}{class}{}
104 }
```

`\LWR@includegraphicsb` \* [*<2: options>*] [*<3: options>*] [*<4: filename>*]

graphics syntax is `\includegraphics * [<llx, lly>] [<urx, ury>] [<file>]`

graphicx syntax is `\includegraphics [<key values>] [<file>]`

If #3 is empty, only one optional argument was given, thus `graphicx` syntax.

```
105 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
106 {%
107   \LWR@traceinfo{\LWR@includegraphicsb #4}%
```

Start the image tag on a new line, allow PDF output word wrap:

```
108 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```

109 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}%
110 {%
111   \ifbool{LWR@infloatrow}%
112   {}
113   {% not in a minipage or a floatrow:
114     \setlength{\linewidth}{6in}%
115     \setlength{\textwidth}{6in}%
116     \setlength{\textheight}{9in}%
117   }%
118 }{}%

119 \begingroup%
120 \renewcommand*{\Gin@setfile}[3]{%
121 \LWR@traceinfo{Gin@setfile ##3}%
122 \xdef\LWR@parsedfilename{##3}%
123 }%
124 \Gininclude@graphics{\detokenize\expandafter{#4}}%
125 \endgroup%
126 \filename@parse{\LWR@parsedfilename}%

```

For correct em sizing during the width and height conversions:

```

127 \large%

```

Reset some defaults, possibly will be changed below if options were given:

```

128 \setlength{\LWR@igwidth}{0pt}%
129 \setlength{\LWR@igheight}{0pt}%
130 \renewcommand*{\LWR@igwidthstyle}{}%
131 \renewcommand*{\LWR@igheightstyle}{}%
132 \renewcommand*{\LWR@igorigin}{}%
133 \renewcommand*{\LWR@igangle}{}%
134 \renewcommand*{\LWR@igxscale}{1}%
135 \renewcommand*{\LWR@igyscale}{1}%
136 \renewcommand*{\LWR@igclass}{inlineimage}%

```

If #3 is empty, only one optional argument was given, thus `graphicx` syntax:

```

137 \IfValueF{#3}{%
138 \IfValueTF{#2}%
139 {\setkeys{igraph}{#2}}%
140 {\setkeys{igraph}}}%
141 }%

```

If formatting for a word processor, find and set the actual image size, without rotation, using PDF instead of SVG to find the original bounding box:

```

142 \ifbool{FormatWP}{%
143   \begingroup%
144   \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
145   \define@key{Gin}{angle}{}%
146   \IfBooleanTF{#1}%
147   {% starred
148     \IfValueTF{#3}%
149     {%
150       \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics* [#2] [#3] {#4}}%
151     }%
152     {%
153       \IfValueTF{#2}%
154       {%
155         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics* [#2] {#4}}%
156       }{%
157         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics* {#4}}%
158       }%
159     }%
160   }% starred
161   {% not starred
162     \IfValueTF{#3}%
163     {%
164       \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics [#2] [#3] {#4}}%
165     }%
166     {%
167       \IfValueTF{#2}%
168       {%
169         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics [#2] {#4}}%
170       }{%
171         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics {#4}}%
172       }%
173     }%
174   }% not starred
175   \endgroup%
176   \uselengthunit{PT}%
177   \settowidth{\LWR@igwidth}{\usebox{\LWR@imagesizebox}}%
178   \global\renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%
179   \settoheight{\LWR@igheight}{\usebox{\LWR@imagesizebox}}%
180   \global\renewcommand*{\LWR@igheightstyle}{height:\rndprintlength{\LWR@igheight}}%
181 }{}%

```

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.

```

182 \LWR@traceinfo{\LWR@includegraphicsb: about to create href}%
183 \href{\LWR@parsedfilename}%
184 {% start of href
185 \LWR@traceinfo{\LWR@includegraphicsb: about to LWR@htmltag}%
186 \LWR@htmltag{% start of image tags
187 img src="\LWR@parsedfilename" \LWR@orignewline%
188 \LWR@origtilde{} alt="[filename@base]" \LWR@orignewline%

```

Only include a style tag if a width, height, angle, or scale was given:

```

189 \ifthenelse{
190   \NOT\equal{\LWR@igwidthstyle}{}} \OR
191   \NOT\equal{\LWR@igheightstyle}{}} \OR
192   \NOT\equal{\LWR@igorigin{}} \OR
193   \NOT\equal{\LWR@igangle{}} \OR
194   \NOT\equal{\LWR@igxscale}{1}} \OR
195   \NOT\equal{\LWR@igyyscale}{1}}
196 }%
197 {\LWR@origtilde{} style="%
198 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{}}%
199 {\LWR@igwidthstyle;}}}%
200 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{}}%
201 {\LWR@igheightstyle;}}}%
202 \ifthenelse{\NOT\equal{\LWR@igorigin{}}}%
203 {\LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{}}%
204 \ifthenelse{\NOT\equal{\LWR@igangle{}}}%
205 {%
206 \LWR@rotstyle{-ms-}{\LWR@igangle}%
207 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
208 \LWR@rotstyle{}{\LWR@igangle%
209 }}}%
210 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}}\OR%
211 \NOT\equal{\LWR@igyyscale}{1}}}%
212 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyyscale}%
213 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyyscale}%
214 \LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyyscale}}}%
215 %
216 \ifthenelse{\NOT\equal{\LWR@opacity}{1}}%
217 {opacity:\LWR@opacity;}%
218 {}%
219 %
220 " \LWR@orignewline}{}}%

```

Set the class:

```

221 \LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%

```

```

222 }% end of image tags
223 }% end of href
224 \endgroup

```

Return to small-sized output:

```

225 \LWR@origscriptsize
226 \LWR@traceinfo{LWR@includegraphicsb done}%
227 }

```

`\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.

```

228 \AtBeginDocument{
229
230 \LWR@traceinfo{Patching includegraphics.}
231
232 \LetLtxMacro\LWR@originincludegraphics\includegraphics
233
234 \renewcommand*{\includegraphics}
235 {%

```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```

236 \LWR@traceinfo{includegraphics}
237 \LWR@ensuredoingapar%
238 \begingroup%
239 \catcode'\_ =12%
240 \LWR@includegraphicsb%
241 }% includegraphics
242 }% AtBeginDocument

```

## § 150.5 Boxes

`\LWR@rotboxorigin` Holds the origin key letters.

```

243 \newcommand*{\LWR@rotboxorigin}{

```

`\LWR@originname` {*<letter>*}

Given one  $\TeX$  origin key value, translate into an HTML origin word:

```

244 \newcommand*{\LWR@originname}[1]{%
245 \ifthenelse{\equal{#1}{t}}{top}{}%
246 \ifthenelse{\equal{#1}{b}}{bottom}{}%
247 \ifthenelse{\equal{#1}{c}}{center}{}%
248 \ifthenelse{\equal{#1}{l}}{left}{}%
249 \ifthenelse{\equal{#1}{r}}{right}{}%
250 }

```

`\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:

```

251 \newcommand*{\LWR@originnames}[1]{%
252 \StrChar{#1}{1}[\LWR@strresult]%
253 \LWR@originname{\LWR@strresult}%
254 \StrChar{#1}{2}[\LWR@strresult]%
255 \LWR@originname{\LWR@strresult}%
256 }

```

Handle the origin key for `\rotatebox`:

```

257 \define@key{krotbox}{origin}{%
258 \renewcommand*{\LWR@rotboxorigin}{#1}%
259 }

```

These keys are ignored:

```

260 \define@key{krotbox}{x}{}
261 \define@key{krotbox}{y}{}
262 \define@key{krotbox}{units}{}

```

`\rotatebox`  $[\langle keyval list \rangle] \{\langle angle \rangle\} \{\langle text \rangle\}$

```

263 \LetLtxMacro\LWR@origrotatebox\rotatebox
264
265 \AtBeginDocument{
266 \RenewDocumentCommand{\rotatebox}{O{} m +m}{%

```

Reset the origin to “none-given”:

```

267 \renewcommand*{\LWR@rotboxorigin}{}

```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```

268 \setkeys{krotbox}{#1}%

```

Select inline-block so that HTML will transform this span:

```
269 \LWR@htmltagc{span style="display: inline-block; %
```

If an origin was given, translate and print the origin information:

```
270 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}%
271 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%
```

Print the rotation information:

```
272 \LWR@rotstyle{-ms-}{#2} %
273 \LWR@rotstyle{-webkit-}{#2} %
274 \LWR@rotstyle{}{#2} %
275 "{}}\LWR@orignewline%
```

Print the text to be rotated:

```
276 \begin{LWR@nestspan}%
277 #3%
```

Close the span:

```
278 \LWR@htmltagc{/span}%
279 \end{LWR@nestspan}%
280 }
281 }% AtBeginDocument
```

`\scalebox`     $\{ \langle h\text{-scale} \rangle \} [ \langle v\text{-scale} \rangle ] \{ \langle \text{text} \rangle \}$

```
282 \LetLtxMacro\LWR@origscalebox\scalebox
283
284 \AtBeginDocument{
285 \RenewDocumentCommand{\scalebox}{m o m}{%
```

Select inline-block so that HTML will transform this span:

```
286 \LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
287 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
288 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
289 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
290 "{}}%
```

Print the text to be scaled:

```
291 \begin{LWR@nestspan}%
292 #3%
```



Close the span:

```
293 \LWR@htmltagc{/span}%
294 \end{LWR@nestspan}%
295 }
296 }% AtBeginDocument
```

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

```
297 \let\LWR@origreflectbox\reflectbox
298
299 \AtBeginDocument{
300 \renewcommand{\reflectbox}[1]{\scalebox{-1}[1]{#1}}
301 }
```

`\resizebox`  $\{\langle h-length \rangle\} \{\langle v-length \rangle\} \{\langle text \rangle\}$

Simply prints its text argument.

```
302 \LetLtxMacro\LWR@origresizebox\resizebox
303
304 \AtBeginDocument{
305 \renewcommand{\resizebox}[3]{#3}
306 }
```

---

## File 72 `lwarp-graphicx.sty`

\$ 151    Package    **graphicx**

Pkg    `graphicx`    `graphicx` is emulated.

`graphicx` loads `graphics`, which also loads `lwarp-graphics`, which remembers the original `graphics` definitions for use inside a `lateximage`, and then patches them `\AtBeginDocument` for HTML output.

`lwarp-graphics` handles the syntax of either `graphics` or `graphicx`.

for HTML output:    `1 \LWR@ProvidesPackagePass{graphicx}`

---

## File 73 `lwarp-grffile.sty`

\$ 152    Package    **grffile**

Pkg    `grffile`    `grffile` is supported as-is. File types known to the browser are displayed, and unknown

⚠ **matching PDF and SVG** file types are given a link. Each PDF image for print mode should be accompanied by an SVG, PNG, or JPG version for HTML.

lwarp-grffile now exists as a placeholder since grffile used to be emulated by lwarp, and thus older versions of lwarp-grffile may exist and should be overwritten by this newer version.

**for HTML output:** 1 \LWR@ProvidesPackagePass{grffile}

---

File 74 **lwarp-hyperref.sty**

## § 153 Package **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]{}

```

\hyperimage {<url>} {<alt text>}

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\DeclareRobustCommand*\hyperimage{%
17\LWR@ensuredoingapar%
18\begingroup\catcode'\_ =12
19\LWR@hyperimageb%
20}

```

\hyperdef {<1: category>} {<2: name>} {<3: text>}

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 }

```

`\LWR@hyperrefb`     $\{\langle 1: URL \rangle\} \{\langle 2: category \rangle\} \{\langle 3: name \rangle\} \{\langle 4: text \rangle\}$   
 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 }

```

`\LWR@hyperrefc`     $[\langle label \rangle] \{\langle text \rangle\}$   
 Creates text as an HTML link to the  $\LaTeX$  label.

```

32 \NewDocumentCommand{\LWR@hyperrefc}{O{label} +m}{%
33 \LWR@startref{#1}%
34 #2%
35 \LWR@htmltag{/a}%
36 \endgroup%
37 }

```

`\hyperref`     $\{\langle 1: URL \rangle\} \{\langle 2: category \rangle\} \{\langle 3: name \rangle\} \{\langle 4: text \rangle\}$     — or —  
 $[\langle 1: label \rangle] \{\langle 2: text \rangle\}$

```

38 \DeclareRobustCommand*\hyperref{%
39 \LWR@ensuredoingapar%
40 \begingroup\catcode'\_ =12
41 \@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
42 }

```

`\hypertarget`     $\{\langle name \rangle\} \{\langle text \rangle\}$   
 Creates an anchor to name with the given text.

```

43 \NewDocumentCommand{\hypertarget}{m +m}{%
44 \label{#1}%
45 #2%
46 }

```

`\hyperlink`     $\{\langle name \rangle\} \{\langle text \rangle\}$   
 Creates a link to the anchor created by `hypertarget`, with the given link text.

```

47 \NewDocumentCommand{\hyperlink}{m +m}{%

```

```
48 \hyperref[#1]{#2}%
49 }
```

`\autoref` \*  $\langle label \rangle$

For HTML, `\cleveref` is used instead.

```
50 \NewDocumentCommand{\autoref}{s m}{%
51 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
52 }
```

`\autopageref`  $\langle label \rangle$

For HTML, `\cleveref` is used instead.

```
53 \NewDocumentCommand{\autopageref}{s m}{%
54 \IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
55 }
```

`\pdfstringdef`  $\langle macroname \rangle$   $\langle T_{\text{E}}\text{Xstring} \rangle$

```
56 \newcommand{\pdfstringdef}[2]{}
```

`\pdfbookmark` [ $\langle level \rangle$ ]  $\langle text \rangle$   $\langle name \rangle$

```
57 \newcommand{\pdfbookmark}[3][{}]{}
```

`\currentpdfbookmark`  $\langle text \rangle$   $\langle name \rangle$

```
58 \newcommand{\currentpdfbookmark}[2]{}
```

`\subpdfbookmark`  $\langle text \rangle$   $\langle name \rangle$

```
59 \newcommand{\subpdfbookmark}[2]{}
```

`\belowpdfbookmark`  $\langle text \rangle$   $\langle name \rangle$

```
60 \newcommand{\belowpdfbookmark}[2]{}
```

`\texorpdfstring`  $\langle T_{\text{E}}\text{Xstring} \rangle$   $\langle PDFstring \rangle$

```
61 \newcommand{\texorpdfstring}[2]{#2}
```

`\hypercalcbp`  $\langle dimen \rangle$  From `hyperref`.

```
62 \def\hypercalcbp#1{%
63 \strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax
64 }%
```

`\Acrobatmenu`     $\{\langle menuoption \rangle\} \{\langle text \rangle\}$

65 `\newcommand{\Acrobatmenu}[2]{}{}`

`\TextField`     $[\langle parameters \rangle] \{\langle label \rangle\}$

66 `\DeclareRobustCommand{\TextField}[2] [] {}{}`

`\CheckBox`     $[\langle parameters \rangle] \{\langle label \rangle\}$

67 `\DeclareRobustCommand{\CheckBox}[2] [] {}{}`

`\ChoiceMenu`     $[\langle parameters \rangle] \{\langle label \rangle\} \{\langle choices \rangle\}$

68 `\DeclareRobustCommand{\ChoiceMenu}[3] [] {}{}`

`\PushButton`     $[\langle parameters \rangle] \{\langle label \rangle\}$

69 `\DeclareRobustCommand{\PushButton}[2] [] {}{}`

`\Submit`     $[\langle parameters \rangle] \{\langle label \rangle\}$

70 `\DeclareRobustCommand{\Submit}[2] [] {}{}`

`\Reset`     $[\langle parameters \rangle] \{\langle label \rangle\}$

71 `\DeclareRobustCommand{\Reset}[2] [] {}{}`

`\Gauge`     $[\langle parameters \rangle] \{\langle label \rangle\}$

72 `\DeclareRobustCommand{\Gauge}[2] [] {}{}`

`\LayoutTextField`     $\{\langle label \rangle\} \{\langle field \rangle\}$

73 `\newcommand*{\LayoutTextField}[2]{}{}`

`\LayoutChoiceField`     $\{\langle label \rangle\} \{\langle field \rangle\}$

74 `\newcommand*{\LayoutChoiceField}[2]{}{}`

`\LayoutCheckField`     $\{\langle label \rangle\} \{\langle field \rangle\}$

75 `\newcommand*{\LayoutCheckField}[2]{}{}`

```

\MakeRadioField    {\langle width\rangle} {\langle height\rangle}
                    76 \newcommand*{\MakeRadioField}[2]{\{}

\MakeCheckField    {\langle width\rangle} {\langle height\rangle}
                    77 \newcommand*{\MakeCheckField}[2]{\{}

\MakeTextField     {\langle width\rangle} {\langle height\rangle}
                    78 \newcommand*{\MakeTextField}[2]{\{}

\MakeChoiceField   {\langle width\rangle} {\langle height\rangle}
                    79 \newcommand*{\MakeChoiceField}[2]{\{}

\MakeFieldButton   {\langle text\rangle}
                    80 \newcommand{\MakeFieldButton}[1]{\{}

```

---

File 75 **lwarp-hyperxmp.sty**

\$ 154    Package **hyperxmp**

    Pkg    hyperxmp    Emulated.

**for HTML output:**    Discard all options for lwarp-hyperxmp:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```

---

File 76 **lwarp-idxlayout.sty**

\$ 155    Package **idxlayout**

    Pkg    idxlayout    Emulated.

**for HTML output:**    Discard all options for lwarp-idxlayout:

```
1 \LWR@ProvidesPackageDrop{idxlayout}
```

```
2 \newcommand{\LWR@indexprenote}{}
3
```

---

```

4 \renewcommand*{\printindex}
5 {
6 \LWR@startpars
7
8 \LWR@indexprenote
9
10 \LWR@origprintindex
11 }
12
13 \newcommand{\setindexprenote}[1]{\renewcommand{\LWR@indexprenote}{#1}}
14 \newcommand*{\noindexprenote}{\renewcommand{\LWR@indexprenote}{}}
15
16 \newcommand{\idxlayout}[1]{ }
17 \newcommand*{\indexfont}{}
18 \newcommand*{\indexjustific}{}
19 \newcommand*{\indexsubsdelim}{}
20 \newcommand*{\indexstheadcase}{}

```

---

File 77 **lwarp-indentfirst.sty**

§ 156     Package **indentfirst**

Pkg   indentfirst   indentfirst is ignored.

Discard all options for lwarp-indentfirst:

**for HTML output:**    1 \LWR@ProvidesPackageDrop{indentfirst}

---

File 78 **lwarp-inputenc.sty**

§ 157     Package **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}

File 79 **lwarp-keyfloat.sty**§ 158 Package **keyfloat**

Pkg keyfloat keyfloat is supported with minor adjustments.

 **keywrap** If placing a `\keyfig[H]` inside a `keywrap`, use an absolute width for `\keyfig`, instead of `lw`-proportional widths. (The [H] option forces the use of a minipage, which internally adjusts for a virtual 6-inch wide minipage, which then corrupts the `lw` option.)

for HTML output: `1 \LWR@ProvidesPackagePass{keyfloat}`

After keyfloat has loaded:

```

2 \AtBeginDocument{

3 \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{KFLT@boxinner: done}%
13 }

14 \DeclareDocumentEnvironment{KFLT@marginfloat}{0}{-1.2ex} m}
15 {%
16 \LWR@BlockClassWP{float:right; width:2in; margin:10pt}{-}{marginblock}%
17 \captionsetup{type=#2}%
18 }
19 {%
20 \endLWR@BlockClassWP%
21 }

22 \DeclareDocumentEnvironment{marginfigure}{o}
23 {\begin{KFLT@marginfloat}{figure}}
24 {\end{KFLT@marginfloat}}
25
26 \DeclareDocumentEnvironment{margintable}{o}
27 {\begin{KFLT@marginfloat}{table}}
28 {\end{KFLT@marginfloat}}
```



---

```

29 \DeclareDocumentEnvironment{keywrap}{m +m}
30 {%
31 \LWR@ensuredoingapar%
32 \setlength{\LWR@templelengthone}{#1}%
33 \uselengthunit{PT}%
34 \begin{LWR@BlockClassWP}{%
35     float:right; width:\rndprintlength{\LWR@templelengthone}; %
36     margin:10pt%
37 }%
38 {%
39     width:\rndprintlength{\LWR@templelengthone}%
40 }%
41 {marginblock}%
42 \setlength{\linewidth}{.95\LWR@templelengthone}%
43 #2%
44 \end{LWR@BlockClassWP}%
45 }
46 {%
47 }

48 }% AtBeginDocument

```

---

File 80 **lwarp-layout.sty**

\$ 159     Package **layout**

Pkg   layout   layout is ignored.

for HTML output: Discard all options for lwarp-layout:

```

1 \LWR@ProvidesPackageDrop{layout}

2 \NewDocumentCommand{\layout}{s}{-}

```

---

File 81 **lwarp-letterspace.sty**

\$ 160     Package **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}

```

---

File 82 **lwarp-lettrine.sty**

§ 161 Package **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 \newcommand*{\LettrineFontHook}{}
27 \newcommand*{\LettrineFont}[1]{\InlineClass{lettrine}{#1}}
28 \newcommand*{\LettrineFontEPS}[1]{\includegraphics[height=1.5ex]{#1}}

```

---

File 83 **lwarp-lips.sty**

§ 162      Package **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 {}

```

---

File 84 **lwarp-listings.sty**

§ 163      Package **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}}

```

`\lst@Init`    `{\backslash-processing}`    Done at the start of a listing.

```

9 \renewcommand{\lst@Init}[1]{%

```

First, perform the listings initialization:

```

10 \LWR@traceinfo{lst@Init}%
11 \renewcommand*{\@capttype}{lstlisting}%
12 \LWR@origlst@Init{#1}%
13 \LWR@traceinfo{finished origlst@Init}%
14 \lst@ifdisplaystyle%

```

Creating a display.

Disable line numbers, produce the <pre>, then reenable line numbers.

```

15 \LWR@traceinfo{About to create verbatim.}%
16 \let\lsthk@EveryPar\relax%
17 \LWR@forcenewpage
18 \LWR@atbeginverbatim{programlisting}%
19
20 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
21 \else%

```

Inline, so open a <span>:

```

22 \ifbool{LWR@verbtags}{\LWR@htmltag{span class="inlineprogramlisting"}}{}%
23 \fi%
24 }

```

`\lst@DeInit`    Done at the end of a listing.

```

25 \renewcommand*{\lst@DeInit}{%
26 \lst@ifdisplaystyle%

```

Creating a display.

Disable line numbers, produce the </pre>, then reenable line numbers:

```

27 \let\lsthk@EveryPar\relax%
28
29 \LWR@afterendverbatim%
30 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
31 \else%

```

Inline, so create the closing </span>:

```

32 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
33 \fi%

```

Final listings deinit:

```
34 \LWR@origlst@DeInit%
35 }
```

```
\lst@MakeCaption {</b>}
```

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```
36 \def\lst@MakeCaption#1{%
37 \LWR@traceinfo{MAKING CAPTION at #1}%
38 \lst@ifdisplaystyle
39 \LWR@traceinfo{making a listings display caption}%
40 \ifx #1t%
41 \ifx\lst@@caption\@empty\expandafter\lst@HRefStepCounter \else
42 \expandafter\refstepcounter
43 \fi {lstlisting}%
44 \LWR@traceinfo{About to assign label: !\lst@label!}%
45 % \ifx\lst@label\@empty\else
46 % \label{\lst@label}\fi
47 \LWR@traceinfo{Finished assigning the label.}%
48 \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
49 \global\let\lst@name\lst@arg \global\let\lstname\lst@name
50 \lst@ifnolol\else
51 \ifx\lst@@caption\@empty
52 \ifx\lst@caption\@empty
53 \ifx\lst@intname\@empty \else \def\lst@temp{ }%
54 \ifx\lst@intname\lst@temp \else
```

This code places a contents entry for a non-float. This would have to be modified for lwarp:

```
55 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
56 % \addcontentsline{lol}{lstlisting}{\lst@name}
57 \fi\fi
58 \fi
59 \else
```

This would have to be modified for lwarp:

```
60 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
61 \addcontentsline{lol}{lstlisting}%
62 {\protect\numberline{\thelstlisting}%
63 {\protect\ignorespaces \lst@@caption \protect\relax}}%
64 \fi
65 \fi
66 \fi
67 \ifx\lst@caption\@empty\else
68 \LWR@traceinfo{lst@caption not empty-}%
69 \lst@ifsubstring #1\lst@captionpos
70 {\begingroup
```

```
71 \LWR@traceinfo{at the selected position}%
```

These space and box commands are not needed for HTML output:

```
72 %           \let\@vskip\vskip
73 %           \def\vskip{\afterassignment\lst@vskip \@tempskipa}%
74 %           \def\lst@vskip{\nobreak\@vskip\@tempskipa\nobreak}%
75 %           \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
76 %           \ifx #1t\allowbreak \fi
77 %           \ifx\lst@title\@empty
```

New lwarp code to create a caption:

```
78 %           \lst@makecaption\fnnum\lstlisting{\ignorespaces \lst@caption}
79 %           \else
```

New lwarp code to create a title:

```
80 %           \lst@maketitle\lst@title % (AS)
81 \LWR@traceinfo{Making title: \lst@title}%
82 \begin{BlockClass}{\lstlistingtitle}% lwarp
83 \lst@maketitle\lst@title% lwarp
84 \end{BlockClass}% lwarp
85 %           \fi
86 \LWR@traceinfo{About to assign label: !\lst@label!}%
87 %           \ifx\lst@label\@empty\else
88 \leavevmode% gets rid of bad space factor error
89 \GetTitleStringExpand{\lst@caption}%
90 \edef\LWR@lntemp{\GetTitleStringResult}%
91 \edef\@currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
92 \label{\lst@label}\fi
93 \LWR@traceinfo{Finished assigning the label.}%
```

Not needed for lwarp:

```
94 %           \ifx #1b\allowbreak \fi
95 %           \endgroup}{}%
96 %           \fi
97 \LWR@traceinfo{end of making a listings display caption}%
98 %           \else
99 \LWR@traceinfo{INLINE}%
100 %           \fi
101 \LWR@traceinfo{DONE WITH CAPTION at #1}%
102 }
```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```
103 \lst@Key{numbers}{none}{%
104 %   \let\lst@PlaceNumber\@empty
105 %   \lstKV@SwitchCases{#1}%
106 %   {none&\\%
107 %     left&\def\lst@PlaceNumber{%
```

---

```

108% \llap{
109 \LWR@originormalfont%
110 \lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
111% }
112}
113\\%
114    right&\def\lst@PlaceNumber{\rlap{\LWR@originormalfont
115                                   \kern\VerbatimHTMLWidth \kern\lst@numbersep
116                                   \lst@numberstyle{\thelstnumber}}}%
117    }\PackageError{Listings}{Numbers #1 unknown}\@ehc}}

118 \end{warpHTML}

```

---

## File 85 `lwarp-longtable.sty`

### \$ 164 Package **longtable**

Pkg `longtable` `longtable` is emulated during HTML output, and the `longtable` package is ignored.

for HTML output: `\LWR@ProvidesPackageDrop{longtable}`



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

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

instead of

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

Doing so helps avoid “Misplaced `\noalign`.” when using `\begin{warpprint}`.

Keep the `\endfirsthead` row, which is still relevant to HTML output.



`\kill` is ignored, place a `\kill` line inside

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

or place it inside `\warpingprintonly`.

See:

<http://tex.stackexchange.com/questions/43006/why-is-input-not-expandable>

Env `longtable` \* [*horizontal*] [*colspec*] Emulates the `longtable` environment.

Per the caption package, the starred version steps the counter per caption. The unstarred version steps the counter once at the beginning, but not at each caption.

Options [c], [l], and [r] are thrown away.

```

2 \newenvironment{longtable*}[2][\{%
3 \LWR@floatbegin{table}%
4 \setcaptiontype{\LTcaption}%
5 \caption@setoptions{longtable}%
6 \caption@setoptions{@longtable}%
7 \caption@LT@setup%
8 \booltrue{LWR@starredlongtable}%
9 \let\captionlistentry\LWR@LTcaptionlistentry%
10 \LWR@tabular{#2}
11 }
12 {\endLWR@tabular\LWR@floatend}
13
14 \newenvironment{longtable}[2][\{%
15 \LWR@floatbegin{table}%
16 \setcaptiontype{\LTcaption}%
17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{\LTcaption}%
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@tabularendoffline}% throws away options //[dim] and /**
28 \def\endfirsthead{\LWR@tabularendoffline}
29 \def\endfoot{\LWR@tabularendoffline}
30 \def\endlastfoot{\LWR@tabularendoffline}
31 \newcommand\tabularnewline{\LWR@tabularendoffline}
32 \newcommand{\setlongtables}{}% Obsolete command, does nothing.
33 \newlength{\LTleft}
34 \newlength{\LTRight}
35 \newlength{\LTpre}
36 \newlength{\LTpost}
37 \newlength{\LTcapwidth}

38 \renewcommand*{\kill}{\LWR@tabularendoffline}

```



---

File 86 **lwarp-lscape.sty**

§ 165    Package **lscape**

Pkg    `lscape`    `lscape` is nullified.

for HTML output:    Discard all options for `lwarp-lscape`.

```
1 \LWR@ProvidesPackageDrop{lscape}

2 \newenvironment*{landscape}{}{}
```

---

File 87 **lwarp-ltcaption.sty**

§ 166    Package **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}{}

```


---

File 88 **lwarp-luatodonotes.sty**

§ 167    Package **luatodonotes**

Pkg    `luatodonotes`    `luatodonotes` is emulated.

`todonotes` and `luatodonotes` are only emulated as if they were loaded with the `disable` option. All functions are disabled, but source compatibility is provided with the print version.

 **xkeyvalue**    `todonotes` and `luatodonotes` use `xkeyval`, so if the user's document uses `\presetkeys`

then xkeyval will have to be loaded explicitly by the user's document.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{luatodonotes}

2 \newcommand{\listoftodos}[1] [] {}
3 \newcommand{\todo}[2] [] {}
4 \newcommand{\missingfigure}[2] [] {}
5 \newcommand{\todotoc}{}
6 \newcommand{\todoarea}[3] [] {}%
```

---

File 89 **lwarp-marginfit.sty**

\$ 168     Package **marginfit**

Pkg marginfit Emulated.

**for HTML output:** Discard all options for lwarp-marginfit:

```

1 \LWR@ProvidesPackageDrop{marginfit}
```

---

File 90 **lwarp-marginfix.sty**

\$ 169     Package **marginfix**

Pkg marginfix Emulated.

**for HTML output:** Discard all options for lwarp-marginfix:

```

1 \LWR@ProvidesPackageDrop{marginfix}

2 \newcommand*{\margin skip}[1] {}
3 \newcommand*{\clearmargin}{}
4 \newcommand*{\softclearmargin}{}
5 \newcommand*{\extendmargin}[1] {}
6 \newcommand*{\mparshift}[1] {}
7 \newdimen\marginheightadjustment
8 \newdimen\marginposadjustment
9 \newcommand*{\blockmargin}[1] [] {}
10 \newcommand*{\unblockmargin}[1] [] {}
11 \newcommand*{\marginphantom}[2] [] {}
```

---

File 91 **lwarp-marginnote.sty**

§ 170      Package **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}{}

```

---

File 92 **lwarp-mcaption.sty**

§ 171      Package **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}

```

---

File 93 **lwarp-mdframed.sty**

§ 172      Package **mdframed**

Pkg    mdframed    mdframed is loaded with options forced to framemethod=none.

### § 172.1 Package loading

**for HTML output:**

```
1 \RequirePackage{xcolor}% for \convertcolorspec
2 \LWR@ProvidesPackageDrop{mdframed}
```

amsthm must be loaded before mdframed

```
3 \LWR@origRequirePackage{amsthm}
```

Do not require Tikz or pstricks:

```
4 \LWR@origRequirePackage[framemethod=none]{mdframed}
```

### § 172.2 Limitations

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

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

**font** For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

where `\textbf` must appear just before the comma and will receive the following text as its argument (since the text happens to be between braces in the mdframed source). Since lwarp does not support `\bfseries` and friends, only one font selection may be made at a time.

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

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

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

**CSS classes** Environments created or encapsulated by mdframed are enclosed in a `<div>` of class `md<environmentname>`, or mdframed otherwise.

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

Pre-existing hooks are used to patch extra functions before and after the frames.

To handle CSS and paragraphs, patch code at start and end of environment and contents. \LWR@origraggedright helps avoid hyphenation.

```

5 \mdfsetup{
6 startcode={\LWR@mdframedstart\LWR@origraggedright},
7 endcode={\LWR@mdframedend},
8 startinnercode={\LWR@startpars\LWR@origraggedright},
9 endinnercode={\LWR@stoppars},
10 }

```

### § 172.3 Color and length HTML conversion

\LWR@mdfprintcolor    {\<mdfcolorkey>}

Given the mdframed key, print the color.

```

11 \newcommand*{\LWR@mdfprintcolor}[1]{%
12 \convertcolorspec{named}{\csuse{mdf@#1}}{HTML}\LWR@tempcolor%
13 \#\LWR@tempcolor
14 }

```

\LWR@mdfprintlength    {\<mdflengthkey>}

Given the mdframed key, print the length.

```

15 \newcommand*{\LWR@mdfprintlength}[1]{%
16 \rndprintlength{\csuse{mdf@#1@length}}
17 }

```

### § 172.4 Environment encapsulation

\LWR@mdframedstart    Actions before an mdframe starts.

Encapsulate a frame inside a <div> of the desired class.

```

18 \newcommand*{\LWR@mdframedstart}{%

```

Turn off paragraph handling during the generation of the encapsulating tags:

```

19 \LWR@stoppars%

```

Below, print HTML pt units:

```

20 \uselengthunit{PT}%

```

Open a <div> and with custom class and custom style:

```

21 \LWR@htmltagc{div class="\LWR@mdthisenv" \LWR@orignewline
22 style=" \LWR@orignewline

```

Convert and print the background color:

```

23 background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline

```

Convert and print the border color and width:

```

24 border: \LWR@mdfprintlength{linewidth} solid
25 \LWR@mdfprintcolor{linecolor} ; \LWR@orignewline

```

Convert and print the border radius:

```

26 border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline

```

Convert and print the shadow:

```

27 \ifbool{mdf@shadow}{%
28     box-shadow:
29     \LWR@mdfprintlength{shadowsize}
30     \LWR@mdfprintlength{shadowsize}
31     \LWR@mdfprintlength{shadowsize}
32     \LWR@mdfprintcolor{shadowcolor} ;
33 }
34 {box-shadow: none ;}
35 \LWR@orignewline

36 "}
37 % \LWR@htmldivclass{\LWR@mdthisenv}

```

mdframed environment may not work with the HTML versions of the following, so restore them to their originals while inside mdframed:

```

38 \LetLtxMacro{\hspace}{\LWR@origspace}%
39 \LetLtxMacro\rule\LWR@origrule%
40 \LetLtxMacro\makebox\LWR@origmakebox%
41 }

```

`\LWR@mdframedend`    Actions after an mdframe ends.

After closing the <div>, globally restore to the default environment type:

```

42 \newcommand*{\LWR@mdframedend}{

```

Close the custom <div>:

```

43 \LWR@htmldivclassend{\LWR@mdthisenv}

```

Reset future custom class to the default:

```

44 \gdef\LWR@mdthisenv{mdframed}

```

Resume paragraph handling:

```

45 \LWR@startpars%
46 }

```

## § 172.5 Titles and subtitles

`\mdfframedtitleenv`    `{\title}`

Encapsulation of the original which places the title inside a `<span>` of class `mdfframedtitle`:

```
47 \LetLtxMacro\LWR@origmdfframedtitleenv\mdfframedtitleenv
48
49 \newlength{\LWR@titleroundcorner}
50
51 \renewrobustcmd\mdfframedtitleenv[1]{%
52 \LWR@origmdfframedtitleenv{%
```

Below, print HTML pt lengths:

```
53 \uselengthunit{PT}%
```

Open a `<span>` with a custom class and custom style:

```
54 \LWR@htmltagc{span class="mdfframedtitle" \LWR@orignewline
55 style=" \LWR@orignewline
```

Convert and print the title background color:

```
56 background:
57 \LWR@mdfprintcolor{frametitlebackgroundcolor}
58 ; \LWR@orignewline
```

Convert and print the title rule:

```
59 \ifbool{mdf@frametitlerule}{%
60   border-bottom:
61   \LWR@mdfprintlength{frametitlerulewidth}
62   solid
63   \LWR@mdfprintcolor{frametitlerulecolor}
64   ; \LWR@orignewline
65 }{}%
```

The title's top border radius is adjusted for the line width:

```
66 border-radius:
67 \setlength{\LWR@titleroundcorner}
68   {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{0pt}}
69   \rndprintlength{\LWR@titleroundcorner}
70   \rndprintlength{\LWR@titleroundcorner}
71   Opt Opt
72   \LWR@orignewline
```

Finish the custom style and the opening span tag:

```
73 " \LWR@orignewline
74 }% span
```

Restrict paragraph tags inside a span:

```
75 \begin{LWR@nestspan}%
```

Print the title inside the span:

```
76 #1%
```

Close the span and unnest the paragraph tag restriction:

```
77 \LWR@htmltagc{/span}%
78 \end{LWR@nestspan}%
79 }
80 }
```

```
\LWR@mdfsubtitlecommon {\langle sub -or- subsub \rangle} [\langle options \rangle] {\langle title \rangle}
```

Common code for \LWR@mdfsubtitle and \LWR@mdfsubsubsubtitle.

Encapsulate the subtitle inside a <span> of class mdframedsubtitle:

```
81 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
82 {% the following empty line is required
83
```

Special handling for mdframed: Subtitles have \pars around them, so temporarily disable them here.

```
84 \let\par\LWR@origpar%
```

Open a <span> with a custom class and custom style:

```
85 \LWR@htmltagc{span class="mdframed#1title"
86 style=" \LWR@orignewline
```

Convert and print the background color:

```
87 background:
88 \LWR@mdfprintcolor{#1titlebackgroundcolor}
89 ; \LWR@orignewline
```

Convert and print the above line:

```
90 \ifbool{mdf@#1titleaboveline}{%
91   border-top:
92   \LWR@mdfprintlength{#1titleabovelinewidth}
93   solid
94   \LWR@mdfprintcolor{#1titleabovelinecolor}
95   ; \LWR@orignewline
96 }{}%
```

Convert and print the below line:

```
97 \ifbool{mdf@#1titlebelowline}{%
98   border-bottom:
99   \LWR@mdfprintlength{#1titlebelowlinewidth}
100   solid
101   \LWR@mdfprintcolor{#1titlebelowlinecolor}
102   ; \LWR@orignewline
103 }{}%
```



Finish the custom style and the opening span tag:

```
104 "% span
```

Restrict paragraph tags inside a span:

```
105 \begin{LWR@nestspan}%
```

Perform the original subtitle action:

```
106 \IfNoValueTF{#2}
107 {\csuse{LWR@origmdf#1title}{#3}}%
108 {\csuse{LWR@origmdf#1title}[#2]{#3}}%
```

Close the span and unnest the paragraph tag restriction:

```
109 \LWR@htmltagc{/span}% the following empty line is required
110 \end{LWR@nestspan}% must follow the /span or an extra <p> appears
111
112 }
```

```
\LWR@mdfsubtitle    [⟨options⟩] {⟨title⟩}
```

```
113 \let\LWR@origmdfsubtitle\mdfsubtitle
114
115 \newcommand*{\LWR@mdfsubtitle}{%
116 \LWR@mdfsubtitlecommon{sub}%
117 }
118 \let\mdfsubtitle\LWR@mdfsubtitle
```

```
\LWR@mdfsubsubtitle    [⟨options⟩] {⟨title⟩}
```

```
119 \let\LWR@origmdfsubsubtitle\mdfsubsubtitle
120
121 \newcommand*{\LWR@mdfsubsubtitle}{%
122 \LWR@mdfsubsubtitlecommon{subsub}%
123 }
124 \let\mdfsubsubtitle\LWR@mdfsubsubtitle
```

## § 172.6 New environments

`\LWR@mdthisenv` Stores the environment of the frame about to be created:

```
125 \newcommand*{\LWR@mdthisenv}{mdframed}
```

```
\newmdenv    [⟨options⟩] {⟨env-name⟩}
```

Modified from the original to remember the environment.

```
126 \renewrobustcmd*\newmdenv[2][{}]{%
127 \newenvironment{#2}%
128 {%
129 \mdfsetup{#1}%
```

```

130 \renewcommand*{\LWR@mdthisenv}{md#2}%
131 \begin{mdframed}%
132 }
133 {\end{mdframed}}%
134 }

```

`\surroundwithmdframed` [*options*] {*environment*}

Modified from the original to remember the environment.

```

135 \renewrobustcmd*{\surroundwithmdframed}[2] [] {%
136 \BeforeBeginEnvironment{#2}{%
137 \renewcommand*{\LWR@mdthisenv}{md#2}%
138 \begin{mdframed}[#1]}%
139 \AfterEndEnvironment{#2}{\end{mdframed}}%
140 }

```

`\mdtheorem` [*mdframed-options*] envname [*numberedlike*] {*caption*} [*within*]

Modified from the original to remember the environment.

```

141 \DeclareDocumentCommand{\mdtheorem}{ O{} m o m o }%
142 {\ifcsdef{#2}%
143   {\mdf@PackageWarning{Environment #2 already exists\MessageBreak}}%
144   {%
145     \IfNoValueTF {#3}%
146     {%#3 not given -- number relationship
147       \IfNoValueTF {#5}%
148       {%#3+#5 not given
149         \@definecounter{#2}%
150         \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
151         \newenvironment{#2}[1] [] {%
152           \refstepcounter{#2}%
153           \ifstrepty{##1}%
154             {\let\@temptitle\relax}%
155             {%
156               \def\@temptitle{\mdf@theoremseparator%
157                 \mdf@theoremspace%
158                 \mdf@theoremtitlefont%
159                 ##1}%
160               \mdf@thm@caption{#2}{\csname the#2\endcsname}{##1}}%
161             }%
162           \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
163             \@temptitle}]]%
164             {\end{mdframed}}%
165           \newenvironment{#2*}[1] [] {%
166             \ifstrepty{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
167             \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]%
168             {\end{mdframed}}%
169             }%

```

```

170      {%#5 given -- reset counter
171      \@definecounter{#2}\@newctr{#2}[#5]%
172      \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
173      \expandafter\xdef\csname the#2\endcsname{%
174          \expandafter\noexpand\csname the#5\endcsname \@thmcountersep%
175              \@thmcounter{#2}}%
176      \newenvironment{#2}[1][]{%
177          \refstepcounter{#2}%
178          \ifstrepty{##1}%
179              {\let\@temptitle\relax}%
180              {%
181                  \def\@temptitle{\mdf@theoremseparator%
182                      \mdf@theoremspace%
183                      \mdf@theoremtitlefont%
184                      ##1}%
185                      \mdf@thm@caption{#2}{\csname the#2\endcsname}{##1}}%
186              }
187          \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
188              \@temptitle}}}%
189          {\end{mdframed}}%
190      \newenvironment{#2*}[1][]{%
191          \ifstrepty{##1}%
192              {\let\@temptitle\relax}%
193              {%
194                  \def\@temptitle{\mdf@theoremseparator%
195                      \mdf@theoremspace%
196                      \mdf@theoremtitlefont%
197                      ##1}%
198                      \mdf@thm@caption{#2}{\csname the#2\endcsname}{##1}}%
199              }%
200          \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}}}%
201          {\end{mdframed}}%
202      }%
203  }%
204  {%#3 given -- number relationship
205      \global\@namedef{the#2}{\@nameuse{the#3}}%
206      \newenvironment{#2}[1][]{%
207          \refstepcounter{#3}%
208          \ifstrepty{##1}%
209              {\let\@temptitle\relax}%
210              {%
211                  \def\@temptitle{\mdf@theoremseparator%
212                      \mdf@theoremspace%
213                      \mdf@theoremtitlefont%
214                      ##1}%
215                      \mdf@thm@caption{#2}{\csname the#2\endcsname}{##1}}%
216              }
217          \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
218              \@temptitle}}}%
219          {\end{mdframed}}%

```

```

220     \newenvironment{#2*}[1] [] {%
221         \ifstrempy{##1}{\let\@temptitle\relax}{\def\@temptitle{:\ ##1}}%
222         \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]}%
223         {\end{mdframed}}%
224     }%
225     \BeforeBeginEnvironment{#2}{\renewcommand*\LWR@mdthisenv}{md#2}}% new
226     \BeforeBeginEnvironment{#2*}{\renewcommand*\LWR@mdthisenv}{md#2}}% new
227 }%
228 }

```

`\newmdtheoremenv` [*mdframed-options*] envname [*numberedlike*] {*caption*} [*within*]

Modified from the original to remember the environment.

```

229 \DeclareDocumentCommand\newmdtheoremenv{0}{m o m o }{%
230     \ifboolexpr{ test {\IfNoValueTF {#3}} and test {\IfNoValueTF {#5}} }%
231     {\newtheorem{#2}{#4}}%
232     {%
233         \IfValueT{#3}{\newtheorem{#2}[#3]{#4}}%
234         \IfValueT{#5}{\newtheorem{#2}{#4}[#5}}%
235     }%
236     \BeforeBeginEnvironment{#2}{%
237         \renewcommand*\LWR@mdthisenv}{md#2}}%
238     \begin{mdframed}[#1]}%
239     \AfterEndEnvironment{#2}{%
240         \end{mdframed}}%
241 }

```

---

File 94 **lwarp-metalogo.sty**

§ 173     Package **metalogo**

Pkg metalogo metalogo is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{metalogo}

```

2 \newcommand\setlogokern[2]{}
3 \newcommand\setlogodrop[2][XeTeX]{}
4 \newcommand\setLaTeXa[1]{}
5 \newcommand\setLaTeXee[1]{}
6 \newcommand\seteverylogo[1]{}
7 \newcommand\everylogo[1]{}

```

File 95 **lwarp-microtype.sty**§ 174 Package **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

```

---

File 96 **lwarp-midfloat.sty**

§ 175     Package **midfloat**

Pkg midfloat midfloat is emulated.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{midfloat}

2 \newenvironment{strip}[1] [] {}{}  
 3 \newskip\stripsep

---

File 97 **lwarp-moreverb.sty**

§ 176     Package **moreverb**

Pkg moreverb moreverb is supported with some patches.

**for HTML output:**    1 \begin{warpHTML}

2 \LWR@ProvidesPackagePass{moreverb}

3 \BeforeBeginEnvironment{verbatim}{%  
 4 \LWR@forcenewpage  
 5 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace\*{-\baselineskip}%  
 6 }  
 7 \AfterEndEnvironment{verbatim}{%  
 8 \LWR@afterendverbatim%  
 9 }  
 10  
 11  
 12 \LetLtxMacro\LWRMV@orig@verbatiminput\@verbatiminput  
 13  
 14 \renewcommand{\@verbatiminput}[2] [] {%  
 15 \LWR@forcenewpage  
 16 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace\*{-\baselineskip}%  
 17 \LWRMV@orig@verbatiminput[#1]{#2}%  
 18 \LWR@afterendverbatim%  
 19 }  
 20  
 21 \BeforeBeginEnvironment{listing}{%  
 22 \LWR@forcenewpage  
 23 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace\*{-\baselineskip}%

---

```

24 }
25
26 \AfterEndEnvironment{listing}{%
27 % \unskip\LWR@origvspace*{-\baselineskip}%
28 \LWR@afterendverbatim%
29 }
30
31 \BeforeBeginEnvironment{listingcont}{%
32 \LWR@forcenewpage
33 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
34 }
35
36 \AfterEndEnvironment{listingcont}{%
37 % \unskip\LWR@origvspace*{-\baselineskip}%
38 \LWR@afterendverbatim%
39 }

40 \LetLtxMacro\LWRMV@@listinginput\@listinginput
41
42 \renewcommand{\@listinginput}[3][]{
43 \LWR@forcenewpage
44 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
45 \LWRMV@@listinginput[#1]{#2}{#3}
46 \LWR@afterendverbatim%
47 }
48
49
50 \renewenvironment*{boxedverbatim}
51 {
52 \LWR@forcenewpage
53 \LWR@atbeginverbatim{boxedverbatim}\unskip\LWR@origvspace*{-\baselineskip}%
54 \verbatim%
55 }
56 {
57 \endverbatim%
58 \LWR@afterendverbatim%
59 }
60
61

62 \end{warpHTML}

```

---

File 98 **lwarp-morewrites.sty**

§ 177 Package **morewrites**

Pkg **morewrites** Error if morewrites is loaded after lwarp.

Discard all options for lwarp-morewrites:

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{morewrites}
2 \LWR@loadbefore{morewrites}
```

---

File 99 **lwarp-mparhack.sty**

§ 178     Package **mparhack**

Pkg   mparhack   Emulated.

**for HTML output:**     Discard all options for lwarp-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```

---

File 100 **lwarp-multicol.sty**

§ 179     Package **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}
```

Env   multicols   \* {\<numcols>} [\<heading>]

```
3 \NewDocumentEnvironment{multicols}{s m o}
```

HTML <div> class to contain everything:

```
4 {
5 \LWR@forcenewpage
6 \BlockClass{multicols}
```

Optional HTML <div> class for the heading:

```
7 \IfValueT{#3}{\begin{BlockClass}{multicolsheading}#3\end{BlockClass}}}
```



When done with the environment, close the <div>:

```
8 {\endBlockClass}
```

Emulated null functions which are not used in HTML:

```
9 \newcommand*{\columnbreak}{}
10 \newcommand*{\RLmulticolcolumns}{}
11 \newcommand*{\LRmulticolcolumns}{}
12
13 \newlength{\premulticols}
14 \newlength{\postmulticols}
15 \newlength{\multicolsep}
16 \newlength{\multicolbaselineskip}
17 \newlength{\multicoltolerance}
18 \newlength{\multicolpretolerance}
19 \newcommand*{\columnseprulecolor}{\normalcolor}
20 \newcounter{columnbadness}
21 \newcounter{finalcolumnbadness}
22 \newcounter{collectmore}
23 \newcounter{unbalance}
24 \newlength{\multicolovershoot}
25 \newlength{\multicolundershoot}

26 \end{warpHTML}
```

---

File 101 **lwarp-multirow.sty**

§ 180    Package **multirow**

Pkg    multirow    multirow is emulated during HTML output, and used as-is while inside a lateximage.

In a lateximage, the original print-mode versions are temporarily restored by `\LWR@restoreorigformatting`.

See section 59.18 for the print-mode versions.

for HTML output:    1 \LWR@ProvidesPackagePass{multirow}

Remember the print-mode version:

```
2 \LetLtxMacro\LWR@origmultirow\multirow
```

\LWR@multirowborder    Set to left or right to create a thick border for the cell, for use by bigdelim:

```
3 \newcommand{\LWR@multirowborder}{}

```

### § 180.1 **Multirow**

`\multirow` [*<vpos>*] [*<numrows>*] [*<bigstruts>*] [*<width>*] [*<fixup>*] [*<text>*]

```
4 \RenewDocumentCommand{\multirow}{0{c} m o m o +m}%
5 {%
6 \LWR@traceinfo{*** multirow #1 #2 #4}
7 \LWR@maybenewtablerow%
8 \LWR@tabularleftedge%
```

Print the start of a new table data cell:

```
9 \LWR@htmltag{td rowspan="#2" %
```

The vertical alignment, if given:

```
10 \IfValueT{#1}{%
11 \ifstrequal{#1}{b}{style="vertical-align:bottom" }{%
12 \ifstrequal{#1}{t}{style="vertical-align:top" }{%
13 }%
```

The left/right border, if given:

```
14 \ifdefvoid{\LWR@multirowborder}{}%
15 style="border-\LWR@multirowborder: 2px dotted black ; %
16 padding-\LWR@multirowborder: 2px" %
17 }%
```

A class adds the column spec and the rule:

```
18 class="td%
```

Append this column's spec:

```
19 \StrChar{\LWR@tablecolspec}{\arabic{\LWR@tablecolindex}}%
```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag. Also add the vertical bar class.

```
20 \LWR@addcmidruletrim%
21 \LWR@addleftmostbartag%
22 \LWR@printbartag{\arabic{\LWR@tablecolindex}}%
23 "%
```

```
24 \LWR@tdstartstyles%
25 \LWR@addcmidrulewidth%
26 \LWR@addformatwpalignment%
27 \LWR@tdendstyles%
```

```
28 }%
```

The column's < spec:

```
29 \LWR@gettexpparray{LWR@colbefore spec}{\arabic{LWR@tablecolindex}}%
```

While printing the text, redefine \ to generate a new line

```
30 \begingroup\LetLtxMacro{\}\{\LWR@endofline}\#6\endgroup%
31 \LWR@stoppar%
32 \global\boolfalse{LWR@intabularmetadata}%
33 \renewcommand{\LWR@multirowborder}{}%
34 \LWR@traceinfo{*** multirow done}%
35 }%
```

## § 180.2 Combined multicolumn and multirow

⚠ `\multicolumn & \multirow`

lwarp does not support directly combining `\multicolumn` and `\multirow`. Use `\multicolumnrow` instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}
```

The two arguments for `\multicolumn` come first, followed by the five arguments for `\multirow`, many of which are optional, followed by the contents.

⚠ **skipped cells**

As per `\multirow`, skipped cells to the right of the `\multicolumnrow` statement are not included in the source code on the same line. On the following lines,

⚠ **empty cells**

`\mcolrowcell` must be used for each cell of each column and each row to be skipped:

```
... & \multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text} & ...
... & \mcolrowcell & \mcolrowcell & ...
... & \mcolrowcell & \mcolrowcell & ...
```

**vposn** Note that recent versions of multirow include a new optional `vposn` argument.

```
\multicolumnrow {<1:cols>} {<2:halign>} [<3:vpos>] {<4:numrows>} [<5:bigstruts>] {<6:width>} [<7:fixup>]
{<8:text>}
```

```
36 \NewDocumentCommand{\multicolumnrow}{m m 0{} m 0{} m 0{} +m}{%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
37 \LWR@tabularhtmlcolumns{\arabic{LWR@tablecolindex}}{#1}
```

Create the multicolumn/multirow tag:

```

38 \begingroup%
39 \LetLtxMacro{\}\{LWR@endofline}%
40 \LWR@domulticolumn[#3][#4]{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#8}%
41 \endgroup%

```

Move to the next  $\text{\LaTeX}$  column:

```

42 \addtocounter{LWR@tablecolindex}{#1}%
43 \addtocounter{LWR@tablecolindex}{-1}%

```

Skip any trailing @ or ! columns for this cell:

```

44 \booltrue{LWR@skipatbang}%
45 }

46 \appto{LWR@restoreorigformatting}{%
47 \LetLtxMacro\multirow\LWR@origmultirow%
48 \renewcommand{\multicolumnrow}{LWR@origmulticolumnrow}%
49 }

```

---

File 102 **lwarp-nameref.sty**

§ 181     Package **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

```

---

File 103 **lwarp-needspace.sty**

§ 182     Package **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}{}

```

---

File 104 **lwarp-newclude.sty**

§ 183    Package **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}

```

---

File 105 **lwarp-newunicodechar.sty**

§ 184    Package **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}

```

---

File 106 **lwarp-nextpage.sty**

§ 185    Package **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] [] {}

```

---

File 107 **lwarp-nonumonpart.sty**

§ 186     Package **nonumonpart**

Pkg   nonumonpart   nonumonpart is ignored.

for HTML output:    1 \LWR@ProvidesPackageDrop{nonumonpart}

---

File 108 **lwarp-nopageno.sty**

§ 187     Package **nopageno**

Pkg   nopageno   nopageno is ignored.

for HTML output:    1 \LWR@ProvidesPackageDrop{nopageno}

---

File 109 **lwarp-nowidow.sty**

§ 188     Package **nowidow**

Pkg   nowidow   nowidow is not used during HTML conversion.

Discard all options for lwarp-nowidow:

for HTML output:    1 \LWR@ProvidesPackageDrop{nowidow}

```

\nowidow    [⟨lines⟩]
\setnowidow [⟨lines⟩]

2 \newcommand*{\nowidow}[1] [] {}
3 \newcommand*{\setnowidow}[1] [] {}

\noclub    [⟨lines⟩]
\setnoclub [⟨lines⟩]

4 \newcommand*{\noclub}[1] [] {}
5 \newcommand*{\setnoclub}[1] [] {}

```

File 110 `lwarp-nttheorem.sty`

§ 189     Package     **nttheorem**

*(Based on original code by WOLFGANG MAY, ANDREAS SCHEDLER.)*

Pkg    nttheorem    nttheorem is patched for use by lwarp.

Table 13: Ntheorem package — 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.


§ 189.1     **Limitations**

 **Font control**    This conversion is not total. Font control is via CSS, and the custom  $\TeX$  font settings are ignored.

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

§ 189.2     **Options**

Options `amsthm` or `standard` choose which set of theorems and proofs to initialize.

 **Disabled options**    The options `thmmarks` and `amsmath` are disabled, since they heavily modify the underlying math code. Theorem marks are emulated. The AMS-math modifications are not done.

Option `thref` is disabled because `cleveref` functions are used instead. `\thref` is emulated.

Option `hyperref` is disabled because lwarp emulated `hyperref`.

**for HTML output:**    Some disabled options:

```
1 \DeclareOption{thref}{}
2
3
```

```

4 \newbool{LWR@theoremmarks}
5 \boolfalse{LWR@theoremmarks}
6
7 \DeclareOption{thmmarks}{
8 \booltrue{LWR@theoremmarks}
9 \newif\ifsetendmark\setendmarktrue
10 }
11
12
13 \newbool{LWR@theoremamsthm}
14 \boolfalse{LWR@theoremamsthm}
15
16 \DeclareOption{amsthm}{\booltrue{LWR@theoremamsthm}}
17
18
19 \DeclareOption{amsmath}{}
20 \DeclareOption{hyperref}{}
21
22
23 \LWR@ProvidesPackagePass{theorem}

```

### § 189.3 Remembering the theorem style

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     \theoremstyle{plain}
29     \renewcommand{\LWR@newtheoremstyle}{plain}% new
30     }%
31   {
32     \theoremstyle{#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  }

```

#### § 189.4 **HTML cross-referencing**

Mimics a float by incrementing the float counter and generating an HTML anchor. These are used for list-of-theorem cross-references.

```

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 }

```

#### § 189.5 **\newtheoremstyle**

The following are patched for css.

These were in individual files thp.sty for plain, thmb.sty for margin break, etc. They are gathered together here.

Each theorem is encased in a BlockClass environment of class theorembody<style>.

Each header is encased in an \InlineClass of class theoremheader<style>.

```

97 \gdef\newtheoremstyle#1#2#3{%
98   \expandafter\@ifundefined{th@#1}%
99   {\expandafter\gdef\csname th@#1\endcsname{%
100     \def\@begintheorem####1####2{%
101 \LWR@forcenewpage% new
102 \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
103 \LWR@inctheorem% new
104 #2}%
105     \def\@opargbegintheorem####1####2####3{%
106 \LWR@forcenewpage% new
107 \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
108 \LWR@inctheorem% new
109 #3}%
110 }%
111 }%
112 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
113 }

```

§ 189.6 **Standard styles**

```

114 \renewtheoremstyle{plain}%
115   {\item[\hskip\labelsep \theorem@headerfont
116     \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]]}%
117   {\item[\hskip\labelsep \theorem@headerfont
118     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]]}
119
120 \renewtheoremstyle{break}%
121   {\item[
122     % \rlap{\vbox{\hbox{
123       \hskip\labelsep \theorem@headerfont
124       \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
125     % }}\hbox{\strut}}}
126     ]}%
127   {\item[
128     % \rlap{\vbox{\hbox{
129       \hskip\labelsep \theorem@headerfont
130       \InlineClass{theoremheaderbreak}%
131       {##1\ ##2\ (##3)\theorem@separator}\newline
132     % }}\hbox{\strut}}}
133     ]}
134
135 \renewtheoremstyle{change}%
136   {\item[\hskip\labelsep
137     \theorem@headerfont
138     \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]]}%
139   {\item[\hskip\labelsep
140     \theorem@headerfont
141     \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]]}
142
143 \renewtheoremstyle{changebreak}%
144   {\item[
145     % \rlap{\vbox{\hbox{
146       \hskip\labelsep \theorem@headerfont
147       \InlineClass{theoremheaderchangebreak}%
148       {##2\ ##1\theorem@separator}\newline
149     % }}\hbox{\strut}}}
150     ]}%
151   {\item[
152     % \rlap{\vbox{\hbox{
153       \hskip\labelsep \theorem@headerfont
154       \InlineClass{theoremheaderchangebreak}%
155       {##2\ ##1\ (##3)\theorem@separator}\newline
156     % }}\hbox{\strut}}}
157     ]}
158
159 \renewtheoremstyle{margin}%
160   {\item[\hskip\labelsep\theorem@headerfont
161     \InlineClass{theoremheadermargin}{##2 \quad ##1\theorem@separator}

```

```

162    ]}%
163    {\item[\hskip\labelsep\theorem@headerfont
164          \InlineClass{theoremheadermargin}{##2 \quad ##1\ (##3)\theorem@separator}
165    ]}
166
167 \renewtheoremstyle{marginbreak}%
168   {\item[\hskip\labelsep\theorem@headerfont
169         \InlineClass{theoremheadermarginbreak}%
170         {##2 \quad ##1\theorem@separator}\newline
171   ]}%
172   {\item[\hskip\labelsep\theorem@headerfont
173         \InlineClass{theoremheadermarginbreak}%
174         {##2 \quad ##1\ (##3)\theorem@separator}\newline
175   ]}
176
177 \renewtheoremstyle{nonumberplain}%
178   {\item[\theorem@headerfont\hskip\labelsep
179         \InlineClass{theoremheaderplain}{##1\theorem@separator}]}%
180   {\item[\theorem@headerfont\hskip \labelsep
181         \InlineClass{theoremheaderplain}{##1\ (##3)\theorem@separator}]}
182
183 \renewtheoremstyle{nonumberbreak}%
184   {\item[
185 % \rlap{\vbox{\hbox{
186   \hskip\labelsep \theorem@headerfont
187   \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline
188 % }\hbox{\strut}}}]
189   ]}%
190   {\item[
191 % \rlap{\vbox{\hbox{
192   \hskip\labelsep \theorem@headerfont
193   \InlineClass{theoremheaderbreak}{##1\ (##3)\theorem@separator}\newline
194 % }\hbox{\strut}}}]
195   ]}
196
197 \renewtheoremstyle{empty}%
198   {\item[]}%
199   {\item[\theorem@headerfont \hskip\labelsep\relax
200         \InlineClass{theoremheaderplain}{##3}]}
201
202 \renewtheoremstyle{emptybreak}%
203   {\item[]}%
204   {\item[\theorem@headerfont \hskip\labelsep\relax
205         \InlineClass{theoremheaderplain}{##3}] \ \newline}

```

### § 189.7 Additional objects

The following manually adjust the css for the standard configuration objects which are not a purely plain style:

```

206 \ifbool{LWR@theoremamsthm}{-}{%
207 % upright text via CSS
208   \newtheoremstyle{plainupright}{%
209     {\item[\hskip\labelsep \theorem@headerfont
210       \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]]}%
211     {\item[\hskip\labelsep \theorem@headerfont
212       \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]]}
213
214 % upright text and small caps header via CSS
215   \newtheoremstyle{nonumberplainuprightsc}{%
216     {\item[\theorem@headerfont\hskip\labelsep
217       \InlineClass{theoremheadersc}{##1\theorem@separator}]]}%
218     {\item[\theorem@headerfont\hskip \labelsep
219       \InlineClass{theoremheadersc}{##1\ (##3)\theorem@separator}]]}
220 }% not amsthm

```

### § 189.8 Renewed standard configuration

The following standard configuration is renewed using the new css:

```

221 \ifbool{LWR@theoremamsthm}{-}{%
222   \theoremstyle{plainupright}
223   \theorembodyfont{\upshape}
224   \theoremsymbol{\HTMLUnicode{25A1}}% UTF-8 white box
225   \renewtheorem{Example}{Example}
226   \renewtheorem{example}{Example}
227   \renewtheorem{Beispiel}{Beispiel}
228   \renewtheorem{beispiel}{Beispiel}
229   \renewtheorem{Bemerkung}{Bemerkung}
230   \renewtheorem{bemerkung}{Bemerkung}
231   \renewtheorem{Anmerkung}{Anmerkung}
232   \renewtheorem{anmerkung}{Anmerkung}
233   \renewtheorem{Remark}{Remark}
234   \renewtheorem{remark}{Remark}
235   \renewtheorem{Definition}{Definition}
236   \renewtheorem{definition}{Definition}
237
238   \theoremstyle{nonumberplainuprightsc}
239   \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
240   \renewtheorem{Proof}{Proof}
241   \renewtheorem{proof}{Proof}
242   \renewtheorem{Beweis}{Beweis}
243   \renewtheorem{beweis}{Beweis}
244   \qedsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof

```

```

245
246   \theoremsymbol{}
247 }% not amsthm

```

## § 189.9 amsthm option

Only if the amsthm option was given:

```

248 \ifbool{LWR@theoremamsthm}{
249
250 \gdef\th@plain{%
251   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
252   \def\@begintheorem##1##2{%
253     \LWR@forcenewpage% new
254       \BlockClass{theorembodyplain}% new
255       \LWR@inctheorem% new
256       \item[\hskip\labelsep
257 %   \theorem@headerfont
258     \InlineClass{theoremheaderplain}{##1\ ##2.}
259     ]}%
260   \def\@opargbegintheorem##1##2##3{%
261     \LWR@forcenewpage% new
262       \BlockClass{theorembodyplain}% new
263       \LWR@inctheorem% new
264       \item[\hskip\labelsep
265 %   \theorem@headerfont
266     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
267     ]}}
268
269 \gdef\th@nonumberplain{%
270   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
271   \def\@begintheorem##1##2{%
272     \LWR@forcenewpage% new
273       \BlockClass{theorembodyplain}% new
274       \LWR@inctheorem% new
275       \item[\hskip\labelsep
276 %   \theorem@headerfont
277     \InlineClass{theoremheaderplain}{##1.}
278     ]}%
279   \def\@opargbegintheorem##1##2##3{%
280     \LWR@forcenewpage% new
281       \BlockClass{theorembodyplain}% new
282       \LWR@inctheorem% new
283       \item[\hskip\labelsep
284 %   \theorem@headerfont
285     \InlineClass{theoremheaderplain}{##1\ (##3).}
286     ]}}
287
288 \gdef\th@definition{%

```

```

289 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
290 \def\@begintheorem##1##2{%
291 \LWR@forcenewpage% new
292 \BlockClass{theorembodydefinition}% new
293 \LWR@inctheorem% new
294 \item[\hskip\labelsep
295 % \theorem@headerfont
296 \InlineClass{theoremheaderdefinition}{##1\ ##2.}
297 ]}%
298 \def\@opargbegintheorem##1##2##3{%
299 \LWR@forcenewpage% new
300 \BlockClass{theorembodydefinition}% new
301 \LWR@inctheorem% new
302 \item[\hskip\labelsep
303 % \theorem@headerfont
304 \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
305 ]}}
306
307 \gdef\th@nonumberdefinition{%
308 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
309 \def\@begintheorem##1##2{%
310 \LWR@forcenewpage% new
311 \BlockClass{theorembodydefinition}% new
312 \LWR@inctheorem% new
313 \item[\hskip\labelsep
314 % \theorem@headerfont
315 \InlineClass{theoremheaderdefinition}{##1.}
316 ]}%
317 \def\@opargbegintheorem##1##2##3{%
318 \LWR@forcenewpage% new
319 \BlockClass{theorembodydefinition}% new
320 \LWR@inctheorem% new
321 \item[\hskip\labelsep
322 % \theorem@headerfont
323 \InlineClass{theoremheaderdefinition}{##1\ (##3).}
324 ]}}
325
326 \gdef\th@remark{%
327 \def\theorem@headerfont{\itshape}\normalfont%
328 \def\@begintheorem##1##2{%
329 \LWR@forcenewpage% new
330 \BlockClass{theorembodyremark}% new
331 \LWR@inctheorem% new
332 \item[\hskip\labelsep
333 % \theorem@headerfont
334 \InlineClass{theoremheaderremark}{##1\ ##2.}
335 ]}%
336 \def\@opargbegintheorem##1##2##3{%
337 \LWR@forcenewpage% new
338 \BlockClass{theorembodyremark}% new

```

```

339      \LWR@inctheorem% new
340      \item[\hskip\labelsep
341 % \theorem@headerfont
342 \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
343 ]}}
344
345 \gdef\th@nonumberremark{%
346 \def\theorem@headerfont{\itshape}\normalfont%
347 \def\@begintheorem##1##2{%
348 \LWR@forcenewpage% new
349 \BlockClass{theorembodyremark}% new
350 \LWR@inctheorem% new
351 \item[\hskip\labelsep
352 % \theorem@headerfont
353 \InlineClass{theoremheaderremark}{##1.}
354 ]}%
355 \def\@opargbegintheorem##1##2##3{%
356 \LWR@forcenewpage% new
357 \BlockClass{theorembodyremark}% new
358 \LWR@inctheorem% new
359 \item[\hskip\labelsep
360 % \theorem@headerfont
361 \InlineClass{theoremheaderremark}{##1\ (##3).}
362 ]}}
363
364 \gdef\th@proof{%
365 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
366 \def\@begintheorem##1##2{%
367 \LWR@forcenewpage% new
368 \BlockClass{theorembodyproof}% new
369 \LWR@inctheorem% new
370 \item[\hskip\labelsep
371 % \theorem@headerfont
372 \InlineClass{theoremheaderproof}{##1.}
373 ]}%
374 \def\@opargbegintheorem##1##2##3{%
375 \LWR@forcenewpage% new
376 \BlockClass{theorembodyproof}% new
377 \LWR@inctheorem% new
378 \item[\hskip\labelsep
379 % \theorem@headerfont
380 \InlineClass{theoremheaderproof}{##1\ (##3).}
381 ]}}
382
383
384
385 \newcounter{proof}%
386 \if@thmmarks
387 \newcounter{currproofctr}%
388 \newcounter{endproofctr}%

```



```

389 \fi
390
391 \gdef\proofSymbol{\openbox}
392
393 \newcommand{\proofname}{Proof}
394
395 \newenvironment{proof}[1][\proofname]{
396     \th@proof
397     \def\theorem@headerfont{\itshape}%
398     \normalfont
399     \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
400     \@thm{proof}{proof}{#1}
401 }%
402 {\@endtheorem}
403
404 }{}% amsthm option

```

## § 189.10 Ending a theorem

Patched for CSS:

```

405 \let\LWR@origendtheorem\@endtheorem
406 \renewcommand{\@endtheorem}{%
407 \ifbool{LWR@theoremmarks}{%
408     \ifsetendmark%
409     \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
410     \setendmarkfalse%
411     \fi%
412 }{}%
413 \LWR@origendtheorem%
414 \ifbool{LWR@theoremmarks}{\global\setendmarktrue}{}%
415 \endBlockClass%
416 }

```

## § 189.11 \NoEndMark

```

417 \gdef\NoEndMark{\global\setendmarkfalse}

```

## § 189.12 List-of

Redefined to reuse the float mechanism to add list-of-theorem links:

```

\thm@thmline {\langle 1: printed type \rangle} {\langle 2: # \rangle} {\langle 3: optional \rangle} {\langle 4: page \rangle}

418 \renewcommand{\thm@thmline@noname}[4]{%
419 \hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
420 }
421
422 \renewcommand{\thm@thmline@name}[4]{%

```

```

423 \hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
424 }

```

This was redefined by ntheorem when loaded, so it is now redefined for lwarp:

```

425 \def\thm@@thmline{\thm@@thmline@name}

```

Patch for CSS:

```

426 \def\listtheorems#1{
427 \LWR@html@elementclass{nav}{lothm}%
428 \begin@group
429 \c@tocdepth=-2%
430 \def\thm@list{#1}\thm@processlist
431 \end@group
432 \LWR@html@elementclassend{nav}{lothm}%
433 }

```

### § 189.13 Symbols

Proof QED symbol:

```

434 \newcommand{\qed}{\quad\the\qedsymbol}
435
436 \AtBeginDocument{
437 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
438 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
439 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
440 }

```

### § 189.14 Cross-referencing

`\thref {<label>}`

```

441 \newcommand*{\thref}[1]{\cref{#1}}

```

---

File 111 **lwarp-overpic.sty**

## § 190 Package **overpic**

Pkg overpic overpic is patched for use by lwarp.



### **scaling**

The macros `\overpicfontsize` and `\overpicfontskip` are used during HTML generation. These are sent to `\fontsize` to adjust the font size for scaling differences between the print and HTML versions of the document. Renew these macros before using the `overpic` and `Overpic` environments.

See section 70.2 for the print-mode version of `\overpicfontsize` and `\overpicfontskip`.

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{overpic}

2 \newcommand*{\overpicfontsize}{12}
3 \newcommand*{\overpicfontskip}{14}
4
5 \BeforeBeginEnvironment{overpic}{%
6   \lateximage%
7   \fontsize{\overpicfontsize}{\overpicfontskip}%
8   \selectfont%
9 }
10
11 \AfterEndEnvironment{overpic}{\endlateximage}
12
13 \BeforeBeginEnvironment{Overpic}{%
14   \lateximage%
15   \fontsize{\overpicfontsize}{\overpicfontskip}%
16   \selectfont%
17 }
18
19 \AfterEndEnvironment{Overpic}{\endlateximage}
```

---

File 112 **lwarp-pagenote.sty**

§ 191     Package **pagenote**

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

**for HTML output:**

```

1 \DeclareOption{page}{}
2 \LWR@ProvidesPackagePass{pagenote}
```

---

File 113 **lwarp-paralist.sty**

§ 192     Package **paralist**

Pkg    paralist    paralist is supported with minor changes.

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{paralist}

2 \AtBeginEnvironment{compactitem}{\LWR@itemizestart}
3 \AtEndEnvironment{compactitem}{\LWR@itemizeend}
4 \AtBeginEnvironment{compactenum}{\LWR@enumeratestart}
5 \AtEndEnvironment{compactenum}{\LWR@enumerateend}
```

---

```

6 \AtBeginEnvironment{compactdesc}{\LWR@descriptionstart}
7 \AtEndEnvironment{compactdesc}{\LWR@descriptionend}
8 \def\paradescriptionlabel#1{{\normalfont\textbf{#1}}}
```

---

File 114 **lwarp-parskip.sty**

§ 193    Package **parskip**

Pkg    parskip    parskip is ignored.

for HTML output:    Discard all options for lwarp-parskip.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

---

File 115 **lwarp-pdfjscape.sty**

§ 194    Package **pdfjscape**

Pkg    pdfjscape    Emulated.

for HTML output:    Discard all options for lwarp-pdfjscape:

```
1 \LWR@ProvidesPackageDrop{pdfjscape}
```

---

File 116 **lwarp-pdfsync.sty**

§ 195    Package **pdfsync**

Pkg    pdfsync    Emulated.

for HTML output:    Discard all options for lwarp-pdfsync:

```

1 \LWR@ProvidesPackageDrop{pdfsync}


2 \let\pdfsync\relax
3 \let\pdfsyncstart\relax
4 \let\pdfsyncstop\relax
```

---

File 117 **lwarp-pfnote.sty**

§ 196    Package **pfnote**

Pkg   pfnote   pfnote is emulated.

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

**for HTML output:**    1 \LWR@ProvidesPackageDrop{pfnote}

---

File 118 **lwarp-placeins.sty**

§ 197    Package **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}{}

---

File 119 **lwarp-prelim2e.sty**

§ 198    Package **prelim2e**

Pkg   prelim2e   Emulated.

**for HTML output:**    Discard all options for lwarp-prelim2e:

1 \LWR@ProvidesPackageDrop{prelim2e}

2 \newcommand{\PrelimText}{}

3 \newcommand{\PrelimTextStyle}{}

4 \newcommand{\PrelimWords}{}

---

File 120 **lwarp-quotchap.sty**

§ 199     Package **quotchap**

Pkg    quotchap    quotchap is emulated.

*(Based on original code by KARSTEN TINNEFELD, JAN KLEVER.)*

**for HTML output:**    1 \LWR@ProvidesPackageDrop{quotchap}

```
2 \newcommand{\@quotchap}{}
3 \newlength{\LWR@quotchapwidth}
4
5 \let\@printcites\relax
6
7 \newcommand*{\@iprintcites}{%
```

Place the quotes inside a <div> of class quotchap, of the maximum selected width:

```
8 \uselengthunit{PT}%
9 \begin{BlockClass}[max-width: \rndprintlength{\LWR@quotchapwidth}]{quotchap}
10 %\begin{minipage}{\LWR@quotchapwidth}
11 \@quotchap
12 %\end{minipage}
13 \end{BlockClass}
```

Deactivate the quote printing:

```
14 \global\let\@printcites\relax
15 }
16
17 \NewEnviron{savequote}[1][\linewidth]{%
```

Remember the width, adjusted for HTML, and make the length assignment global, per:

<https://tex.stackexchange.com/questions/300823/why-is-setlength-ineffective-inside-a-tabular-environment>

```
18 \setlength{\LWR@quotchapwidth}{#1*2}%
19 \global\LWR@quotchapwidth=\LWR@quotchapwidth%
```

Remember the body, and activate the quote printing:

```
20 \global\let\@quotchap\BODY
```

```
21 \global\let\@printcites\@iprintcites%
22 }
```

The quotation author is placed inside a <div> of class qauthor:

```
23 \newcommand{\qauthor}[1]{\begin{BlockClass}{qauthor}{#1}\end{BlockClass}}
```

\qsetcnfont is ignored:

```
24 \newcommand{\qsetcnfont}[1]{}
```

---

File 121 **lwarp-ragged2e.sty**

§ 200 Package **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}
```

---

File 122 **lwarp-realscripts.sty**

§ 201      Package **realscripts**

Pkg    **realscripts**    realscripts is emulated. See `lwarp.css` for the `<span>` of class `supsubscript`.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{realscripts}

2 \let\realsuperscript\textsuperscript
3 \let\realsubscript\textsubscript
4
5 \let\fakesuperscript\textsuperscript
6 \let\fakeSubscript\textsubscript
7
8 \newlength{\subsupersep}
9
10 \newcommand*{\LWR@realscriptsalign}{}
11
12 \newcommand*{\LWR@setrealscriptsalign}[1]{%
13 \renewcommand*{\LWR@realscriptsalign}{}%
14 \ifthenelse{\equal{#1}{c}}{\renewcommand{\LWR@realscriptsalign}{text-align:center;}}{}%
15 \ifthenelse{\equal{#1}{r}}{\renewcommand{\LWR@realscriptsalign}{text-align:right;}}{}%
16 }
17
18 \DeclareDocumentCommand \textsubsuperscript {s O{1} mm} {%
19 \LWR@setrealscriptsalign{#2}%
20 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
21 \textsuperscript{#4}\textsubscript{#3}%
22 }%
23 }
24
25 \DeclareDocumentCommand \textsupersubscript {s O{1} mm} {%
26 \LWR@setrealscriptsalign{#2}%
27 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
28 \textsubscript{#4}\textsuperscript{#3}%
29 }%
30 }

```

---

File 123 **lwarp-relsize.sty**

§ 202      Package **relsize**


Pkg    **relsize**    relsize is patched for use by lwarp.



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

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

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

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

for HTML output:

```

1 \LWR@ProvidesPackagePass{relsize}

2 \let\LWR@origrelsize\relsize
3 \LetLtxMacro\LWR@origlarger\larger
4 \LetLtxMacro\LWR@origsmaller\smaller
5 \let\LWR@relscale\relscale
6 \LetLtxMacro\LWR@origtextlarger\textlarger
7 \LetLtxMacro\LWR@origtextsmaller\textsmaller
8 \let\LWR@textscale\textscale
9
10 \appto{\LWR@restoreorigformatting}{%
11 \let\relsize\LWR@origrelsize%
12 \LetLtxMacro\larger\LWR@origlarger%
13 \LetLtxMacro\smaller\LWR@origsmaller%
14 \let\relscale\LWR@relscale%
15 \LetLtxMacro\textlarger\LWR@origtextlarger%
16 \LetLtxMacro\textsmaller\LWR@origtextsmaller%
17 \let\textscale\LWR@textscale%
18 }
19
20 \newcounter{LWR@relsizetemp}
21
22 \renewcommand*{\relsize}[1]{%
23 \renewcommand*{\larger}[1][1]{%
24 \renewcommand*{\smaller}[1][1]{%
25 \renewcommand*{\relscale}[1]{%
26
27 \renewcommand*{\textlarger}[2][1]{%
28 \setcounter{LWR@relsizetemp}{100+(#1*20)}%
29 \InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{\textlarger}{#2}%
30 }
31
32 \renewcommand*{\textsmaller}[2][1]{%
33 \setcounter{LWR@relsizetemp}{100-(#1*20)}%
34 \InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{\textsmaller}{#2}%
35 }
36
37 \renewcommand*{\textscale}[2]{%
38 \setcounter{LWR@relsizetemp}{100*\real{#1}}%

```

---

```
39 \InlineClass[font-size:\arabic{LWR@relsize temp}\%]{textscalescale}{#2}%
40 }
```

---

File 124 **lwarp-romanbar.sty**

§ 203     Package **romanbar**

Pkg    romanbar    romanbar is patched for use by lwarp.

An inline class with an overline and underline is used.

**for HTML output:**

```
1 \LWR@ProvidesPackagePass{romanbar}

2 \DeclareRobustCommand{\Roman@bar}[1]{% #1 is in Roman, i.e. MMXII
3 \InlineClass[%
4     text-decoration: overline underline ;
5 ]{romanbar}{#1}%
6 }
```

---

File 125 **lwarp-romanbarpagenumber.sty**

§ 204     Package **romanbarpagenumber**

Pkg    romanbarpagenumber    romanbarpagenumber is ignored.

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{romanbarpagenumber}
```

---

File 126 **lwarp-rotating.sty**

§ 205     Package **rotating**

Pkg    rotating    rotating is emulated during HTML output, and the rotating package is ignored.

All rotations are ignored in HTML output.

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{rotating}

2 \LetLtxMacro\sidewaystable\table
3 \let\endsidewaystable\endtable
4
5 \LetLtxMacro\sidewaysfigure\figure
```

---

```

6 \let\endsidewaysfigure\endfigure
7
8 \newenvironment*{sideways}{\}{\}
9 \newenvironment*{turn}[1]{\}{\}
10 \newenvironment*{rotate}[1]{\}{\}
11 \NewDocumentCommand{\turnbox}{m +m}{#2}
12 \let\rotcaption\caption
13 \let\@makerotcaption\@makecaption

```

---

File 127 **lwarp-rotfloat.sty**

§ 206     Package **rotfloat**

Pkg    rotfloat    rotfloat is emulated during HTML output, and the rotfloat package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{rotfloat}
2
3 \RequirePackage{float}

```

\newfloat    {<1: type>} {<2: placement>} {<3: ext>} [<4: within>]

Emulates the \newfloat command from the float package. Sideways floats are \let to the same as regular floats.

“placement” is ignored.

```

4 \RenewDocumentCommand{\newfloat}{m m m o}{%
5 \IfValueTF{#4}
6 {
7   \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}
8 }
9 {
10  \DeclareFloatingEnvironment[fileext=#3]{#1}
11  \DeclareFloatingEnvironment[fileext=#3]{sideways#1}
12 }
13 \csletcs{sideways#1}{#1}
14 \csletcs{endsideways#1}{end#1}

```

newfloat package automatically creates the \listof command for new floats, but float does not, so remove \listof here in case it is manually created later.

```

15 \cslet{listof#1s}\relax
16 \cslet{listof#1es}\relax
17 }

```

---

File 128 **lwarp-savetrees.sty**

§ 207     Package **savetrees**

Pkg    savetrees    Emulated.

**for HTML output:**    Discard all options for lwarp-savetrees:

```
1 \LWR@ProvidesPackageDrop{savetrees}
```

---

File 129 **lwarp-scalefnt.sty**

§ 208     Package **scalefnt**

Pkg    scalefnt    scalefnt is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{scalefnt}

```
2 \DeclareRobustCommand\scalefont[1]{}

```

---

File 130 **lwarp-sectsty.sty**

§ 209     Package **sectsty**

Pkg    sectsty    sectsty is emulated.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{sectsty}

```
2 \newcommand*{\partfont}            [1] {}
3 \newcommand*{\partnumberfont}    [1] {}
4 \newcommand*{\parttitlefont}    [1] {}
5 \newcommand*{\chapterfont}       [1] {}
6 \newcommand*{\chapternumberfont} [1] {}
7 \newcommand*{\chaptertitlefont} [1] {}
8 \newcommand*{\sectionfont}       [1] {}
9 \newcommand*{\subsectionfont}    [1] {}
10 \newcommand*{\subsubsectionfont} [1] {}
11 \newcommand*{\paragraphfont}    [1] {}
12 \newcommand*{\subparagraphfont} [1] {}
13 \newcommand*{\minisecfont}      [1] {}

```

```

14 \newcommand*{\allsectionsfont}[1] {}
15 \newcommand{\nohang}{}

```

\sectionrule is only to be used in \*font commands, thus it is ignored.

```

16 \newcommand*{\sectionrule}[5] {}
17
18 \def\ulemheading#1#2{}

```

---

File 131 **lwarp-setspace.sty**

§ 210 Package **setspace**

Pkg setspace setspace is not used during HTML conversion.

Discard all options for lwarp-setspace:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{setspace}
2
3 \newcommand*{\setstretch}[1] {}
4 \newcommand*{\SetSinglespace}[1] {}
5 \newcommand*{\singlespacing}{}
6 \newcommand*{\onehalfspacing}{}
7 \newcommand*{\doublespacing}{}
8
9 \newenvironment*{singlespace}
10 {
11 \LWR@forcenewpage
12 \BlockClass{singlespace}
13 }
14 {\endBlockClass}
15
16 \newenvironment*{singlespace*}
17 {
18 \LWR@forcenewpage
19 \BlockClass{singlespace}
20 }
21 {\endBlockClass}
22
23 \newenvironment*{spacing}[1] {
24
25 }{
26
27 }
28
29 \newenvironment*{onehalfspace}
30 {

```

---

```

31 \LWR@forcenewpage
32 \BlockClass{onehalfspace}
33 }
34 {\endBlockClass}
35
36 \newenvironment*{doublespace}
37 {
38 \LWR@forcenewpage
39 \BlockClass{doublespace}
40 }
41 {\endBlockClass}

```

---

File 132 **lwarp-shadow.sty**

§ 211     Package **shadow**

Pkg   shadow   shadow is emulated.

for HTML output:   Discard all options for lwarp-shadow:

```

1 \LWR@ProvidesPackageDrop{shadow}

2 \newdimen\sbboxsep
3 \newdimen\sbboxrule
4 \newdimen\sdim
5
6 \newcommand{\shabox}[1]{%
7 \InlineClass{shabox}{#1}%
8 }

```

---

File 133 **lwarp-showidx.sty**

§ 212     Package **showidx**

Pkg   showidx   showidx is ignored.

for HTML output:   Discard all options for lwarp-showidx:

```

1 \LWR@ProvidesPackageDrop{showidx}

```

---

File 134 **lwarp-showkeys.sty**

§ 213    Package **showkeys**

Pkg    showkeys    showkeys is ignored.

for HTML output:    Discard all options for lwarp-showkeys:

```
1 \LWR@ProvidesPackageDrop{showkeys}

2 \NewDocumentCommand{\showkeys}{s}{}

```

---

File 135 **lwarp-sidecap.sty**

§ 214    Package **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}{}{}

```

---

File 136 **lwarp-sidenotes.sty**

§ 215     Package **sidenotes**

*(Based on original code by ANDY THOMAS, OLIVER SCHEBAUM.)*

Pkg    sidenotes    Patched for lwarp.

for HTML output:    Load the original package:

```
1 \LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with lwarp:

Stop paragraph handling while creating the caption:

```

2 \RenewDocumentCommand \sidecaption {s o o m}
3 {
4   \LWR@stoppars
5   \captionsetup{style=sidecaption}
6   \IfBooleanTF{#1}
7   { % starred
8     \IfNoValueOrEmptyTF{#2}
9     {\marginnote{\caption*{#4}}}
10    {\marginnote{\caption*{#4}}[#2]}
11  }
12  { % unstarred
13    \IfNoValueOrEmptyTF{#2}
14    {\def\@sidenotes@sidecaption@tof{#4}}
15    {\def\@sidenotes@sidecaption@tof{#2}}
16    \IfNoValueOrEmptyTF{#3}
17    {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}}
18    {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}[#3]}
19  }
20  \LWR@startpars
21 }

```

Borrowed from the lwarp version of keyfloat:

```
22 \NewDocumentEnvironment{KFLT sidenotes@marginfloat}{0}{-1.2ex} m}
```



```

23 {% start
24 \LWR@BlockClassWP{float:right; width:2in; margin:10pt}{\marginblock}%
25 \captionsetup{type=#2}%
26 }
27 {%
28 \endLWR@BlockClassWP%
29 }
30
31 \RenewDocumentEnvironment{marginfigure}{o}
32   {\begin{KFLTsidenotes@marginfloat}{figure}}
33   {\end{KFLTsidenotes@marginfloat}}
34
35 \RenewDocumentEnvironment{margintable}{o}
36   {\begin{KFLTsidenotes@marginfloat}{table}}
37   {\end{KFLTsidenotes@marginfloat}}

```

The following were changed by sidenotes, and now are reset back to their lwarp-supported originals:

Restoring the definition from the  $\text{\LaTeX}$  2<sub>ε</sub> `article.cls` source:

```

38 \renewenvironment{figure*}
39     {\@dblfloat{figure}}
40     {\end@dblfloat}
41
42 \renewenvironment{table*}
43     {\@dblfloat{table}}
44     {\end@dblfloat}


```

---

File 137 **lwarp-siunitx.sty**

§ 216    Package **siunitx**

Pkg    siunitx    siunitx is patched for use by lwarp.

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

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

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

**for HTML output:**    1 \LWR@ProvidesPackagePass{siunitx}

2 \AtBeginDocument{% in case textcomp was not loaded

```

3 \DeclareSIUnit\bohr{\textit{a}\textsubscript{0}}
4 \DeclareSIUnit\clight{\textit{c}\textsubscript{0}}
5 \DeclareSIUnit\elementarycharge{\textit{e}}
6 \DeclareSIUnit\electronmass{\textit{m}\textsubscript{e}}
7 \DeclareSIUnit\hartree{\textit{E}\textsubscript{h}}
8}% AtBeginDocument

```

The following is executed by siunitx at the end of `\document`. `\@ensuredmath` is not supported inside an `\hbox`, so it must temporarily be restored to its original.

```

9 \ExplSyntaxOn
10 \cs_undefine:N \__siunitx_set_math_fam:n
11 \cs_new_protected:Npn \__siunitx_set_math_fam:n #1 {
12   \LWR@traceinfo{sunitx set math fam}
13   \int_new:c { c__siunitx_math #1 _int }
14   \group_begin:
15     \LetLtxMacro\@ensuredmath\LWR@origensuredmath
16     \hbox_set:Nn \l__siunitx_tmp_box
17       {
18         \ensuremath
19           {
20             \use:c { math #1 }
21             {
22               \int_gset:cn { c__siunitx_math #1 _int } { \fam }
23             }
24           }
25       }
26   \group_end:
27   \LWR@traceinfo{sunitx set math fam: done}
28 }
29
30 \cs_undefine:N \__siunitx_combined_output:n
31 \cs_new_protected:Npn \__siunitx_combined_output:n #1 {
32   \group_begin:
33   \LetLtxMacro\@ensuredmath\LWR@origensuredmath
34   \bool_if:NTF \l__siunitx_number_parse_bool
35     {
36       \tl_clear:N \l__siunitx_number_out_tl
37       \bool_set_false:N \l__siunitx_number_compound_bool
38       \__siunitx_number_output_parse:n {#1}
39     }
40     {
41       \__siunitx_unit_output_pre_print:
42       \__siunitx_print:nn { number } { \ensuremath {#1} }
43       \__siunitx_unit_output_print:
44     }
45   \group_end:
46 }
47

```

---

```
48 \ExplSyntaxOff
```

---

File 138 **lwarp-soul.sty**

§ 217     Package **soul**

*(Based on original code by MELCHIOR FRANZ.)*

Pkg    soul    Emulated.

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{soul}
2 \RequirePackage{xcolor}% for \convertcolorspec
```

Storage for the colors to use:

```
3 \newcommand*{\LWR@soululcolor}{}
4
5 \newcommand*{\LWR@soulstcolor}{}
6
7 % \definecolor{\LWR@soulhlcolordefault}{HTML}{F8E800}
8 % \newcommand*{\LWR@soulhlcolor}{\LWR@soulhlcolordefault}
9 \newcommand*{\LWR@soulhlcolor}{}

```

`\so`     $\{\langle text \rangle\}$

Basic markup with css:

```
10 \newcommand{\so}[1]{%
11 \LWR@HTMLtextstyle{letter-spacing:.2ex}{letterspacing}{#1}%
12 }
```

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

```
13 \newcommand{\caps}[1]{%
14 \LWR@HTMLtextstyle%
15     {font-variant:small-caps;letter-spacing:.1ex}%
16     {capsspacing}{#1}%
17 }
```

`\LWR@soulcolor`     $\{\langle text \rangle\} \{\langle color \rangle\} \{\langle class \rangle\} \{\langle colorstyle \rangle\} \{\langle FormatWPstyle \rangle\}$

Add colors if not empty:

```
18 \newcommand{\LWR@soulcolor}[5]{%
19 \ifcsempy{#2}%
20 {\LWR@HTMLtextstyle{#5}{#3}{#1}}%
21 }
```

```

22 \convertcolorspec{named}{\csuse{#2}}{HTML}\LWR@tempcolor%
23 \LWR@HTMLtextstyle{#5;#4:\#\LWR@tempcolor}{#3}{#1}%
24 }%
25 }

26 \newcommand{\ul}[1]{%
27 \LWR@soulcolor{#1}{\LWR@soululcolor}{uline}{text-decoration-color}%
28 {text-decoration:underline;text-decoration-skip;}%
29 }
30
31 \newcommand{\st}[1]{
32 \LWR@soulcolor{#1}{\LWR@soulstcolor}{sout}{text-decoration-color}%
33 {text-decoration:line-through}%
34 }
35
36 \newcommand{\hl}[1]{
37 \LWR@soulcolor{#1}{\LWR@soulhlcolor}{highlight}{background-color}%
38 {background:\#F8E800}
39 }

```

Nullified:

```

40 \newcommand*{\soulaccent}[1]{}
41 \newcommand*{\soulregister}[2]{}
42 \newcommand{\sloppyword}[1]{#1}
43 \newcommand*{\sodef}[5]{\DeclareRobustCommand*#1[1]{\so{##1}}}
44 \newcommand*{\resetso}{}
45 \newcommand*{\capsdef}[5]{}
46 \newcommand*{\capsreset}{}
47 \newcommand*{\capssave}[1]{}
48 \newcommand*{\capsselect}[1]{}
49 \newcommand*{\setul}[2]{}
50 \newcommand*{\resetul}{}
51 \newcommand*{\setuldepth}[1]{}
52 \newcommand*{\setuloverlap}[1]{}

```

Set colors:

```

53 \newcommand*{\setulcolor}[1]{\renewcommand{\LWR@soululcolor}{#1}}
54 \newcommand*{\setstcolor}[1]{\renewcommand{\LWR@soulstcolor}{#1}}
55 \newcommand*{\sethlcolor}[1]{\renewcommand{\LWR@soulhlcolor}{#1}}

```

Long versions of the user-level macros:

```

56 \let\textso\so
57 \let\textul\ul
58 \let\texthl\hl
59 \let\textcaps\caps

```

---

File 139 **lwarp-stabular.sty**

§ 218 Package **stabular**

Pkg stabular stabular is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{stabular}

Env stabular [ $\langle vpos \rangle$ ] [ $\langle colspec \rangle$ ]

```
2 \newenvironment{stabular}[2][c]
3 {
4 \renewcommand{\noalign}[1]{
5 \begin{tabular}[#1]{#2}
6 }
7 {\end{tabular}}
```

Env stabular [ $\langle width \rangle$ ] [ $\langle vpos \rangle$ ] [ $\langle colspec \rangle$ ]

```
8 \NewDocumentEnvironment{stabular*}{m o m}
9 {
10 \renewcommand{\noalign}[1]{
11 \begin{tabular}[#2]{#3}
12 }
13 {\end{tabular}}
```


---

File 140 **lwarp-subfig.sty**

§ 219 Package **subfig**

*(Based on original code by STEVEN DOUGLAS COCHRAN.)*

Pkg subfig subfig is supported and patched by lwarp.

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

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

for HTML output:

Accept all options for lwarp-subfig:

```
1 \LWR@ProvidesPackagePass{subfig}
```

```
\sf@@@subfloat {<1 type>} [<2 lof entry>] [<3 caption>] {<4 contents>}
```

The outer minipage allows side-by-side subfloats with \hfill between.

```
2 \long\def\sf@@@subfloat#1[#2][#3]#4{%
3 \begin{minipage}{\linewidth}% new
4 \LWR@stoppars% new
5 \ifundefined{FBsc@max}{}%
6 {\FB@readaux{\let\FBsuboheight\relax}}%
7 \@tempcnta=\@ne
8 \if@minipage
9 \@tempcnta=\z@
10 \else\ifdim \lastskip=\z@ \else
11 \@tempcnta=\tw@
12 \fi\fi
13 \ifmaincaptiontop
14 \sf@top=\sf@nearskip
15 \sf@bottom=\sf@farskip
16 \else
17 \sf@top=\sf@farskip
18 \sf@bottom=\sf@nearskip
19 \fi
20 \leavevmode
21 \setbox\@tempboxa \hbox{#4}%
22 \@tempdima=\wd\@tempboxa
23 \ifundefined{FBsc@max}{}%
24 {\global\advance\Xhsize-\wd\@tempboxa
25 \dimen@=\ht\@tempboxa
26 \advance\dimen@\dp\@tempboxa
27 \ifdim\dimen@>\FBso@max
28 \global\FBso@max\dimen@
29 \fi}%
30 \vtop\bgroup
31 \vbox\bgroup
32 \ifcase\@tempcnta
33 \@minipagefalse
34 \or
35 \vskip\sf@top
36 \or
37 \ifdim \lastskip=\z@ \else
38 \@tempskipb\sf@top\relax\@xaddvskip
39 \fi
40 \fi
41 \sf@ifpositiontop{%
42 \ifx \@empty#3\relax \else
43 \sf@subcaption{#1}{#2}{#3}%
```

```

44         \vskip\sf@capskip
45         \vskip\sf@captopadj
46     \fi\egroup
47     \hrule width0pt height0pt depth0pt
48     \LWR@startpars% new
49 % \box\@tempboxa
50     #4
51     \LWR@stoppars% new
52 }{%
53     \LWR@startpars% new
54     \@ifundefined{FBsc@max}%
55     {
56 % \box\@tempboxa
57     #4
58     }%
59     {\ifx\FBsuboheight\relax
60 %         \box\@tempboxa
61     #4
62     \else
63 %         \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
64     #4
65     \fi}%
66     \LWR@stoppars% new
67     \egroup
68     \ifx \@empty#3\relax \else
69         \vskip\sf@capskip
70         \hrule width0pt height0pt depth0pt
71         \sf@subcaption{#1}{#2}{#3}%
72     \fi
73 }%
74     \vskip\sf@bottom
75 \egroup
76 \@ifundefined{FBsc@max}{}%
77     {\addtocounter{FRobj}{-1}%
78     \ifnum\c@FRobj=0\else
79         \subfloatrowsep
80     \fi}%
81 \ifmaincaptiontop\else
82     \global\advance\@nameuse{c@\@captype}\m@ne
83 \fi
84 \end{minipage}% new
85 \LWR@startpars% new
86 \endgroup\ignorespaces%
87 }%

```

\sf@subcaption {<1 type>} {<2 lof entry>} {<3 caption>}

```

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{\@captype}}{\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\@captype name}}}%
110 % %           {\@nameuse{thesub\@captype}}}%
111 % %         {#3}
112 % %       }%
113 % %       \hss
114 % %     }
115 % %   }%
116 \ifundefined{FBsc@max}%
117   {\box0}%
118   {
119 %   \parbox[t]{\the\@tempdima}{%
120 \LWR@traceinfo{sfs subcap B1}% new
121   \LWR@figcaption% new
122   \caption@make
123     {\@nameuse{sub\@captype name}}}%
124     {\@nameuse{thesub\@captype}}}%
125     {#3}
126   \LWR@figcaption% new
127 \LWR@traceinfo{sfs subcap 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@iflfl\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 }

```

\subfloat@label 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\@captive\space
209     \ifundefined{thechapter}{\@nameuse{thechapter}\space}%
210     \@nameuse{p@sub\@captive}%
211     \@nameuse{thesub\@captive}.)}}

```

Patches for \subref.

\sf@subref {\label{}}

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 }

```

\sf@@subref {\label{}}

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.

\@newsubfloat [<keys/values>] {\float name}

```

216 \LetLtxMacro\LWR@orig@newsubfloat\@newsubfloat
217
218 \def\@newsubfloat[#1]#2{%
219 \LWR@orig@newsubfloat[#1]{#2}%
220 \renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}
221 }

```

Pre-defined for figures and tables:

```

\l@subfigure  {\langle text \rangle} {\langle pagenum \rangle}

222 \renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2}}

\l@subtable   {\langle text \rangle} {\langle pagenum \rangle}

223 \renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}

```


---

File 141 **lwarp-supertabular.sty**

§ 220     Package **supertabular**

Pkg    supertabular    supertabular is emulated during HTML output.

for HTML output:    1 \LWR@ProvidesPackageDrop{supertabular}

 misplaced alignment tab character &    For \tablefirsthead, etc., enclose them as follows:

```

\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars

```

See section 8.7.

```

2 \newcommand{\LWRST@firsthead}{}
3
4 \newcommand{\tablefirsthead}[1]{%
5   \long\gdef\LWRST@firsthead{#1}%
6 }
7
8 \newcommand{\tablehead}[1]{}
9 \newcommand{\tabletail}[1]{}
10
11 \newcommand{\LWRST@lasttail}{}
12
13 \newcommand{\tablelasttail}[1]{%
14   \long\gdef\LWRST@lasttail{#1}%

```

---

```

15 }
16
17 \newcommand{\tablecaption}[2][{}]{%
18   \long\gdef\LWRST@caption{\caption[#1]{#2}}%
19 }
20
21 \let\topcaption\tablecaption
22 \let\bottomcaption\tablecaption
23
24 \global\let\LWRST@caption\relax
25
26 \newcommand*{\shrinkheight}[1]{%
27
28 \NewDocumentEnvironment{supertabular}{s o m}
29 {
30 \LWR@traceinfo{supertabular}
31 \table
32 \LWRST@caption
33 \begin{tabular}{#3}
34 \TabularMacro\ifdefvoid{\LWRST@firsthead}%
35 {\LWR@getmynexttoken}%
36 {\expandafter\LWR@getmynexttoken\LWRST@firsthead}%
37 }
38 {%
39 \ifdefvoid{\LWRST@lasttail}%
40 {}%
41 {%
42 \TabularMacro\ResumeTabular%
43 \LWRST@lasttail%
44 }%
45 \end{tabular}
46 \endtable
47 \LWR@traceinfo{supertabular done}
48 }
49
50 \NewDocumentEnvironment{mpsupertabular}{s o m}
51 {\minipage{\linewidth}\supertabular{#3}}
52 {\endsupertabular\endminipage}

```

---

File 142 **lwarp-syntonly.sty**

§ 221 Package **syntonly**

Pkg syntonly Emulated.

for HTML output: Discard all options for lwarp-syntonly:

---

```

1 \LWR@ProvidesPackageDrop{syntonly}

2 \newif\ifsyntax@
3 \syntax@false
4
5 \newcommand*{\syntaxonly}{}
6
7 \@onlypreamble\syntaxonly

```

---

File 143 **lwarp-tables.sty**

§ 222     Package **tables**

Pkg    **tables**    **tables** is emulated. \LWR@hline is used to handle the optional argument when **tables** is loaded.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{tables}

2 \newdimen\tablinesep
3 \newdimen\arraylinesep
4 \newdimen\extrarulesep

```

---

File 144 **lwarp-tabularx.sty**

§ 223     Package **tabularx**

Pkg    **tabularx**    **tabularx** is emulated by **lwarp**.

**for HTML output:**    Discard all options for **lwarp-tabularx**:

```

1 \LWR@ProvidesPackageDrop{tabularx}

2 \NewDocumentEnvironment{tabularx}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabularx*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}

```

---

File 145 **lwarp-tabulary.sty**

§ 224      Package **tabulary**

Pkg    **tabulary**    tabulary is emulated by lwarp.

**for HTML output:**    Discard all options for lwarp-tabulary.

Column types L, C, R, and J are emulated by lwarp core code.

```

1 \LWR@ProvidesPackageDrop{tabulary}

2 \NewDocumentEnvironment{tabulary}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabulary*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}
9
10 \newdimen\tymin
11 \newdimen\tymax
12 \def\tyformat{}
```

---

File 146 **lwarp-textarea.sty**

§ 225      Package **textarea**

Pkg    **textarea**    textarea is emulated.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{textarea}

```

2 \newcommand\StartFromTextArea{}
3 \newcommand\StartFromHeaderArea{}
4 \newcommand*\RestoreTextArea{}
5 \newcommand*\ExpandTextArea[1][*]{}
6 \let\NCC@restorettextarea\@empty
```


File 147 **lwarp-textcomp.sty**

§ 226     Package **textcomp**

Pkg    textcomp    textcomp is patched for use by lwarp.

§ 226.1    **Limitations**

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

 **Missing symbols**    Many textcomp symbols are not supported by many fonts. Try using more complete fonts in the CSS, but expect to see gaps in coverage.

§ 226.2    **Package loading**

for HTML output:    1 \LWR@ProvidesPackagePass{textcomp}

§ 226.3    **Remembering original definitions**

The following are restored for print when inside a lateximage:

```

2 \let\LWR@origtextdegree\textdegree
3 \let\LWR@origtextcelsius\textcelsius
4 \let\LWR@origtextohm\textohm
5 \let\LWR@origtextmu\textmu
6 \let\LWR@origtextlquill\textlquill
7 \let\LWR@origtexttrquill\texttrquill
8 \let\LWR@origtextcircledP\textcircledP
9 \let\LWR@origtexttwelveudash\texttwelveudash
10 \let\LWR@origtextthreequartersemdash\textthreequartersemdash
11 \let\LWR@origtextmho\textmho
12 \let\LWR@origtextnaira\textnaira
13 \let\LWR@origtextpeso\textpeso
14 \let\LWR@origtextrecipe\textrecipe
15 \let\LWR@origtextinterrobangdown\textinterrobangdown
16 \let\LWR@origtextpertenthousand\textpertenthousand
17 \let\LWR@origtextbaht\textbaht
18 \let\LWR@origtextdiscount\textdiscount
19 \let\LWR@origtextservicemark\textservicemark
20 \LetLtxMacro\LWR@origcapitalcedilla\capitalcedilla
21 \LetLtxMacro\LWR@origcapitalogonek\capitalogonek
22 \LetLtxMacro\LWR@origcapitalgrave\capitalgrave
23 \LetLtxMacro\LWR@origcapitalacute\capitalacute
24 \LetLtxMacro\LWR@origcapitalcircumflex\capitalcircumflex

```

```

25 \LetLtxMacro\LWR@origcapitaltilde\capitaltilde
26 \LetLtxMacro\LWR@origcapitaldieresis\capitaldieresis
27 \LetLtxMacro\LWR@origcapitalhungarumlaut\capitalhungarumlaut
28 \LetLtxMacro\LWR@origcapitalring\capitalring
29 \LetLtxMacro\LWR@origcapitalcaron\capitalcaron
30 \LetLtxMacro\LWR@origcapitalbreve\capitalbreve
31 \LetLtxMacro\LWR@origcapitalmacron\capitalmacron
32 \LetLtxMacro\LWR@origcapitaldotaccent\capitaldotaccent
33 \LetLtxMacro\LWR@origtextcircled\textcircled

```

#### § 226.4 HTML symbols

For HTML, use HTML entities or direct Unicode, depending on the engine.

`\AtBeginDocument` improves support for Lua $\TeX$  and Xe $\TeX$ .

##### § 226.4.1 pdf $\TeX$ symbols

```

34 \AtBeginDocument{
35 \ifPDFTeX
36 \renewcommand*{\textdegree}{\HTMLentity{deg}}
37 \renewcommand*{\textcelsius}{\HTMLunicode{2103}}
38 \renewcommand*{\textohm}{\HTMLunicode{2126}}
39 \renewcommand*{\textmu}{\HTMLunicode{00B5}}
40 \renewcommand*{\textlquill}{\HTMLunicode{2045}}
41 \renewcommand*{\textrquill}{\HTMLunicode{2046}}
42 \renewcommand*{\textcircledP}{\HTMLunicode{2117}}
43 \renewcommand*{\texttwelveudash}{\HTMLunicode{2014}}% emdash
44 \renewcommand*{\textthreequartersemdash}{\HTMLunicode{2014}}% emdash
45 \renewcommand*{\textmho}{\HTMLunicode{2127}}
46 \renewcommand*{\textnaira}{\HTMLunicode{20A6}}
47 \renewcommand*{\textpeso}{\HTMLunicode{20B1}}
48 \renewcommand*{\textrecipe}{\HTMLunicode{211E}}
49 \renewcommand*{\textinterrobangdown}{\HTMLunicode{2E18}}
50 \renewcommand*{\textpertenthousand}{\HTMLunicode{2031}}
51 \renewcommand*{\textbaht}{\HTMLunicode{0E3F}}
52 \renewcommand*{\textdiscount}{\}%
53 \renewcommand*{\textservicemark}{\HTMLunicode{2120}}
54 \else

```

##### § 226.4.2 Xe $\TeX$ and Lua $\TeX$ symbols

NOTE: Some of the following do not print well in the listing. Consult the .dtx or .sty file for the actual characters.

```

55 \renewcommand*{\textdegree}{°}
56 \renewcommand*{\textcelsius}{°C}

```



```

57 \renewcommand*{\textohm}{\Omega}
58 \renewcommand*{\textmu}{\mu}
59 \renewcommand*{\textlquill}{\{ }
60 \renewcommand*{\textrquill}{\} }
61 \renewcommand*{\textcircledP}{\textcircled{P}}
62 \renewcommand*{\texttwelveudash}{--}% emdash
63 \renewcommand*{\textthreequartersemdash}{--}% emdash
64 \renewcommand*{\textmho}{\Omega}
65 \renewcommand*{\textnaira}{\N}
66 \renewcommand*{\textpeso}{\P}
67 \renewcommand*{\textrecipe}{\R}
68 \renewcommand*{\textinterrobangdown}{\textinterrobangdown}
69 \renewcommand*{\textpertenthousand}{\textpertenthousand}
70 \renewcommand*{\textbaht}{\B}
71 \renewcommand*{\textdiscount}{\textdiscount}
72 \renewcommand*{\textservice}{\textservice}
73 \fi

```

## § 226.5 HTML dicritics

For HTML, Unicode diacritical marks are used:

```

74 \renewcommand*{\capitalcedilla}[1]{\text{\HTMLUnicode{0327}}#1}
75 \renewcommand*{\capitalogonek}[1]{\text{\HTMLUnicode{0328}}#1}
76 \renewcommand*{\capitalgrave}[1]{\text{\HTMLUnicode{0300}}#1}
77 \renewcommand*{\capitalacute}[1]{\text{\HTMLUnicode{0301}}#1}
78 \renewcommand*{\capitalcircumflex}[1]{\text{\HTMLUnicode{0302}}#1}
79 \renewcommand*{\capitaltilde}[1]{\text{\HTMLUnicode{0303}}#1}
80 \renewcommand*{\capitaldieresis}[1]{\text{\HTMLUnicode{0308}}#1}
81 \renewcommand*{\capitalhungarumlaut}[1]{\text{\HTMLUnicode{030B}}#1}
82 \renewcommand*{\capitalring}[1]{\text{\HTMLUnicode{030A}}#1}
83 \renewcommand*{\capitalcaron}[1]{\text{\HTMLUnicode{030C}}#1}
84 \renewcommand*{\capitalbreve}[1]{\text{\HTMLUnicode{0306}}#1}
85 \renewcommand*{\capitalmacron}[1]{\text{\HTMLUnicode{0304}}#1}
86 \renewcommand*{\capitaldotaccent}[1]{\text{\HTMLUnicode{0307}}#1}

```

`\textcircled` becomes a span with a rounded border:

```

87 \renewcommand*{\textcircled}[1]{\text{\HTMLUnicode{0307}}#1}
88 \InlineClass[border: 1px solid \LWR@currenttextcolor]{\textcircled}{#1}%
89 }
90 }% AtBeginDocument

```

## § 226.6 Inside a lateximage

When a lateximage is begun:

```

91 \appto{\LWR@restoreorigformatting}{%
92 \let\textdegree\LWR@origtextdegree%
93 \let\textcelsius\LWR@origtextcelsius%
94 \let\textohm\LWR@origtextohm%
95 \let\textmu\LWR@origtextmu%
96 \let\textlquill\LWR@origtextlquill%
97 \let\texttrquill\LWR@origtexttrquill%
98 \let\textcircledP\LWR@origtextcircledP%
99 \let\texttwelveudash\LWR@origtexttwelveudash%
100 \let\textthreequartersemdash\LWR@origtextthreequartersemdash%
101 \let\textmho\LWR@origtextmho%
102 \let\textnaira\LWR@origtextnaira%
103 \let\textpeso\LWR@origtextpeso%
104 \let\textrecipe\LWR@origtextrecipe%
105 \let\textinterrobangdown\LWR@origtextinterrobangdown%
106 \let\textpertenthousand\LWR@origtextpertenthousand%
107 \let\textbaht\LWR@origtextbaht%
108 \let\textdiscount\LWR@origtextdiscount%
109 \let\textservicemark\LWR@origtextservicemark%
110 \LetLtxMacro\capitalcedilla\LWR@origcapitalcedilla%
111 \LetLtxMacro\capitalogonek\LWR@origcapitalogonek%
112 \LetLtxMacro\capitalgrave\LWR@origcapitalgrave%
113 \LetLtxMacro\capitalacute\LWR@origcapitalacute%
114 \LetLtxMacro\capitalcircumflex\LWR@origcapitalcircumflex%
115 \LetLtxMacro\capitaltilde\LWR@origcapitaltilde%
116 \LetLtxMacro\capitaldieresis\LWR@origcapitaldieresis%
117 \LetLtxMacro\capitalhungarumlaut\LWR@origcapitalhungarumlaut%
118 \LetLtxMacro\capitalring\LWR@origcapitalring%
119 \LetLtxMacro\capitalcaron\LWR@origcapitalcaron%
120 \LetLtxMacro\capitalbreve\LWR@origcapitalbreve%
121 \LetLtxMacro\capitalmacron\LWR@origcapitalmacron%
122 \LetLtxMacro\capitaldotaccent\LWR@origcapitaldotaccent%
123 \LetLtxMacro\textcircled\LWR@origtextcircled%
124 }
```

---

File 148 **lwarp-textpos.sty**

## § 227 Package **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]{\}

```

---

File 149 **lwarp-theorem.sty**

§ 228      Package **theorem**

*(Based on original code by FRANK MITTELBACH.)*

Pkg    theorem    theorem is patched for use by lwarp.

---

Table 14: Theorem package — 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}

§ 228.1    **Remembering the theorem style**

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```

3 \gdef\theoremstyle#1{%
4   \ifundefined{th@#1}{\@warning

```

```

5      {Unknown theoremstyle '#1'. Using 'plain'}%
6      \theoremstyle{plain}%
7      \renewcommand{\LWR@newtheoremstyle}{plain}% new
8      }%
9      {%
10     \theoremstyle{#1}%
11     \renewcommand{\LWR@newtheoremstyle}{#1}% new
12     }%
13     \begingroup
14     \csname th@the\theoremstyle \endcsname
15     \endgroup}

```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

16 \gdef\xnthm#1#2[#3]{%
17   \expandafter\@ifdefinable\csname #1\endcsname
18   {%
19     \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
20     \@definecounter{#1}\@newctr{#1}[#3]%
21     \expandafter\xdef\csname the#1\endcsname
22     {\expandafter \noexpand \csname the#3\endcsname
23      \@thmcountersep \@thmcounter{#1}}}%
24     \def\@tempa{\global\@namedef{#1}}%
25     \expandafter \@tempa \expandafter{%
26       \csname th@the \theoremstyle
27       \expandafter \endcsname \the \theorem@bodyfont
28       \@thm{#1}{#2}}}%
29     \global \expandafter \let \csname end#1\endcsname \@endtheorem
30     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{\LWR@thmstyle#1}}}% new
31   }}
32
33 \gdef\ynthm#1#2{%
34   \expandafter\@ifdefinable\csname #1\endcsname
35   {
36     \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
37     \@definecounter{#1}%
38     \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
39     \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
40     \expandafter{\csname th@the \theoremstyle \expandafter
41       \endcsname \the\theorem@bodyfont \@thm{#1}{#2}}}%
42     \global \expandafter \let \csname end#1\endcsname \@endtheorem
43     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{\LWR@thmstyle#1}}}% new
44   }}
45
46 \gdef\othm#1[#2]#3{%
47   \expandafter\ifx\csname c@#2\endcsname\relax
48     \@nocounterr{#2}%
49   \else

```

```

50 \expandafter\@ifdefinable\csname #1\endcsname
51 {
52   \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
53   \expandafter \xdef \csname the#1\endcsname
54     {\expandafter \noexpand \csname the#2\endcsname}%
55   \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
56     \expandafter{\csname th@\the \theoremstyle \expandafter
57       \endcsname \the\theorembodyfont \@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}

```

## § 228.2 CSS patches

The following are patched for css.

These were in individual files thp.sty for plain, thmb.sty for margin break, etc. They are gathered together here.

Each theorem is encased in a BlockClass environment of class theorembody<style>.

Each header is encased in an \InlineClass of class theoremheader.

```

62 \gdef\th@plain{%\normalfont\itshape
63   \def\@begintheorem##1##2{%
64     \LWR@forcenewpage% new
65     \BlockClass{theorembody\LWR@thisthmstyle}% new
66     \item[\hskip\labelsep
67       \InlineClass{theoremheader}{##1\ ##2}
68     ]}%
69 \def\@opargbegintheorem##1##2##3{%
70   \LWR@forcenewpage% new
71   \BlockClass{theorembody\LWR@thisthmstyle}% new
72   \item[\hskip\labelsep
73     \InlineClass{theoremheader}{##1\ ##2\ (##3)}
74   ]}
75 }
76
77 \gdef\th@break{%\normalfont\slshape
78   \def\@begintheorem##1##2{%
79     \LWR@forcenewpage% new
80     \BlockClass{theorembody\LWR@thisthmstyle}% new
81     \item[\hskip\labelsep
82       \InlineClass{theoremheader}{##1\ ##2}\newline%
83     ]}%
84 \def\@opargbegintheorem##1##2##3{%
85   \LWR@forcenewpage% new
86   \BlockClass{theorembody\LWR@thisthmstyle}% new
87   \item[\hskip\labelsep

```

```

88      \InlineClass{theoremheader}{##1\ ##2\ (##3)}\newline
89    ]}
90 }
91
92 \gdef\th@marginbreak{%\normalfont\slshape
93   \def\@begintheorem##1##2{
94     \LWR@forcenewpage% new
95     \BlockClass{theorembody\LWR@thisthmstyle}% new
96     \item[\hskip\labelsep %
97       \InlineClass{theoremheader}{##2 \quad ##1}\newline
98     ]}%
99 \def\@opargbegintheorem##1##2##3{%
100 \LWR@forcenewpage% new
101   \BlockClass{theorembody\LWR@thisthmstyle}% new
102   \item[\hskip\labelsep %
103     \InlineClass{theoremheader}{##2 \quad ##1\ %
104       (##3)}\newline
105   ]}
106 }
107
108 \gdef\th@changebreak{%\normalfont\slshape
109   \def\@begintheorem##1##2{
110     \LWR@forcenewpage% new
111     \BlockClass{theorembody\LWR@thisthmstyle}% new
112     \item[\hskip\labelsep
113       \InlineClass{theoremheader}{##2\ ##1}\newline
114     ]}%
115 \def\@opargbegintheorem##1##2##3{%
116 \LWR@forcenewpage% new
117   \BlockClass{theorembody\LWR@thisthmstyle}% new
118   \item[\hskip\labelsep
119     \InlineClass{theoremheader}{ ##2\ ##1\ %
120       (##3)}\newline
121   ]}
122 }
123
124 \gdef\th@change{%\normalfont\slshape
125   \def\@begintheorem##1##2{
126     \LWR@forcenewpage% new
127     \BlockClass{theorembody\LWR@thisthmstyle}% new
128     \item[\hskip\labelsep
129       \InlineClass{theoremheader}{##2\ ##1}
130     ]}%
131 \def\@opargbegintheorem##1##2##3{%
132 \LWR@forcenewpage% new
133   \BlockClass{theorembody\LWR@thisthmstyle}% new
134   \item[\hskip\labelsep
135     \InlineClass{theoremheader}{##2\ ##1\ (##3)}
136   ]}
137 }

```

```

138
139 \gdef\th@margin{%\normalfont\slshape
140   \def\@begintheorem##1##2{
141     \LWR@forcenewpage% new
142     \BlockClass{theorembody\LWR@thisthmstyle}% new
143     \item[\hskip\labelsep
144       \InlineClass{theoremheader}{##2 \quad ##1}
145     ]}%
146 \def\@opargbegintheorem##1##2##3{%
147   \LWR@forcenewpage% new
148   \BlockClass{theorembody\LWR@thisthmstyle}% new
149   \item[\hskip\labelsep
150     \InlineClass{theoremheader}{##2 \quad ##1\ (#3)}
151   ]}
152 }

```

Patched for css:

```

153 \gdef\@endtheorem{\endBlockClass\endtrivlist}

```

---

File 150 **lwarp-threeparttable.sty**

§ 229     Package **threeparttable**

Pkg    **threeparttable**    threeparttable is emulated during HTML output, and the threeparttable package is ignored.

Table note are contained inside a css <div> of class tnotes. If enumitem is used, the note item labels are also individually highlighted with an additional css <span> of class tnoteitemheader, otherwise they are plain text.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{threeparttable}

\LWR@printtablenote    {\text}

Prints the table note item header inside a css class of tnoteitemheader.

```

2 \newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}

```

Env    **threeparttable**    [*alignment*] To emulate threeparttable:

```

3 \newenvironment*{threeparttable}[1][b]{}{}

```

Env    **tablenotes**    [*options*]

```

4 \newenvironment*{tablenotes}[1]{}

```

```

5 {%
6 \LWR@forcenewpage
7 \BlockClass{tnotes}%
8 \ltx@ifpackageloaded{enumitem}{%
9 \setlist[description]{format=\LWR@printtablenote}%
10 }{}%
11 \description%
12 }
13 {%
14 \enddescription%
15 \endBlockClass%
16 }

```

`\tnote`    `{\text{}}`

```
17 \newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}
```

---

File 151    **lwarp-tikz.sty**

§ 230    Package    **tikz**

Pkg    **tikz**    tikz is supported.

Accept all options for lwarp-tikz:

```
1 \LWR@ProvidesPackagePass{tikz}
```

**catcodes**    lwarp changes the catcode of \$ for its own use. The Tikz babel library temporarily changes catcodes back to normal for Tikz's use. tikz v3.0.0 introduced the babel library which handles catcode changes. For older versions, lwarp must change \$'s catcode itself.

Also see:

<https://tex.stackexchange.com/questions/16199/test-if-a-package-or-package-option-is-loaded>

**for HTML output:**    2 \begin{warpHTML}

```

3 \newboolean{LWR@tikzbabel}
4
5 \@ifpackagelater{tikz}{2013/12/20}% Test for Tikz version v3.0.0
6 {\usetikzlibrary{babel}\booltrue{LWR@tikzbabel}}
7 {\boolfalse{LWR@tikzbabel}}

```



Env `tikzpicture` `tikzpicture` environment is enclosed inside a `\lateximage`. May be used as-is, and its contents will be converted to an image.

```

8 \BeforeBeginEnvironment{tikzpicture}{%
9 \lateximage%
10 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
11 {}%
12 {\catcode'\$=3} % dollar sign is math shift
13 }
14
15 \AfterEndEnvironment{tikzpicture}{%
16 \endlateximage%
17 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
18 {}%
19 {\catcode'\$=\active}%
20 }

21 \end{warpHTML}

```

---

File 152 `lwarp-titles.sty`

## § 231 Package **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.

```

\newpagestyle {\langle name \rangle} [\langle style \rangle] {\langle commands \rangle}
2 \NewDocumentCommand{\newpagestyle}{m o m}{\}

```

```

\renewpagestyle {\langle name \rangle} [\langle style \rangle] {\langle commands \rangle}
3 \NewDocumentCommand{\renewpagestyle}{m o m}{\}

```

```

\sethead [\langle el \rangle] [\langle ec \rangle] [\langle er \rangle] {\langle ol \rangle} {\langle oc \rangle} {\langle or \rangle}
4 \NewDocumentCommand{\sethead}{o o o m m m}{\}

```

```

\setfoot [\langle el \rangle] [\langle ec \rangle] [\langle er \rangle] {\langle ol \rangle} {\langle oc \rangle} {\langle or \rangle}

```

```
5 \NewDocumentCommand{\setfoot}{o o o m m m}{}

\settitledmarks * {\langle names \rangle}

6 \NewDocumentCommand{\settitledmarks}{s m}{}

\headrule

7 \newcommand*{\headrule}{}

\footrule

8 \newcommand*{\footrule}{}

\setheadrule {\langle length \rangle}

9 \newcommand*{\setheadrule}[1]{}

\setfootrule {\langle length \rangle}

10 \newcommand*{\setfootrule}[1]{}

\makeheadrule

11 \newcommand*{\makeheadrule}{}

\makefootrule

12 \newcommand*{\makefootrule}{}

\setmarkboth {\langle code \rangle}

13 \newcommand{\setmarkboth}[1]{}

\widenhead

14 \NewDocumentCommand{\widenhead}{s o o m m}{}

\bottommarks

15 \newcommand*{\bottommarks}{}

\topmarks

16 \newcommand*{\topmarks}{}

```

`\firsttitlemarks`

17 `\newcommand*{\firsttitlemarks}{}`

`\nexttitlemarks`

18 `\newcommand*{\nexttoptitlemarks}{}`

`\outertitlemarks`

19 `\newcommand*{\outertitlemarks}{}`

`\innertitlemarks`

20 `\newcommand*{\innertitlemarks}{}`

`\newtitlemark` \* `{\langle name \rangle}`

21 `\NewDocumentCommand{\newtitlemark}{s m}{}{}`

`\pretitlemark` \* `{\langle section \rangle} {\langle text \rangle}`

22 `\NewDocumentCommand{\pretitlemark}{s m m}{}{}`

`\ifsamemark` `{\langle group \rangle} {\langle command \rangle} {\langle true \rangle} {\langle false \rangle}`

23 `\newcommand{\ifsamemark}[4]{}{}`

`\setfloathead` \* `[\langle . \rangle] [\langle . \rangle] [\langle . \rangle] {\langle . \rangle} {\langle . \rangle} {\langle . \rangle} {\langle extra \rangle} [\langle which \rangle]`

24 `\NewDocumentCommand{\setfloathead}{s o o m m m m m}{}{}`

`\setfloatfoot` \* `[\langle . \rangle] [\langle . \rangle] [\langle . \rangle] {\langle . \rangle} {\langle . \rangle} {\langle . \rangle} {\langle extra \rangle} [\langle which \rangle]`

25 `\NewDocumentCommand{\setfloatfoot}{s o o m m m m m}{}{}`

`\nextfloathead` \* `[\langle . \rangle] [\langle . \rangle] [\langle . \rangle] {\langle . \rangle} {\langle . \rangle} {\langle . \rangle} {\langle extra \rangle} [\langle which \rangle]`

26 `\NewDocumentCommand{\nextfloathead}{s o o m m m m m}{}{}`

`\nextfloatfoot` \* `[\langle . \rangle] [\langle . \rangle] [\langle . \rangle] {\langle . \rangle} {\langle . \rangle} {\langle . \rangle} {\langle extra \rangle} [\langle which \rangle]`

27 `\NewDocumentCommand{\nextfloatfoot}{s o o m m m m m}{}{}`

`\newmarkset` `{\langle markset \rangle}`

28 `\newcommand{\newmarkset}[1]{}{}`

---

```

\newextramark * {\<markset>} {\<macro-name>}
29 \NewDocumentCommand{\newextramarkset}{s m m}{}

\botextramarks {\<markset>}
30 \newcommand{\botextramarks}[1]{}

\topextramarks {\<markset>}
31 \newcommand{\topextramarks}[1]{}

\firstextramarks {\<markset>}
32 \newcommand{\firstextramarks}[1]{}

\nextramarks {\<markset>}
33 \newcommand{\nexttopextramarks}[1]{}

\outerextramarks {\<markset>}
34 \newcommand{\outerextramarks}[1]{}

\innerextramarks {\<markset>}
35 \newcommand{\innerextramarks}[1]{}

```

---

File 153 **lwarp-titleref.sty**

§ 232 Package **titleref**

Pkg titleref titleref is superceded by hyperref and nameref.

for HTML output: 1 \LWR@loadnever{titleref}{hyperref and nameref}

---

File 154 **lwarp-titlesec.sty**

§ 233 Package **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}

\titlelabel {\langle label-format \rangle}

2 \newcommand\*{\titlelabel}[1]{}

\titleformat\* {\langle command \rangle} {\langle format \rangle}

\titleformat {\langle command \rangle} [\langle shape \rangle] {\langle format \rangle} {\langle label \rangle} {\langle sep \rangle} {\langle before \rangle} [\langle after \rangle]

3 \newcommand\titleformat{%

4 \ifstar{\ttl@format@s}%

5 {\ttl@format@i}}

6 \newcommand{\ttl@format@s}[1]{}

7 \NewDocumentCommand{\ttl@format@i}{m o m m m o}{}

\chaptertitlename

8 \@ifundefined{@chapapp}{\let\@chapapp\chaptername}{}

9 \newcommand\chaptertitlename{\@chapapp}

\titlespacing \* {\langle command \rangle} {\langle left \rangle} {\langle before \rangle} {\langle after \rangle} [\langle right \rangle]

10 \NewDocumentCommand{\titlespacing}{s m m m m o}{}

\filright

11 \newcommand\*{\filright}{}

\filcenter

12 \newcommand\*{\filcenter}{}

\filleft

13 \newcommand\*{\filleft}{}

\fillast

14 \newcommand\*{\fillast}{}

\filinner

15 \newcommand\*{\filinner}{}

---

```

\filouter
16 \newcommand*\filouter{}

\wordsep
17 \newcommand\wordsep{\fontdimen\tw@\font \@plus
18 \fontdimen\thr@@\font \@minus \fontdimen4\font}

\titlename * [\langle align \rangle] {\langle material \rangle}
19 \NewDocumentCommand{\titlename}{s o m}{}

\titlerule [\langle height \rangle]
20 \providecommand*\titlerule{\@ifstar{\ttl@row}{\ttl@rule}}
21 \newcommand*\{\ttl@rule}[1] [] {}
22 \newcommand*\{\ttl@row}[2] [] {}

\iftitlemeasuring {\langle true \rangle} {\langle false \rangle}
23 \newcommand{\iftitlemeasuring}[2] {\#2}

\assignpagestyle {\langle command \rangle} {\langle pagestyle \rangle}
24 \newcommand{\assignpagestyle}[2] {\#2}

\titleclass {\langle name \rangle} [\langle startlevel \rangle] {\langle class \rangle} [\langle cmd \rangle]
25 \NewDocumentCommand{\titleclass}{m o m o}

```

---

File 155 **lwarp-titletoc.sty**

## § 234 Package **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}

```

\dottedcontents {\langle section \rangle} [\langle left \rangle] {\langle above \rangle} {\langle label \rangle} {\langle leader \rangle}
2 \NewDocumentCommand{\dottedcontents}{m o m m m}{}

```

---

```

\titlecontents * {\<section>} [\<left>] {\<above>} {\<numbered>} {\<numberless>} {\<filler>} [\<below
or begin>] [\<separator>] [\<end>]

3 \newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
4 \NewDocumentCommand{\ttl@tcstar}{m o m m m o o o}{}
5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m m o}{}

\contentsmargin [\<correction>] {\<right>}

6 \newcommand{\contentsmargin}[2] [] {}

\thecontentslabel

7 \newcommand*{\thecontentslabel}{thecontentslabel}

\thecontentspage

8 \newcommand*{\thecontentspage}{thecontentspage}

\contentslabel [\<format>] {\<space>}

9 \newcommand{\contentslabel}[2] [] {\thecontentslabel}

\contentspage [\<format>]

10 \newcommand{\contentspage}[1] [] {\thecontentspage}

\contentspush {\<text>}

11 \newcommand{\contentspush}[1] {}

\contentsuse {\<name>} {\<text>}

12 \newcommand{\contentsuse}[2] {}

\startcontents [\<name>]

13 \newcommand*{\startcontents}[1] [] {}

\stopcontents [\<name>]

14 \newcommand*{\stopcontents}[1] [] {}

\resumecontents [\<name>]

15 \newcommand*{\resumecontents}[1] [] {}

```

```

\printcontents  [⟨name⟩] {⟨prefix⟩} {⟨start⟩} {⟨code⟩}
                16 \newcommand{\printcontents}[4] [] {}

\startlist      [⟨name⟩] {⟨list⟩}
                17 \newcommand{\startlist}[2] [] {}

\stoplist       [⟨name⟩] {⟨list⟩}
                18 \newcommand{\stoplist}[2] [] {}

\resumelist     [⟨name⟩] {⟨list⟩}
                19 \newcommand{\resumelist}[2] [] {}

\printlist      [⟨name⟩] {⟨list⟩} {⟨prefix⟩} {⟨code⟩}
                20 \newcommand{\printlist}[4] [] {}


```

---

File 156 **lwarp-titling.sty**

§ 235     Package **titling**

Pkg    titling

**package support**    lwarp supports the native  $\LaTeX$  titling commands, and also supports the packages  
 **load order**    authblk and titling. If both are used, authblk should be loaded before titling.

**\published and \subtitle**    If using the titling package, additional titlepage fields for \published and \subtitle  
may be added by using \AddSubtitlePublished in the preamble. See section 53.7.

The various titling footnote restyling commands have no effect.

Pass all options to lwarp-titling:

**for HTML output:**    1 \LWR@ProvidesPackagePass{titling}

```

\@bsmtitleempty Patch \@bsmtitleempty:

                2 \let\LWR@orig@bsmtitleempty\@bsmtitleempty
                3 \renewcommand*{\@bsmtitleempty}{%
                4 \LWR@orig@bsmtitleempty%
                5 }

```

**\keepthetitle**    Patch \keepthetitle:



```

6 \let\LWR@origkeepthetitle\keepthetitle
7 \renewcommand*{\keepthetitle}{%
8 \LWR@orig@keepthetitle%
9 }

```

`\killtitle` Patch `\killtitle`:

```

10 \let\LWR@origkilltitle\killtitle
11 \renewcommand*{\killtitle}{%
12 \LWR@orig@killtitle%
13 }

```

Env `titlingpage`

```

14 \renewenvironment*{titlingpage}
15 {%

```

Start an HTML titlepage div:

```

16 \LWR@printpendingfootnotes
17 \begin{titlepage}

```

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

```

18 \LWR@maketitlesetup
19 \let\maketitle\LWR@titlingmaketitle
20 }
21 {

```

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

```

22 \end{titlepage}
23 }

```

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

```

24
25 \pretitle{}
26 \posttitle{}
27
28 \preauthor{}
29 \postauthor{}
30
31 \predate{}
32 \postdate{}

```

`\LWR@maketitlesetup` Patches `\thanks` macros.

```
33 \renewcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
34 \def\@makefnmark{\textsuperscript{\@thefnmark}}
```

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

Redefine the footnote text:

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

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

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

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

Print the text:

```
37 ##1%
38 }%
39 }
```

`\maketitle` HTML mode. Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

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

An HTML titlepage `<div>` is used for all classes.

```
41 \begin{titlepage}
```

Select which kind of footnote marks to use:

```
42 \@bsmarkseries
```

Set up special patches:

```
43 \LWR@maketitlesetup
```

Typeset the title, etc:

```
44 \@maketitle
```

Immediately generate any \thanks footnotes:

```
45 \@thanks
```

Close the HTML titlepage div:

```
46 \end{titlepage}
```

Reset the footnote counter:

```
47 \@bscontmark
48 }
```

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

```
49 \DeclareDocumentCommand{\@maketitle}{-}{%
50   \maketitlehooka
51   {
52     \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
53     \@bsprefix \@title \@bsprefix
54     \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
55   }
56   \maketitlehookb
57   {
58     \begin{BlockClass}{author}
59     \renewcommand{\and}{\and}
60     \end{BlockClass}
61     \begin{BlockClass}{oneauthor}
62     }
63     \begin{BlockClass}{oneauthor}
64     \@bsprefix \@author \@bsprefix
65     \end{BlockClass}
66     \end{BlockClass}
67   }
68   \maketitlehookc
69   {
70     \begin{BlockClass}{titledate}
71     \@bsprefix \@date \@bsprefix
72     \end{BlockClass}
73   }
74   \maketitlehookd
75 }
```

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

```
76 \renewcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
77 \@thanks
```

Select which kind of footnote marks to use:

```
78 \@bsmarkseries
```

Set up special patches:

```
79 \LWR@maketitlesetup
```

Typeset the title, etc:

```
80 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
81 \@thanks
```

Reset the footnote counter:

```
82 \@bscontmark
83 }
```

`\thanksmarkseries` `{\langle series \rangle}`

Sets the type of footnote marks used by `\thanks`, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc.

```
84 \renewcommand{\thanksmarkseries}[1]{%
85 \def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
86 }
```

Set default titlepage thanks footnote marks. See section [53.6](#).

```
87 \if@titlepage
88   \thanksmarkseries{arabic}
89 \else
90   \thanksmarkseries{fnsymbol}
91 \fi
```


---

File 157 **lwarp-tocbibind.sty**

§ 236      Package **tocbibind**

Pkg    tocbibind    tocbibind is patched for use by lwarp.

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

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

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

**Inline, with a manual TOC entry:**

A commonly-used method to introduce an index in a  $\TeX$  document:

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

**On its own HTML page, with a manual TOC entry:**

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

**Inline, with an automatic TOC entry:**

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

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

**On its own HTML page, with an automatic TOC entry:**

```

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

```

Opt tocbibind numindex  
numbered index section

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

for HTML output:

```

1 \let\simplechapterdelim\relax
2
3 \LWR@ProvidesPackagePass{tocbibind}

4 \renewenvironment{theindex}%
5   {%
6     \if@bibchapter
7       \if@donumindex
8         \chapter{\indexname}
9       \else
10        \if@dotocind
11          \chapter*{\indexname}
12          \addcontentsline{toc}{chapter}{\indexname}
13        \else
14          \chapter*{\indexname}
15        \fi
16      \fi
17    \else
18      \if@donumindex
19        \section{\indexname}
20      \else
21        \if@dotocind
22          \section*{\indexname}
23          \addcontentsline{toc}{@tocextra}{\indexname}
24        \else
25          \section*{\indexname}
26        \fi
27      \fi
28    \fi
29 \let\item\LWR@indexitem%
30 \let\subitem\LWR@indexsubitem%
31 \let\subsubitem\LWR@indexsubsubitem%
32 }

```

The following code is shared by anonchap.

```

33 \renewcommand{\simplechapter}[1][\@empty]{%
34   \def\@chapcntformat##1{%

```

---

```

35      #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
36    }%
37 }
38
39 \renewcommand{\restorechapter}{%
40 \let\@chapcntformat\@secntformat%
41 }


```

---

File 158 **lwarp-tocloft.sty**

§ 237    Package **tocloft**

Pkg    tocloft    tocloft is emulated. Most user options and macros are ignored and disabled. `\newlistof` and `\cftchapterprecis` are supported.

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

Discard all options for lwarp-tocloft:

**for HTML output:**    1 `\LWR@ProvidesPackageDrop{tocloft}`

`\tocloftpagestyle`    `{\style}`  
                           2 `\newcommand{\tocloftpagestyle}[1]{}`

`\cftmarktoc`  
                           3 `\newcommand*{\cftmarktoc}{}`

`\cfttoctitlefont`  
                           4 `\newcommand*{\cfttoctitlefont}{}`

`\cftaftertoctitle`  
                           5 `\newcommand*{\cftaftertoctitle}{}`  
                           6 `\newlength{\cftbeforetoctitleskip}`  
                           7 `\newlength{\cftaftertoctitleskip}`

`\cftmarklof`  
                           8 `\newcommand*{\cftmarklof}{}`

`\cftlofttitlefont`

9 `\newcommand*{\cftlofttitlefont}{}`

`\cftafterlofttitle`

10 `\newcommand*{\cftafterlofttitle}{}`

11 `\newlength{\cftbeforelofttitleskip}`

12 `\newlength{\cftafterlofttitleskip}`

`\cftmarklot`

13 `\newcommand*{\cftmarklot}{}`

`\cftlottitlefont`

14 `\newcommand*{\cftlottitlefont}{}`

`\cftafterlottitle`

15 `\newcommand*{\cftafterlottitle}{}`

16 `\newlength{\cftbeforelottitleskip}`

17 `\newlength{\cftafterlottitleskip}`

`\cftdot`

18 `\newcommand*{\cftdot}{.}`

`\cftdotsep`

19 `\providecommand*{\cftdotsep}{1}`

`\cftnodots`

20 `\newcommand*{\cftnodots}{5000}`

`\cftdotfill` `{\langle sep \rangle}`

21 `\providecommand{\cftdotfill}[1]{}`

`\cftsetpnumwidth` `{\langle length \rangle}`

22 `\newcommand*{\cftsetpnumwidth}[1]{}`



```

\cftsetrmarg    {\langle length \rangle}
23 \newcommand*{\cftsetrmarg}[1]{}

\cftpnumalign   {\langle alignment \rangle}
24 \newcommand*{\cftpnumalign}[1]{}

25 \newlength{\cftparskip}

26 \newlength{\cftbeforepartskip}
27 \newlength{\cftpartincent}
28 \newlength{\cftpartinwidth}
29 \newcommand*{\cftpartinfont}{}
30 \newcommand*{\cftpartinpresnum}{}
31 \newcommand*{\cftpartinftersnum}{}
32 \newcommand*{\cftpartinftersnumb}{}
33 \newcommand*{\cftpartinleader}{}
34 \newcommand*{\cftpartinsep}{}
35 \newcommand*{\cftpartinfont}{}
36 \newcommand*{\cftpartinfterpnum}{}

37 \newlength{\cftbeforechapskip}
38 \newlength{\cftchapindent}
39 \newlength{\cftchapnumwidth}
40 \newcommand*{\cftchapfont}{}
41 \newcommand*{\cftchappresnum}{}
42 \newcommand*{\cftchaptftersnum}{}
43 \newcommand*{\cftchaptftersnumb}{}
44 \newcommand*{\cftchapleader}{}
45 \newcommand*{\cftchapdotsep}{}
46 \newcommand*{\cftchappfont}{}
47 \newcommand*{\cftchapafterpnum}{}

48 \newlength{\cftbeforeseccskip}
49 \newlength{\cftsecindent}
50 \newlength{\cftsecnumwidth}
51 \newcommand*{\cftsecfont}{}
52 \newcommand*{\cftsecpresnum}{}
53 \newcommand*{\cftsecaftersnum}{}
54 \newcommand*{\cftsecaftersnumb}{}
55 \newcommand*{\cftsecleader}{}
56 \newcommand*{\cftsecdotsep}{}
57 \newcommand*{\cftsecpagefont}{}
58 \newcommand*{\cftsecafterpnum}{}

59 \newlength{\cftbeforesubsecskip}

```

```

60 \newlength{\cftsubsecindent}
61 \newlength{\cftsubsecnumwidth}
62 \newcommand*{\cftsubsecfont}{}
63 \newcommand*{\cftsubsecpresnum}{}
64 \newcommand*{\cftsubsecaftersnum}{}
65 \newcommand*{\cftsubsecaftersnumb}{}
66 \newcommand*{\cftsubsecleader}{}
67 \newcommand*{\cftsubsecdotsep}{1}
68 \newcommand*{\cftsubsecpagefont}{}
69 \newcommand*{\cftsubsecafterpnum}{}

70 \newlength{\cftbeforesubsubsecskip}
71 \newlength{\cftsubsubsecindent}
72 \newlength{\cftsubsubsecnumwidth}
73 \newcommand*{\cftsubsubsecfont}{}
74 \newcommand*{\cftsubsubsecpresnum}{}
75 \newcommand*{\cftsubsubsecaftersnum}{}
76 \newcommand*{\cftsubsubsecaftersnumb}{}
77 \newcommand*{\cftsubsubsecleader}{}
78 \newcommand*{\cftsubsubsecdotsep}{1}
79 \newcommand*{\cftsubsubsecpagefont}{}
80 \newcommand*{\cftsubsubsecafterpnum}{}

81 \newlength{\cftbeforeparaskip}
82 \newlength{\cftparaindent}
83 \newlength{\cftparanumwidth}
84 \newcommand*{\cftparafont}{}
85 \newcommand*{\cftparapresnum}{}
86 \newcommand*{\cftparaftersnum}{}
87 \newcommand*{\cftparaftersnumb}{}
88 \newcommand*{\cftparaleader}{}
89 \newcommand*{\cftparadotsep}{1}
90 \newcommand*{\cftparapagefont}{}
91 \newcommand*{\cftparaafterpnum}{}

92 \newlength{\cftbeforesubparaskip}
93 \newlength{\cftsubparaindent}
94 \newlength{\cftsubparanumwidth}
95 \newcommand*{\cftsubparafont}{}
96 \newcommand*{\cftsubparapresnum}{}
97 \newcommand*{\cftsubparaftersnum}{}
98 \newcommand*{\cftsubparaftersnumb}{}
99 \newcommand*{\cftsubparaleader}{}
100 \newcommand*{\cftsubparadotsep}{1}
101 \newcommand*{\cftsubparapagefont}{}
102 \newcommand*{\cftsubparaafterpnum}{}

103 \newlength{\cftbeforefigskip}
104 \newlength{\cftfigindent}

```

```
105 \newlength{\cftfignumwidth}
106 \newcommand*{\cftfigfont}{}
107 \newcommand*{\cftfigpresnum}{}
108 \newcommand*{\cftfigaftersnum}{}
109 \newcommand*{\cftfigaftersnumb}{}
110 \newcommand*{\cftfigleader}{}
111 \newcommand*{\cftfigdotsep}{1}
112 \newcommand*{\cftfigpagefont}{}
113 \newcommand*{\cftfigafterpnum}{}

114 \newlength{\cftbeforesubfigskip}
115 \newlength{\cftsubfigindent}
116 \newlength{\cftsubfignumwidth}
117 \newcommand*{\cftsubfigfont}{}
118 \newcommand*{\cftsubfigpresnum}{}
119 \newcommand*{\cftsubfigaftersnum}{}
120 \newcommand*{\cftsubfigaftersnumb}{}
121 \newcommand*{\cftsubfigleader}{}
122 \newcommand*{\cftsubfigdotsep}{1}
123 \newcommand*{\cftsubfigpagefont}{}
124 \newcommand*{\cftsubfigafterpnum}{}

125 \newlength{\cftbeforetabskip}
126 \newlength{\cfttabindent}
127 \newlength{\cfttabnumwidth}
128 \newcommand*{\cfttabfont}{}
129 \newcommand*{\cfttabpresnum}{}
130 \newcommand*{\cfttabaftersnum}{}
131 \newcommand*{\cfttabaftersnumb}{}
132 \newcommand*{\cfttableader}{}
133 \newcommand*{\cfttabdotsep}{1}
134 \newcommand*{\cfttabpagefont}{}
135 \newcommand*{\cfttabafterpnum}{}

136 \newlength{\cftbeforesubtabskip}
137 \newlength{\cftsubtabindent}
138 \newlength{\cftsubtabnumwidth}
139 \newcommand*{\cftsubtabfont}{}
140 \newcommand*{\cftsubtabpresnum}{}
141 \newcommand*{\cftsubtabaftersnum}{}
142 \newcommand*{\cftsubtabaftersnumb}{}
143 \newcommand*{\cftsubtableader}{}
144 \newcommand*{\cftsubtabdotsep}{1}
145 \newcommand*{\cftsubtabpagefont}{}
146 \newcommand*{\cftsubtabafterpnum}{}

147 \newcommand{\cftsetindents}[3]{}

148 \newcommand{\pagenumbersoff}[1]{}

```

```
149 \newcommand{\pagenumberson}[1]{}
```

`\newlistof` [*<within>*] [*<type>*] [*<ext>*] [*<listofname>*]  
Emulated through the `\newfloat` mechanism.

```
150 \NewDocumentCommand{\newlistof}{o m m m}
151 {%
152 \IfValueTF{#1}
153 {\newfloat{#2}{tbp}{#3}[#1]}
154 {\newfloat{#2}{tbp}{#3}}
155 \@namedef{listof#2}{\listof{#2}{#4}}
156 \@namedef{#2depth}{1}
157 \expandafter\newlength\csuse{cftbefore#2skip}
158 \expandafter\newlength\csuse{cft#2indent}
159 \expandafter\newlength\csuse{cft#2numwidth}
160 \@namedef{cft#2font}{}
161 \@namedef{cft#2presnum}{}
162 \@namedef{cft#2aftersnum}{}
163 \@namedef{cft#2aftersnumb}{}
164 \@namedef{cft#2leader}{}
165 \@namedef{cft#2dotsep}{1}
166 \@namedef{cft#2pagefont}{}
167 \@namedef{cft#2afterpnum}{}
168 }
```

`\cftchapterprecis` [*<text>*]

```
169 \newcommand{\cftchapterprecis}[1]{%
170   \cftchapterprecishere{#1}
171   \cftchapterprecistoc{#1}}
172 \newcommand{\cftchapterprecishere}[1]{%
173   \begin{quote}\textit{#1}\end{quote}}
174 \newcommand{\cftchapterprecistoc}[1]{
175   \addtocontents{toc}{%
176     {
177       \protect\begin{quote}#1\protect\end{quote}}
178   }
179 }
```


---

File 159 **lwarp-todonotes.sty**

§ 238     Package **todonotes**

Pkg    `todonotes`    `todonotes` is emulated.

todonotes and luatodonotes are only emulated as if they were loaded with the `disable` option. All functions are disabled, but source compatibility is provided with the print version.

 **xkeyvalue** todonotes and luatodonotes use xkeyval, so if the user's document uses `\presetkeys` then xkeyval will have to be loaded explicitly by the user's document.

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{todonotes}

2 \newcommand{\listoftodos}[1] [] {}
3 \newcommand{\todo}[2] [] {}
4 \newcommand{\missingfigure}[2] [] {}
5 \newcommand{\todototoc} {}
```

---

File 160 **lwarp-transparent.sty**

§ 239 Package **transparent**

*(Based on original code by HEIKO OBERDIEK.)*

Pkg transparent Emulated. `\texttransparent` works for inline objects. `\transparent` only works for `\includegraphics`.

 **Not Xe<sub>La</sub>TeX!** Note that transparent does not work with Xe<sub>La</sub>TeX.

**for HTML output:** Discard all options for lwarp-transparent:

```
1 \LWR@ProvidesPackageDrop{transparent}

2 \newcommand*{\transparent}[1]{\edef\LWR@opacity{#1}}
3
4 \newcommand*{\texttransparent}[2]{%
5 \begingroup%
6 \transparent{#1}%
7 \InlineClass[opacity: #1]{transparent}{#2}%
8 \endgroup%
9 }
```

---

File 161 **lwarp-trivfloat.sty**

§ 240 Package **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{\examplename}{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}
```

## § 240.1 **Combining `\newfloat`, `\trivfloat`, and `algorithmicx`**

**for HTML & PRINT:**

```
8 \begin{warpall}
```

**For both print and HTML output:**



When using float, trivfloat, or algorithmicx at the same time, be aware of conflicting file usage. algorithmicx uses .loa. trivfloat by default starts with .loa and goes up for additional floats, skipping .lof and .lot.



When using `\newfloat`, be sure to manually assign higher letters to the `\newfloat` files to avoid .loa used by algorithmicx, and any files used by trivfloat. Also avoid using .lof and .lot.



When using `\trivfloat`, you may force it to avoid conflicting with algorithmicx by starting trivfloat's file extensions with .lob:

---

```
\makeatletter
\setcounter{tfl@float@cnt}{1}} % start trivfloats with .lob
\makeatletter
```

---

---

```
9 \end{warpall}
```

---

File 162 **lwarp-typearea.sty**

§ 241 Package **typearea**

Pkg typearea typearea is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{typearea}

2 \newcommand*\typearea[2][\z@]{}
3 \newcommand*\recalc{typearea}{}
4 \@ifundefined{footheight}{\newlength{footheight}}{}
5 \newcommand*\areaset[3][\z@]{}

```

---

File 163 **lwarp-ulem.sty**

§ 242 Package **ulem**

*(Based on original code by DONALD ARSENEAU.)*

Pkg ulem Emulated.

for HTML output: Original lwarp definitions:

```
1 \LetLtxMacro\LWR@ulemorigemph\emph
2 \LetLtxMacro\LWR@ulemorigtextbf\textbf

```

Basic markup commands, using CSS:

```
3 \NewDocumentCommand{\uline}{+m}{%
4 \LWR@HTMLtextstyle%
5   {text-decoration:underline;text-decoration-skip}%
6   {uline}{#1}%
7 }
8
9 \NewDocumentCommand{\uuline}{+m}{%
10 \LWR@HTMLtextstyle%
11   {%
12     text-decoration:underline;text-decoration-skip;%
13     text-decoration-style:double%
14   }%
15   {\uuline}{#1}%
16 }

```

```

17
18 \NewDocumentCommand{\uwave}{+m}{%
19 \LWR@HTMLtextstyle%
20   {%
21     text-decoration:underline;text-decoration-skip;%
22     text-decoration-style:wavy%
23   }%
24   {uwave}{#1}%
25 }
26
27 \NewDocumentCommand{\sout}{+m}{%
28 \LWR@HTMLtextstyle%
29   {text-decoration:line-through}%
30   {sout}{#1}%
31 }
32
33 \NewDocumentCommand{\xout}{+m}{%
34 \LWR@HTMLtextstyle%
35   {text-decoration:line-through}%
36   {xout}{#1}%
37 }
38
39 \NewDocumentCommand{\dashuline}{+m}{%
40 \LWR@HTMLtextstyle%
41   {%
42     text-decoration:underline;%
43     text-decoration-skip;%
44     text-decoration-style:dashed%
45   }%
46   {dashuline}{#1}%
47 }
48
49 \NewDocumentCommand{\dotuline}{+m}{%
50 \LWR@HTMLtextstyle%
51   {%
52     text-decoration:underline;%
53     text-decoration-skip;%
54     text-decoration-style:dotted%
55   }%
56   {dotuline}{#1}%
57 }

```

Nullified parameters:

```

58 \NewDocumentCommand{\ULthickness}{-}{-}
59 \newlength{\ULdepth}

```

Nullified/emulated macros:



```

60 \NewDocumentCommand{\markoverwith}{m}{\}
61 \NewDocumentCommand{\ULon}{+m}{\uline{#1}\egroup}

```

\useunder only works with \textbf, etc, but not \bfseries, etc.

```

62 \NewDocumentCommand{\useunder}{m m m}{\%
63 \relax%
64 \ifx\relax#3\relax\else % argumentative command
65   \def#3{#1}\MakeRobust{#3}\fi
66 }

```

Triggered by package options, also available for the users:

```

67 \newcommand*{\normalem}{\LetLtxMacro\emph\LWR@ulemorigemph}
68 \newcommand*{\ULforem}{\LetLtxMacro\emph\uline}
69 \ULforem% default

```

Package options:

```

70 \DeclareOption{normalem}{\normalem}
71 \DeclareOption{ULforem}{\ULforem}
72 \DeclareOption{normalbf}{\}
73 \DeclareOption{UWforbf}{\useunder{\uwave}{\bf}{\textbf}}

```

Emulate the original package:

```

74 \LWR@ProvidesPackageDrop{ulem}

```

---

File 164 **lwarp-upref.sty**

§ 243     Package **upref**

    Pkg upref Ignored.

    for HTML output: Discard all options for lwarp-upref:

```

1 \LWR@ProvidesPackageDrop{upref}

```

---

File 165 **lwarp-verse.sty**

§ 244     Package **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 \HTMLleftmargin 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 \HTMLleftmargin is if there is a wide \flagverse in use, such as the word “Chorus”, in which case the value of \HTMLleftmargin 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 (-layout option) of the text in the HTML-tagged PDF output. For some settings of \HTMLleftmargin or \HTMLleftskip the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

Env verse The verse environment will be placed inside a HTML pre.

```
2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching verse.}
```

At the beginning of the verse environment:

```
4 \AtBeginEnvironment{verse}
5 {%
```

Pkg verse The verse or memoir packages can place stanza numbers to the left with their \flagverse command. Do not allow them to go into the left margin, which would cause pdfcrop to crop the entire page further to the left:

Pkg memoir

\flagverse

Len \leftskip

---

```

6 \ifdef{\vleftskip}{%
7 \setlength{\vleftskip}{\HTMLvleftskip}
8 \setlength{\leftmargini}{\HTMLleftmargini}
9 }{}
10 \LWR@forcenewpage
11 \LWR@atbeginverbatim{verse}
12 \unskip\LWR@origvspace{-\baselineskip}
13 }

```

After the end of the verse environment, which places the pre tag at the regular left margin:

```

14 \AfterEndEnvironment{verse}{
15 \unskip\LWR@origvspace{-\baselineskip}
16 \LWR@afterendverbatim
17 }

```

Patch to place poemtitle inside an HTML span of class poemtitle:

```

18 \ifdef{\poemtitle}{
19 \DeclareDocumentCommand{\@vstypeptitle}{m}{%
20   \vspace{\beforepoemtitleskip}%
21   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
22   \vspace{\afterpoemtitleskip}%
23 }
24 }{}
25
26 \LWR@traceinfo{Finished patching verse.}
27 }% AfterEndPreamble

```

---

File 166 **lwarp-wallpaper.sty**

## § 245 Package **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}

```

---

File 167 **lwarp-wrapfig.sty**

## \$ 246 Package **wrapfig**

Pkg wrapfig wrapfig is emulated during HTML output, and the wrapfig package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{wrapfig}

2 \newcommand*{\LWR@wrapposition}{}
3
4 \newcommand*{\LWR@subwrapfigure}[2]{}%
5 \renewcommand*{\LWR@wrapposition}{}%
6 \ifthenelse{%
7   \equal{#1}{r}\OR\equal{#1}{R}\OR%
8   \equal{#1}{o}\OR\equal{#1}{O}}%
9 {}%
10 {\renewcommand*{\LWR@wrapposition}{float:right}}%
11 {\renewcommand*{\LWR@wrapposition}{float:left}}%
12 \setlength{\LWR@templengthone}{#2}%
13 \uselengthunit{PT}%
14 \LWR@BlockClassWP{%
15   width:\rndprintlength{\LWR@templengthone}; \LWR@wrapposition; %
16   margin:10pt%
17 }%
18 {%
19   width:\rndprintlength{\LWR@templengthone}; \LWR@wrapposition%
20 }%
21 {marginblock}%
22 }
23
24
25 \NewDocumentEnvironment{wrapfigure}{o m o m}
26 {%
27 \LWR@subwrapfigure{#2}{#4}%
28 \captionsetup{type=figure}%
29 }
30 {%

```

---

```

31 \endLWR@BlockClassWP%
32 }
33
34
35 \NewDocumentEnvironment{wraptable}{o m o m}
36 {%
37 \LWR@subwrapfigure{#2}{#4}%
38 \captionsetup{type=table}%
39 }
40 {%
41 \endLWR@BlockClassWP%
42 }
43
44
45 \NewDocumentEnvironment{wrapfloat}{m o m o m}
46 {%
47 \LWR@subwrapfigure{#3}{#5}%
48 \captionsetup{type=#1}%
49 }
50 {%
51 \endLWR@BlockClassWP%
52 }
53
54 \newlength{\wrapoverhang}

```

---

File 168 **lwarp-xcolor.sty**

§ 247     Package **xcolor**

Pkg    **xcolor**    xcolor is supported by lwarp.

#### § 247.1    **Limitations**

**\colorboxBlock and \fcolorboxBlock**    \colorboxBlock and \fcolorboxBlock are provided for increased HTML compatibility, and they are identical to \colorbox and \fcolorbox in print mode. In HTML mode they place their contents into a <div> instead of a <span>. These <div>s are set to display: inline-block so adjacent \colorboxBlocks appear side-by-side in HTML, although text is placed before or after each.

Print-mode definitions for \colorboxBlock and \fcolorboxBlock are created by lwarp's core if xcolor is loaded.

**background: none**    \fcolorbox and \fcolorboxBlock allow a background color of none, in which case only the frame is drawn, which can be useful for HTML.

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

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

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

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

## § 247.2 **Xcolor definitions: location and timing**

The lwarp core and its lwarp-xcolor package are tightly integrated to allow comparable results for print, HTML and print inside an HTML lateximage. This requires a number of definitions and redefinitions depending on whether each of xcolor and lateximage is being used, and whether print or HTML is being generated. Some of these actions are one-time when xcolor is loaded, and others are temporary as lateximage is used.

**When xcolor is loaded in print mode:** No special actions are taken at the time that xcolor is loaded in print mode, but see `\AtBeginDocument` below.

**When lwarp-xcolor is loaded in HTML mode:** xcolor's original definitions are saved for later restoration. `\LWR@restoreorigformatting` is appended to restore these definitions for use inside a lateximage. New HTML-mode definitions are created for `\textcolor`, `\pagecolor`, `\nopagecolor`, `\colorbox`, `\colorboxBlock`, `\fcolorbox`, `\fcolorboxBlock`, and `fcolorminipage`.

**`\AtBeginDocument` in print or HTML mode:** See Section 71. If xcolor has been loaded, the print-mode `\fcolorbox` is modified to accept a background color of none, and additional definitions are created for lwarp's new macros print-mode macros `\colorboxBlock`, `\fcolorboxBlock`, and `fcolorminipage`. The HTML versions of these macros will already have been created by lwarp-xcolor if it has been loaded.

For use inside an HTML lateximage, `\LWR@restoreorigformatting` is appended to temporarily set these functions to their print-mode versions.

**In a lateximage in HTML mode:** `\LWR@restoreorigformatting` temporarily restores the print-mode definitions of xcolor's functions. See `\LWR@restoreorigformatting` on page 357.

**`\color`:**

**Print:** Used as-is.

**HTML:** Ignored by pdf2text, and will not appear.

**HTML lateximage:** Colors will appear in a lateximage.

**`\textcolor`:**

**Print:** Used as-is.

**HTML:** Redefined by lwarp-xcolor, page 609.

**HTML lateximage:** Remembers and reuses the print version.

**\pagecolor:**

**Print:** Used as-is.

**HTML:** Ignored.

**HTML lateximage:** Colors will be picked up in a lateximage.

**\nopagecolor:**

**Print:** Used as-is.

**HTML:** Ignored.

**HTML lateximage:** Colors will be picked up in a lateximage.

**\colorbox:**

**Print:** Used as-is.

**HTML:** Redefined by lwarp-xcolor, page 610.

**HTML lateximage:** Remembers and reuses the print version.

**\colorboxBlock:**

**Print:** Becomes \colorbox.

**HTML:** Newly defined by lwarp-xcolor to use a <div>, page 610.

**HTML lateximage:** Remembers and reuses the print version \colorbox.

**\fcolorbox:**

**Print:** Modified to allow a background of none.

**LWRprint@fcolorbox** at section 71

**HTML:** Redefined by lwarp-xcolor, page 610.

**HTML lateximage:** Remembers and reuses the print version.

**\fcolorboxBlock:**

**Print:** Becomes \fcolorbox. Section 71

**HTML:** Newly defined by lwarp-xcolor to use a <div>, page 611.

**HTML lateximage:** Remembers and reuses the print version \fcolorbox.

**fcolorminipage:**

**Print:** Newly defined in the lwarp core.

**LWRprint@fcolorminipage** at section 71

**HTML:** Newly defined by lwarp-xcolor, page 612.

**HTML lateximage:** Uses the print version.

**\boxframe:**

**Print:** Used as-is.

**HTML:** Redefined by lwarp-xcolor, page 612.

**HTML lateximage:** Remembers and reuses the print version.

### § 247.3 Package loading

**for HTML output:**

```
1 \LWR@ProvidesPackagePass{xcolor}
2 \begin{warpHTML}
```

### § 247.4 Remembering and restoring original definitions

Remember the following print-mode actions to be restored when inside a lateximage environment:

```
3 \LetLtxMacro\LWRprint@textcolor\textcolor
4 \LetLtxMacro\LWRprint@pagecolor\pagecolor
5 \LetLtxMacro\LWRprint@nopagecolor\nopagecolor
6 \LetLtxMacro\LWRprint@colorbox\colorbox
7 \LetLtxMacro\LWRprint@colorboxBlock\colorbox
8 \LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox
9 \LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox
10 \LetLtxMacro\LWRorigprint@boxframe\boxframe
```

**\LWR@restoreorigformatting** Inside a lateximage the following gets restored to their print-mode actions:

```
11 \appto{\LWR@restoreorigformatting}{%
12 \LetLtxMacro\textcolor\LWRprint@textcolor%
13 \LetLtxMacro\pagecolor\LWRprint@pagecolor%
14 \LetLtxMacro\nopagecolor\LWRprint@nopagecolor%
15 \LetLtxMacro\colorbox\LWRprint@colorbox%
16 \LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
17 \LetLtxMacro\boxframe\LWRorigprint@boxframe%
18 }
```

### § 247.5 HTML color style

**\LWR@tempcolor** The color converted to HTML colorspace.

```
19 \newcommand*{\LWR@tempcolor}{}
20 \newcommand*{\LWR@tempcolortwo}{}%
```

**\LWR@colorstyle**  $\{\langle 1: styletext \rangle\} \{\langle 2: model \rangle\} \{\langle 3: color \rangle\}$

For a color style, prints the color converted to HTML colors.

```
21 \NewDocumentCommand{\LWR@colorstyle}{m m}{%
22 \begingroup%
23 \LWR@FBcancel%
```



Use the xcolor package to convert to an HTML color space:

```
24 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
```

Print the converted color:

```
25 \#\LWR@tempcolor%
26 \endgroup%
27 }
```

## § 247.6 HTML border

`\LWR@borderpadding` `{\colorstyle}` `{\color}` Prints the HTML attributes for a black border and padding. `\LWR@forceminwidth` must be used first in order to set the border width.

```
28 \newcommand*{\LWR@borderpadding}[2]{%
29 \uselengthunit{PT}%
30 border:\rndprintlength{\LWR@atleastonept} solid \LWR@colorstyle{#1}{#2} ; %
31 padding:\rndprintlength{\fboxsep}%
32 }
```

## § 247.7 High-level macros

`\color` `\color` appears in the  $\text{\LaTeX}$  PDF output, but is ignored by `pdftotext` and thus is ignored in the HTML file. Text styling by local group is not yet supported.

Each of the following macros is given a temporary name, and is `\let` to the final name once the HTML conversion starts.

`\textcolor` `[\model]` `{\color}` `{\text}`

Converted into an HTML hex color span.

```
33 \RenewDocumentCommand{\textcolor}{O{named} m m}{%
34 \begingroup%
35 \LWR@FBCancel%
36 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
37 \InlineClass[color:\LWR@colorstyle{#1}{#2}]{\textcolor}{%
38 \renewcommand*{\LWR@currenttextcolor}{\#\LWR@tempcolor}%
39 #3%
40 }%
41 \endgroup%
42 }
```

`\pagecolor` `[\model]` `{\color}`

Ignored. Use `\CSSFilename` instead.

```
43 \renewcommand*{\pagecolor}[2][named]{}
```

`\nopagecolor` Ignored.

```
44 \let\nopagecolor\relax
```

`\colorbox` [*model*] {*color*} {*text*}

Converted into an HTML hex background color `<span>`.

```
45 \RenewDocumentCommand{\colorbox}{O{named} m +m}{%
46 \begingroup%
47 \LWR@FBcancel%
48 \uselengthunit{PT}%
49 \InlineClass[%
50 background:\LWR@colorstyle{#1}{#2} ; %
51 padding:\rndprintlength{\fboxsep}%
52 ]{\colorbox}{#3}%
53 \endgroup%
54 }
```

`\colorboxBlock` [*model*] {*color*} {*text*}

Converted into an HTML hex background color `<div>`.

```
55 \NewDocumentCommand{\colorboxBlock}{O{named} m +m}{%
56 \begingroup%
57 \LWR@FBcancel%
58 \uselengthunit{PT}%
59 \begin{BlockClass}%
60 background:\LWR@colorstyle{#1}{#2} ; %
61 padding:\rndprintlength{\fboxsep}%
62 ]{\colorboxBlock}
63 #3
64 \end{BlockClass}%
65 \endgroup%
66 }
```

`\fcolorbox` [*framemodel*] {*framecolor*} [*boxmodel*] {*boxcolor*} {*text*}

Converted into a framed HTML hex background color `span`.

A background color of none creates a colored frame without a background color.

```
67 \RenewDocumentCommand{\fcolorbox}{O{named} m O{named} m +m}{%
68 \LWR@traceinfo{HTML fcolorbox #2 #4}%
69 \begingroup%
```

```

70 \LWR@FBcancel%
71 \uselengthunit{PT}%
72 \LWR@forceminwidth{\fboxrule}%
73 \ifthenelse{\equal{#4}{none}}%
74 {% no background color
75   \InlineClass[%
76     \LWR@borderpadding{#1}{#2}%
77   ]{fcolorbox}{#5}%
78 }%
79 {% yes background color
80   \InlineClass[%
81     \LWR@borderpadding{#1}{#2} ; %
82     background:\LWR@colorstyle{#3}{#4}%
83   ]{fcolorbox}{#5}%
84 }%
85 \endgroup%
86 }

```

`\fcolorboxBlock` [*(framemodel)*] [*(framecolor)*] [*(boxmodel)*] [*(boxcolor)*] [*(text)*]

Converted into a framed HTML hex background color span.

A background color of none creates a colored frame without a background color.

```

87 \NewDocumentCommand{\fcolorboxBlock}{0{named} m 0{named} m +m}{%
88 \LWR@traceinfo{HTML fcolorboxBlock #2 #4}%
89 \begingroup%
90 \LWR@FBcancel%
91 \uselengthunit{PT}%
92 \LWR@forceminwidth{\fboxrule}%
93 \ifthenelse{\equal{#4}{none}}%
94 {% no background color
95   \begin{BlockClass}[%
96     \LWR@borderpadding{#1}{#2}%
97   ]{fcolorboxBlock}
98   #5
99   \end{BlockClass}%
100 }%
101 {% yes background color
102   \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
103   \begin{BlockClass}[%
104     background:\#\LWR@tempcolortwo; %
105     \LWR@borderpadding{#1}{#2}%
106   ]{fcolorboxBlock}
107   #5
108   \end{BlockClass}%
109 }%
110 \endgroup%

```

```
111 \LWR@traceinfo{HTML fcolorboxBlock done}%
112 }
```

Creates a framed HTML <div> around its contents.

A print-output version is defined in the lwarp core: section 71

```
\LWR@subfcolorminipage  {\langle framemodel \rangle} {\langle framecolor \rangle} {\langle background tag \rangle} {\langle height \rangle}

113 \NewDocumentCommand{\LWR@subfcolorminipage}{m m m m}{%
114 \begin{BlockClass}[%
115 #3%
116 \LWR@borderpadding{#1}{#2} ; %
117 \IfValueT{#4}{height:\rndprintlength{\LWR@tempheight} ; }%
118 width:\rndprintlength{\LWR@tempwidth}%
119 ]{fcolorminipage}%
120 }

Env  fcolorminipage  [\langle 1:framemodel \rangle] {\langle 2:framecolor \rangle} [\langle 3:boxmodel \rangle] {\langle 4:boxcolor \rangle} [\langle 5:align \rangle] [\langle 6:height \rangle]
[\langle 7:inner-align \rangle] {\langle 8:width \rangle}

121 \NewDocumentEnvironment{fcolorminipage}{0{named} m 0{named} m 0{c} o o m}
122 {%
123 \LWR@FBCancel%
124 \setlength{\LWR@tempwidth}{#8}%
125 \IfValueT{#6}{\setlength{\LWR@tempheight}{#6}}%
126 \uselengthunit{PT}%
127 \LWR@forceminwidth{\fboxrule}%
128 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
129 \ifthenelse{\equal{#4}{none}}%
130 {\LWR@subfcolorminipage{#1}{#2}{#6}}%
131 {%
132   \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
133   \LWR@subfcolorminipage{#1}{#2}{background:\#\LWR@tempcolortwo\ ; }{#6}%
134 }%
135 }
136 {\end{BlockClass}}
```

```
\boxframe  {\langle width \rangle} {\langle height \rangle} {\langle depth \rangle}
```

The depth is added to the height, but the box is not decended below by the depth.  
\textcolor is honored.

```
137 \renewcommand*\boxframe}[3]{%
138 {%
139 \setlength{\LWR@tempwidth}{#1}%
```

---

```

140 \setlength{\LWR@tempheight}{#2}%
141 \addtolength{\LWR@tempheight}{#3}%
142 \uselengthunit{PT}%
143 \LWR@forceminwidth{\fboxrule}%
144 \InlineClass[%
145 display:inline-block ; %
146 border:\rndprintlength{\LWR@atleastonept} solid \LWR@currenttextcolor{} ; %
147 width:\rndprintlength{\LWR@tempwidth} ; %
148 height:\rndprintlength{\LWR@tempheight}%
149 ]{\boxframe}{}%
150 }%
151 }

152 \end{warpHTML}

```


---

File 169 **lwarp-xfrac.sty**

§ 248     Package **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*] {\langle num \rangle} [\langle sep \rangle] {\langle denom \rangle}

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. \scalebox is made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a lateximage, no adjustments are necessary.

for HTML & PRINT:    2 \begin{warpall}

\xfracHTMLfontsize    User-redefinable macro which controls the font size of the fraction.

```
3 \newcommand*{\xfracHTMLfontsize}{.6em}
```

```
4 \end{warpall}
```

for HTML output: 

```
5 \begin{warpHTML}
```

**font size** A span for a small font, used in the numerator and denominator:

```
6 \newcommand*{\LWR@htmlsmallfontstart}{%
7 \LWR@htmltagc{span style="font-size:\xfracHTMLfontsize"{}}%
8 \LWR@nestspan%
9 %
10 }
11
12 \newcommand*{\LWR@htmlsmallfontend}{%
13 \LWR@htmltagc{/span}%
14 \endLWR@nestspan%
15 }
```

**\scalebox** A nullified `\scalebox` command, to avoid introducing HTML scaling tags:

```
16 \NewDocumentCommand{\LWR@noscalebox}{m o m}{#3}
```

**instances** Instances of `xfrac` for various font choices:

Produce HTML tags for a small superscript numerator and a small (non-subscript) denominator.

Scaling is turned off so that `pdftotext` correctly reads the result.

```
17 \DeclareInstance{xfrac}{default}{text}{
18 numerator-format = {%
19 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
20 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
21 denominator-format = {%
22 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
23 \LWR@htmlsmallfontstart{#1}\,\LWR@htmlsmallfontend},
```

For `pdftotext`, do not scale the text:

```
24 scaling = false
25 }
26
27 \DeclareInstance{xfrac}{lmr}{text}{
28 numerator-format = {%
29 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
30 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
31 denominator-format = {%
32 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
33 \LWR@htmlsmallfontstart{#1}\,\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```

34 scaling = false
35 }
36
37 \DeclareInstance{xfrac}{lmss}{text}{
38 numerator-format = {%
39 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
40 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
41 denominator-format = {%
42 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
43 \LWR@htmlsmallfontstart{ }\,#1\LWR@htmlsmallfontend},

```

For pdftotext, do not scale the text:

```

44 scaling = false
45 }
46
47 \DeclareInstance{xfrac}{lmtt}{text}{
48 numerator-format = {%
49 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
50 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
51 denominator-format = {%
52 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
53 \LWR@htmlsmallfontstart{ }\,#1\LWR@htmlsmallfontend},

```

For pdftotext, do not scale the text:

```

54 scaling = false
55 }

56 \end{warpHTML}

```

---

File 170 **lwarp-xltxtra.sty**

§ 249     Package **xltxtra**

Pkg    xltxtra    xltxtra is emulated.

(Based on original code by WILL ROBERTSON, JONATHAN KEW.)

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{xltxtra}

2 \RequirePackage{realscripts}
3 \RequirePackage{metalogo}
4 \newcommand*{\TeX@logo@spacing[6]}{

```

---

```

5
6 \newcommand*{\vfrac}[2]{%
7 \textsuperscript{#1}/\textsubscript{#2}%
8 }
9
10 \newcommand\namedglyph[1]{%
11   \@tempcnta=\XeTeXglyphindex "1"\relax
12   \ifnum\@tempcnta>0
13     \XeTeXglyph\@tempcnta
14   \else
15     \xxt@namedglyph@fallback{#1}%
16   \fi}
17
18 \newcommand\xxt@namedglyph@fallback[1]{[#1]}
19
20 \DeclareDocumentCommand{\showhyphens}{m}{-}

```

---

File 171 **lwarp-xmpincl.sty**

§ 250    Package **xmpincl**

Pkg    xmpincl    Emulated.

for HTML output:    Discard all options for lwarp-xmpincl:

```

1 \LWR@ProvidesPackageDrop{xmpincl}

2 \newcommand*{\includexmp}[1]{}

```


---

File 172 **lwarp-xtab.sty**

§ 251    Package **xtab**

Pkg    xtab    xtab is emulated during HTML output.

for HTML output:    1 \LWR@ProvidesPackageDrop{xtab}

 misplaced alignment alignment tab character &

For \tablefirsthead, etc., enclose them as follows:

```

\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars

```



See section 8.7.

```

2 \newcommand{\LWRXT@firsthead}{}
3
4 \newcommand{\tablefirsthead}[1]{%
5     \long\gdef\LWRXT@firsthead{#1}%
6 }
7
8 \newcommand{\tablehead}[1]{%
9
10 \newcommand{\tablelasthead}[1]{%
11
12 \newcommand{\notablelasthead}{}
13
14 \newcommand{\tabletail}[1]{%
15
16 \newcommand{\LWRXT@lasttail}{}
17
18 \newcommand{\tablelasttail}[1]{%
19     \long\gdef\LWRXT@lasttail{#1}%
20 }
21
22 \newcommand{\tablecaption}[2][{}]{%
23     \long\gdef\LWRXT@caption{\caption[#1]{#2}}%
24 }
25
26 \let\topcaption\tablecaption
27 \let\bottomcaption\tablecaption
28
29 \global\let\LWRXT@caption\relax
30
31 \newcommand*{\shrinkheight}[1]{%
32
33 \newcommand*{\xentrystretch}[1]{%
34
35 \NewDocumentEnvironment{xtabular}{s o m}
36 {
37 \LWR@traceinfo{xtabular}
38 \table
39 \LWRXT@caption
40 \begin{tabular}{#3}
41 \TabularMacro\ifdefvoid{\LWRXT@firsthead}%
42 {\LWR@getmynexttoken}%
43 {\expandafter\LWR@getmynexttoken\LWRXT@firsthead}%
44 }
45 {%
46 \ifdefvoid{\LWRXT@lasttail}%
47 {}%
48 {%
49 \TabularMacro\ResumeTabular%
```

---

```

50 \LWRXT@lasttail%
51 }%
52 \end{tabular}
53 \endtable
54 \LWR@traceinfo{xtabular done}
55 }
56
57 \NewDocumentEnvironment{mpxtabular}{s o m}
58 {\minipage{\linewidth}\xtabular{#3}}
59 {\endxtabular\endminipage}

```

---

File 173 **lwarp-zwpage layout.sty**

§ 252 Package **zwpage layout**

Pkg zwpage layout zwpage layout is ignored.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{zwpage layout}

2 \def\noBboxes{}
3 \@onlypreamble\noBboxes
4
5 \expandafter\ifx\csname definecolor\endcsname\relax \else
6   \definecolor{cmykblack}{cmyk}{0,0,0,1}
7   \definecolor{grblack}{gray}{0}
8%   \ifzwpl@redefineblack
9%     \definecolor{black}{cmyk}{0,0,0,1}\color{black}
10%   \fi
11  \definecolor{cmykred}{cmyk}{0,1,1,0}
12  \definecolor{cmykgreen}{cmyk}{1,0,1,0}
13  \definecolor{cmykblue}{cmyk}{1,1,0,0}
14  \definecolor{rgbred}{rgb}{1,0,0}
15  \definecolor{rgbgreen}{rgb}{0,1,0}
16  \definecolor{rgbblue}{rgb}{0,0,1}
17%   \ifzwpl@redefinetcmyk
18%     \definecolor{red}{cmyk}{0,1,1,0}
19%     \definecolor{green}{cmyk}{1,0,1,0}
20%     \definecolor{blue}{cmyk}{1,1,0,0}
21%   \fi
22 \fi
23
24 \let\OverprintXeTeXExtGState\relax
25
26 \DeclareRobustCommand\SetOverprint{\ignorespaces}
27 \DeclareRobustCommand\SetKnockout{\ignorespaces}
28 \DeclareRobustCommand\textoverprint[1]{\SetOverprint#1}
29 \DeclareRobustCommand\textknockout[1]{\SetKnockout#1}

```

```
30
31 \def\SetPDFminorversion#1{}
32 \@onlypreamble\SetPDFminorversion
33
34 \newcommand*\Vcorr{}
35
36 \DeclareRobustCommand\vb[1] [] {}
37 \NewDocumentCommand{\NewOddPage}{* o}{}
38 \NewDocumentCommand{\NewEvenPage}{* o}{}
39 \def\SetOddPageMessage#{\gdef\ZW@oddwarning}
40 \def\SetEvenPageMessage#{\gdef\ZW@evenwarning}
41 \def\ZW@oddwarning{Empty page inserted}\let\ZW@evenwarning\ZW@oddwarning
42
43 \def\clap#1{#1}
44
45 \def\CropFlap{2in}
46 \def\CropSpine{1in}
47 \def\CropXSpine{1in}
48 \def\CropXtrim{.25in}
49 \def\CropYtrim{.25in}
50 \def\UserWidth{5in}
51 \def\UserLeftMargin{1in}
52 \def\UserRightMargin{1in}
53 \def\UserTopMargin{1in}
54 \def\UserBotMargin{1in}
55 \def\thePageNumber{\#\,\arabic{page}}
56 \ifXeTeX
57 \def\ifcaseZWdriver{\ifcase2}
58 \else
59 \def\ifcaseZWdriver{\ifcase1}
60 \fi
61 \DeclareRobustCommand\ZWifdriver[2]{}

```

# Change History and Index

## § 252 Change History

v0.10		Docs: Table: Float data structures. 341
General: 2016/03/08 Initial version . . .	1	Docs: Trademarks section. . . . . 424
v0.11		Docs: Troubleshooting
General: 2016/03/11 . . . . .	1	cross-references. . . . . 118
Added section: Operating-System		Test Suite: Assigned cleveref name
portability. . . . .	127	for Test Float. . . . . 1
Added section: Selecting the		Test Suite: Floatrow . . . . . 1
operating system. . . . .	78	
Test Suite: limages and index in		v0.15
README.txt . . . . .	1	General: 2016/04/06 . . . . . 1
Test Suite: MS-Windows in		Added . . . . . 467
README.txt . . . . .	1	Ampersand (&): Fixed handling
		when passed as an argument. . . 289
v0.12		Docs: Added warning icons for
\LWR@newhtmlfile: Bugfix: TOC		items needing special attention. 124
with numbered files. . . . .	243	Docs: Clarify print/HTML output. 78
General: 2016/03/14 . . . . .	1	Docs: Moved the supported
Global: Uses \p@(type) in float		functions table to the
captions. . . . .	1	introduction. . . . . 41
Test Suite: Sub-figures . . . . .	1	Files: lwarp_formal.css added. . . . 1
v0.13		Fix: steps counter . . . . . 467
\CaptionSeparator: Fix for newer		Fixed & handling. . . . . 465
babel package. . . . .	344	Test Suite: test_suite_formal.css
\LWR@LwarpStart: \up and \fup . .	259	file added. . . . . 1
General: 2016/03/24 . . . . .	1	v0.16
Removed package: subfig . . . . .	1	General: 2016/04/11 . . . . . 1
Test Suite: Ordinals, Subcaption . .	1	\titlingpage: Improved
tikzpicture: Fix dollar-redefined		print-output spacing. . . . . 267
bug for newer package. . . . .	577	Added XeLaTeX, LuaLaTeX
v0.14		support. . . . . 137
\LWR@htmlsectionfilename: Fix:		Docs: Font and UTF-8 support. . . 68
Links to home page. . . . .	212	Docs: Moved location of
General: 2016/03/31 . . . . .	1	\usepackage{lwarp}. . . . . 70
floatrow: Added. . . . .	463	Docs: Text not converting. . . . . 118
Docs: Commands for a successful		Fix: amsmath options clash . . . . 141
HTML conversion. . . . .	82	Fix: newtxmath compatibility. . . 141
Docs: Commands into a warpprint		Lwarp no longer selects fonts. 68, 137
environment. . . . .	80	Removed package: suffix . . . . . 1
Docs: Newclude limitations. . . .	105	Test Suite: Improved titlingpage. 267
Docs: Table: Cross-referencing		Test Suite: Lwarp no longer selects
data structures. . . . .	331	fonts. . . . . 1

Test Suite: Supports XeLaTeX, LuaLaTeX. . . . .	1	verse: Supports verse, memoir packages. . . . .	602
xfrac: Adjusted for the use of any font: . . . . .	614	minipage: Fix: \linewidth, \textwidth, \textheight inside a minipage. . . . .	398
v0.17		v0.19	
\LWR@htmlsectionfilename: Fix: Links when entire doc is one HTML page. . . . .	212	\HTMLFilename: Docs: Escape filename underscores. . . . .	212
General: 2016/04/14 . . . . .	1	\HomeHTMLFilename: Docs: Escape filename underscores. . . . .	212
mdframed: Added. . . . .	507	\LWR@LwarpStart: Enabled \ equal to \newline. . . . .	258
Test Suite: Fix: Print-version front-matter page numbers. . . . .	1	\LWR@doubledollar: MathJax support. . . . .	362
Test Suite: Mdfamed . . . . .	1	\LWR@filestart: lwarp_mathjax.txt loaded. . . . .	256
v0.18		\LWR@hspace: Fix: \hspace length computations. . . . .	415
\LWR@hspace: \hspace supported. . . . .	415	\LWR@minipagestartpars: Surpresses paragraph tags between minipages. . . . .	413
\LWR@includegraphicsb: Add: svgz file extension. . . . .	482	\LWR@singledollar: MathJax support. . . . .	363
em, ex, %, px dimensions preserved. . . . .	482	\LateximageFontSizeName: Add: User-adjustable math/lateximage font size. . . . .	378
Fix: \linewidth, \textwidth, \textheight inside a minipage. . . . .	482	\minipagefullwidth: Added: No width tag for the next minipage in HTML. . . . .	398
Improved HTML output linebreaks. . . . .	482	\rule: Added . . . . .	419
\LWR@myshorttoc: Reorganize \HomeHTMLFilename logic. . . . .	348	\warpHTMLonly: Added. . . . .	131
\LWR@newhtmlfile: sideroc after title, improving responsive design. . . . .	242	\warpprintonly: Replaces \rowprintedonly. . . . .	131
\LWR@requesttoc: Reorganize \HomeHTMLFilename logic. . . . .	261	\xfracHTMLfontsize: Added. . . . .	613
\LWR@subhyperref: Improved HTML output linebreaks. . . . .	339	General: 2016/06/08 . . . . .	1
\LWR@subhyperrefclass: Improved HTML output linebreaks. . . . .	339	Added optional args. . . . .	522
\LWR@subinlineimage: Surpress extra space. . . . .	340	Avoids MathJax. . . . .	362
General: 2016/05/19 . . . . .	1	cleveref: Loaded \AtEndPreamble. . . . .	394
File: lwarp.css: Improved TOC outline display. . . . .	1	CSS for table note item. . . . .	575
Files: lwarp.css and lwarp_formal.css: Improved responsive design. . . . .	1	Docs: Math options. . . . .	70
Microtype disabled during HTML generation . . . . .	137	Docs: Table: Cross-referencing data structures, updated. . . . .	331
PDF Unicode input characters. . . . .	125	File: lwarp_mathjax.txt added. . . . .	1
Test Suite: Verse package . . . . .	1	File: lwarp.css: tnoteitemheader added . . . . .	1
\lateximage: pdfcrop: --hires added. . . . .	382	Introduction: MathJax support mentioned. . . . .	40
Reorganize \HomeHTMLFilename logic. . . . .	382	MathJax support added. . . . .	364, 369, 370
Surpress extra space. . . . .	382	Options: mathsvg and mathjax . . . . .	129

Supports colored <code>\rule</code> . . . . .	608	<code>\LWR@subhtmlclass</code> :	
<code>titles: null \pagestyle</code> and		Factored code. . . . .	222
<code>\thispagestyle</code> for HTML. . .	577	<code>\SetHTMLFileNumber</code> : Add: Control	
<code>tikzpicture</code> : Adapts to tikz version. . . . .	577	file numbers. . . . .	212
<code>equation</code> : MathJax support. . . . .	367	<code>\cpagerefFor</code> : User-redefinable	
v0.20		word for page references. . . . .	394
<code>\BlockClassSingle</code> : Renamed from		<code>\dotfill</code> : Inserts an ellipsis. . . . .	412
" <code>LWR@htmldivclassline</code> ". . . . .	224	<code>\hfill</code> : Inserts a <code>\quad</code> . . . . .	411
<code>\HTMLDescription</code> : Added		<code>\hrulefill</code> : Inserts a short rule. . . . .	411
<code>\NewHTMLdescription</code> .		<code>\hyperindexref</code> : Print mode	
(Renamed in v0.30.) . . . . .	235	provided in case <code>hyperref</code> not	
<code>\HTMLFilename</code> : No longer escape		used. . . . .	357
<code>underscores</code> . . . . .	212	<code>\pageref</code> : Added. . . . .	338
<code>\HomeHTMLFilename</code> : No longer		<code>\tracinglwrap</code> : Added. . . . .	146
escape <code>underscores</code> . . . . .	212	General: 2017/02/09 . . . . .	1
<code>\InlineClass</code> : Renamed from		<code>afterpage</code> : Added. . . . .	428
" <code>inlineclass</code> ". . . . .	225	<code>alltt</code> : Added. . . . .	430
<code>\LWR@LwrapStart</code> : Fix: math cross		<code>bookmark</code> : Added. . . . .	437
<code>references</code> . . . . .	259	<code>caption</code> and <code>subcaption</code> supported. . . . .	1
<code>\LWR@closeparagraph</code> : <code>\unskip</code>		<code>cleveref</code> and referencing patches:	
extra spaces. . . . .	228	Applied <code>\AfterEndPreamble</code> . . . . .	394
No break tags in the start/end of a		<code>draftwatermark</code> : Added. . . . .	443
<code>tabular</code> . . . . .	228	<code>eso-pic</code> : Added. . . . .	447
<code>\LWR@endoffline</code> : Fix: <code>\\</code> . . . . .	413	<code>everypage</code> : Added. . . . .	448
<code>\LWR@filestart</code> : Adds meta		<code>extramarks</code> : Added. . . . .	449
<code>description</code> . . . . .	256	<code>fancyhdr</code> : Added. . . . .	453
<code>\LWR@hspace</code> : Add: Supports HTML		<code>hyperref</code> : Additional user macros. . . . .	490
thin breakable space. . . . .	415	<code>keyfloat</code> : Added. . . . .	496
<code>\LWR@htmldivclass</code> : Added		<code>letterspace</code> : User-interface	
optional style. . . . .	223	emulated. . . . .	497
<code>\LWR@htmlclass</code> : Added		<code>listings</code> : Added. . . . .	499
optional style. . . . .	222	<code>lrcaption</code> : Added. . . . .	505
<code>\LWR@htmlsectionfilename</code> :		<code>lwrap-newproject</code> : Added. . . . .	154
<code>HTMLFilename</code> : removed		<code>microtype</code> : User-interface	
additional trailing <code>'</code> , and may be		emulated. . . . .	517
empty. . . . .	212	<code>needspace</code> : Added. . . . .	524
Sections called "Index" or "index"		<code>nowidow</code> : Added. . . . .	526
have an underscore prepended to		<code>placeins</code> : Added. . . . .	541
their filenames if no prefix. . . . .	212	<code>ragged2e</code> : Added. . . . .	543
<code>\LWR@includegraphicsb</code> : Fix:		<code>setspace</code> : Improved support. . . . .	549
<code>\linewidth</code> in a floatrow. . . . .	482	<code>textpos</code> : Added. . . . .	570
Fix: Expands filename. . . . .	482	<code>titles</code> : Added. . . . .	577
<code>\LWR@longtablecaptiontag</code> :		<code>titlesec</code> : Added. . . . .	580
Fix: Pairs in captions. . . . .	316	<code>titletoc</code> : Added. . . . .	582
<code>\LWR@section</code> : Combined		<code>titling</code> : Improved compatibility. . . . .	584
higher-level sections together		<code>tocloft</code> : Added. . . . .	591
into files. . . . .	248	<code>wallpaper</code> : Added. . . . .	603
<code>\LWR@setOSWindows</code> : Auto-detects		<code>wrapfig</code> : Added. . . . .	604
operating system. . . . .	127	Added <code>@</code> , <code>&lt;</code> , <code>&gt;</code> columns. . . . .	284

Added single-expansion data arrays. . . . .	210	lwarpmk: Fix: lwarpmk limages for Windows. . . . .	195
Code factored into independent lwarpmk_html files. . . . .	424	lwarpmk: Fix: lwarpmk uses lateximages text file instead of shell script. . . . .	195
Docs: Examples for generating HTML file names. . . . .	76	Add: Errors for misplaced packages. . . . .	132
Docs: Improved index. . . . .	1	Docs: Added internet class. . . .	45
Enhanced titling support. . . . .	265	Docs: Added TeX2page, GladTeX. .	45
File: lwarpmk.css: Minor fixes for validation. . . . .	1	Docs: Installing on Windows. . . .	51
File: lwarpmk used to compile print, HTML, indexes, and lateximages. . . . .	1	File lwarpmk_tutorial.txt added. .	55
Fix: \linewidth in a floatrow. . .	465	v0.22	
Improved float caption type handling. . . . .	460	\LWR@parseDcolumn: Added tabular D column. . . . .	297
Moved sidebar and example code to test suite. . . . .	1	\LWR@parsebangcolumn: Added tabular ! column. . . . .	294
Page geometry set to 6in wide with large margins. . . . .	138	\LWR@parsetablecols: Unknown table column types become 1. Added tabular D, !, X columns. . .	298
Parallel versions of aux files for print/HTML. . . . .	1	\LWR@printmccoldata: Added tabular D, !, and X columns. . .	312
Removed reliance on make, grep, gawk. . . . .	1	General: 2017/03/02 . . . . .	1
Tabular: \unskip extra spaces. . .	284	abstract: Added. . . . .	426
Test Suite: HTML meta descriptions. . . . .	1	change page: Added. . . . .	440
Verbatim: Added. . . . .	455	dcolumn: Added. . . . .	443
verbatim: Added. . . . .	276	ftnright: Added. . . . .	475
BlockClass: Added optional style. .	224	geometry: Nullified commands. .	476
Renamed from "blockclass". . . .	224	indentfirst: Added. . . . .	495
LWR@nestspan: Fix: Minipages inside a span. . . . .	219	layout: Added. . . . .	497
v0.21		lscape: Added. . . . .	505
\LWR@LwarpStart: Changed lateximages to a .txt file. . .	258	mcaption: Added. . . . .	507
\LWR@filestart: Skip title if not given. . . . .	256	nameref: Added. . . . .	524
\LWR@newhtmlfile: Skip title if not given. . . . .	242	nextpage: Added. . . . .	525
\marginpar: Fixed source listing. .	239	parskip: Added. . . . .	540
General: 2017/02/23 . . . . .	1	showkeys: Added. . . . .	551
fontenc: Added. . . . .	469	sidecap: Added. . . . .	551
fontspec: Added. . . . .	470	tabularx: Added. . . . .	565
inputenc: Added. . . . .	495	varioref: Supported. . . . .	89
newclude: Added. . . . .	525	verse: Added. . . . .	602
newunicodechar: Added. . . . .	525	v0.23	
lwarpmk: Fix: lwarpmk again for Windows. . . . .	195	\LWR@parsetablecols: Fix for vert bar column type. . . . .	298
		\LWR@printmccoldata: Fix for vert bar column type. . . . .	312
		General: 2017/03/02 . . . . .	1
		v0.24	
		\LWR@hspace: Add: \hspace \fill converts to 2em . . . . .	415
		\LWR@htmlfileref: Fix: Index links while \tracinglwarpmk. . . . .	334

\hypertocfloat: List of floats responds to lofdepth, lotdepth. . . . .	354	sidenotes: Added. . . . .	552
General: 2017/03/15 . . . . .	1	lwarp.css: Improved responsive marginpar and marginblock. . .	156
floatrow: Support for subfig. . . .	463	Docs: Improved MiKTeX install instructions. . . . .	48, 51
subfig: Added. . . . .	557	Dollar span avoided in a lateximage. . . . .	362
tikz: For tikz v3.0.0 or later, auto-loads tikz babel library if necessary. . . . .	576	Footnotes now are $\LaTeX$ boxes instead of pagenotes. . . . .	235
Docs: Filename underscore. . .	73, 83	lateximage: Labels track page numbers of lateximages. . . . .	382
No longer preloads subcaption; conflicted with subfig. . . . .	140	Print mode now uses a minipage of \linewidth. . . . .	382
picture: Fix for inline images. . .	395	picture: Fix for \makebox in picture. . . . .	395
tikzpicture: Fix for inline images. .	577		
v0.25		v0.27	
\LWR@loadnever: Added the ability to prevent conflicting packages. .	133	\LWR@footnotetext: Fix for table footnote par tags. . . . .	236
\addcontentsline: Handles theorems. . . . .	346	General: 2017/04/04 . . . . .	1
General: 2016/03/22 . . . . .	1	lettrine: Added. . . . .	498
amsthm: Added. . . . .	430	microtype: Fix with Xe $\LaTeX$ , Lua $\LaTeX$ . . . . .	517
ccaption: Prevented. . . . .	440	soul: Added. . . . .	555
ellipsis: Added. . . . .	444	ulem: Added. . . . .	599
emptypage: Added. . . . .	444	Docs: Installing utilities for MacOS. . . . .	53
framed: Added. . . . .	473	Docs: Limitations of saveboxes. .	84
lips: Added. . . . .	499	Page geometry modified to reduce line overflow. . . . .	138
mdframed: Help avoid hyphenation. . . . .	509	v0.28	
nththeorem: Added. . . . .	527	\@wrindex: Improved indexing. . .	356
showidx: Added. . . . .	550	\HTMLAuthor: Added \HTMLauthor. (Renamed in v0.30.) . . . . .	234
theorem: Added. . . . .	571	\LWR@LwarpEnd: If FormatEpub or FormatWordProcessor, no bottom nav. . . . .	261
Basic $\LaTeX$ theorems: improved css. . . . .	277	\LWR@LwarpStart: FormatWordProcessor forces single-file output. . . . .	258
Docs: Adds credits for patched code. . . . .	1	\LWR@filestart: Adds HTML meta author. . . . .	256
Docs: Testing lwarp. . . . .	116	\LWR@forcenewpage: Forces new PDF page before major environments. . . . .	216
Fix: Allows Xe $\LaTeX$ and Lua $\LaTeX$ to preload graphics and graphicx. .	133	\LWR@htmlcomment: Breaks ligatures in HTML comments. . . . .	221
v0.26		\LWR@includegraphicsb: Adapts to graphics syntax. . . . .	482
General: 2017/03/31 . . . . .	1		
cutwin: Added. . . . .	442		
endnotes: Added. . . . .	445		
floatflt: Added. . . . .	461		
footmisc: Added. . . . .	470		
footnotehyper: Added. . . . .	472		
footnote: Added. . . . .	471		
marginfix: Added. . . . .	506		
marginnote: Added. . . . .	507		
mparhack: Added. . . . .	520		
pagenote: Supported as-is. . . .	539		



\LWR@newhtmlfile: If FormatEPUB or FormatWordProcessor: skips headers, footers, nav. . . . .	242	\HTMLDescription: Renamed from \NewHTMLdescription. . . . .	235
\LWR@parsetablecols: Added L, C, R, J column types. . . . .	298	\HTMLFirstPageTop: Renamed from \SetFirstPageTop. . . . .	233
\LWR@startref: Removed space. . .	336	\HTMLLanguage: Renamed from \MetaLanguage. . . . .	255
\chapter: If EPUB, prints footnotes before each section. . . . .	254	\HTMLPageBottom: Renamed from \SetPageBottom. . . . .	233
\hyperindexref: Improved indexing. . . . .	357	\HTMLPageTop: Renamed from \SetPageTop. . . . .	233
\textup: Fixed span class. . . . .	407	General: 2017/04/29 . . . . .	1
General: 2017/04/14 . . . . .	1	lwrap-newproject removed, and combined with lwrap. . . . .	154
glossaries: Added. . . . .	477	lwrapmk: Add: xdyfile configuration option. . . . .	195
graphics: Added. . . . .	478	lwrapmk: Fix: xindy and texindy adjusted for pdflatex, xelatex and lualatex. . . . .	195
tabularx: Fix for optional pos. . .	565	lwrapmk: Fix: xindy now used for print index generation with latexmk. . . . .	195
tabulary: Added. . . . .	566	lwrapmk: language now used for both index and glossary generation. . . . .	195
lwrapmk: Add: printglossary and htmlglossary commands. . . .	195	File: lwrap_html.xdy renamed to lwrap.xdy. . . . .	193
Added boolean FormatEPUB. . . .	147	Fix: *.css files only written in print mode. . . . .	156
Added boolean FormatWP. . . . .	147	Fix: lwrap.xdy only written in print mode. . . . .	193
Added boolean HTMLDebugComments. . . . .	147	Fix: lwrap_mathjax.txt: Only written in print mode. . . . .	193
Added boolean HTMLMarkFloats, changed to WPMarkFloats as of v0.42. . . . .	148	Option OSWindows replaces macro \warpOSWindows. . . . .	130
Docs: Modfying lwrapmk and index processing. . . . .	116	Option latexmk replaces macro \UseLatexmk. . . . .	130
File lwrap_mathjax.txt: Updated CDN repository. . . . .	193	Option lwrapmklang changed to IndexLanguage. . . . .	129
Forced oneside to maintain large right margin. . . . .	138	Option xdyFilename added. . . .	129
v0.29		Options HomeHTMLFilename and HTMLFilename replace macros \HomeHTMLFilename and \HTMLFilename. . . . .	130
\LWR@includegraphicsb: Fix: Error when no optional arguments. . .	482	v0.31	
General: 2017/04/15 . . . . .	1	General: 2017/05/15 . . . . .	1
lwrapmk: Add: language option for config files. . . . .	195	keyfloat: Improved compatibility. .	496
Add: lwrapmklang option for lwrap. . . . .	129	v0.32	
Docs: Using a glossary . . . . .	66	\RequirePackage: Fix: Ignores blanks in package list. . . . .	142
File *.lwrapmkconf: Add: language option for config files. .	155	General: 2016/06/09 . . . . .	1
File lwrapmk.conf: Add: language option for config files. .	155		
v0.30			
\CSSFilename: Renamed from \NewCSS. . . . .	234		
\HTMLAuthor: Renamed from \HTMLauthor. . . . .	234		

glossaries: Prevent error with \glo@name not defined. . . . .	357	\LWR@htmlspanclass: Moved optional argument in front of mandatory. . . . .	220
lwarpmk: Fix: io.lines() changed to file:lines() due to luatex changes. . . . .	195	\LWR@nullfonts: Improved font control. . . . .	409
v0.33		\LWR@restoreorigformatting: booktabs: Works inside lateximage. . . . .	357
\HTMLAuthor: Fix: Provides empty default author if none given. . .	234	Improved font control. . . . .	357
\LWR@loadbefore: Fix: No \PackageError if already loaded. . . . .	132	\LWR@subhtmlclass: Moved optional argument in front of mandatory. . . . .	222
\LWR@parseatcolumn: Fix: Column alignment with leftmost @. . . .	294	\LWR@tabledatacolumnntag: booktabs: Works inside lateximage. . . . .	323
\LWR@tabledatasinglecolumnntag: Fix: Macros in tabular could cause extra data cell. . . . .	304	\fboxBlock: Added. . . . .	403
\LWR@vspace: Add: \vspace nullified. . . . .	416	\makebox: Fix: Handles paren arg. .	401
\StartDefiningTabulars: Add: Avoids error: misplaced alignment tab character &. . . .	288	General: 2017/08/08 . . . . .	1
General: 2017/07/10 . . . . .	1	babel-french: Adds fixed-width HTML spaces to punctuation. .	217
amsmath: Removed fleqn option. . .	141	balance: Added. . . . .	435
fancyhdr: Fix: Optional args for \thead, etc. . . . .	453	booktabs: Works inside lateximage. . . . .	326, 438
Add: Tabular at and bang columns now have their own HTML columns. . . . .	284	boxedminipage2e: Added. . . . .	438
cleveref: Fix: Loaded \AtEndPreamble. . . . .	394	boxedminipage: Prevented. . . .	438
Fix: Incorrectly-inline math environments. . . . .	370	crop: Added. . . . .	441
New handling of & to localize catcode changes. . . . .	284	enumerate: Added. . . . .	445
v0.34		enumitem: Added, no longer required. . . . .	446
\@fnsymbol: Text symbols instead of math. . . . .	268	everyshi: Added. . . . .	448
\InlineClass: Moved optional argument in front of mandatory. .	225	fancybox: Added. . . . .	450
\LWR@htmldivclass: Moved optional argument in front of mandatory. . . . .	223	fancyvrb: Added, no longer required. . . . .	454
\LWR@htmlclass: Moved optional argument in front of mandatory. . . . .	222	figcaps: Added. . . . .	459
\LWR@htmlclassline: Moved optional argument in front of mandatory. . . . .	223	filecontents: Required. Patched for morewrites. . . . .	139
		floatpag: Added. . . . .	462
		flushend: Added. . . . .	468
		fullpage: Added. . . . .	475
		hyperxmp: Added. . . . .	494
		idxlayout: Added. . . . .	494
		marginfit: Added. . . . .	506
		mdframed: Improved mdtheorem patch. . . . .	514
		moreverb: Added. . . . .	518
		paralist: Added. . . . .	539
		pdfscape: Added. . . . .	540
		pdfsync: Added. . . . .	540
		prelim2e: Added. . . . .	541
		rotfloat: Added. . . . .	547

savetrees: Added. . . . .	548	\LWR@restoreorigformatting:	
shadow: Added. . . . .	550	siunitx: Improved	
syntonly: Added. . . . .	564	super/subscripts in a	
titles: No longer required. . . . .	577	lateximage. . . . .	357
titleref: Prevented. . . . .	580	\LWR@section: Improved spacing. . . . .	248
xmpincl: Added. . . . .	616	\LWR@stoppars: Extra HTML source	
Added. . . . .	612	space after paragraphs. . . . .	231
Docs: Horizontal space		\fbox: Fix: Uses \fboxrule and	
limitations. . . . .	1	\fboxsep. . . . .	403
Docs: Misplaced alignment		\framebox: Fix: Handles width and	
character. . . . .	118	horiz position. . . . .	402
File: README.txt: updated. . . . .	1	\makebox: Fix: Handles width and	
File: lwarp_mathjax.txt: Version		horiz position. . . . .	401
change. . . . .	193	General: 2017/08/17 . . . . .	1
Fix: Added the eqnarray		babel-french: Adjustements for	
environments. . . . .	370	French variants, load order,	
Improved font control. . . . .	406	footnotes, ellipses. . . . .	217
Lists refactored to remove		footnote: Extra HTML source space	
enumitem requirement. . . . .	278	after paragraphs. . . . .	471
Verbatim refactored to remove		siunitx: Fix for babel-french. . . . .	387
fancyvrb requirement. . . . .	274	siunitx: Improved symbol support. . . . .	553
lateximage: Fix: lateximage with		transparent: Added. . . . .	597
minipage, \parbox, \makebox,		upref: Added. . . . .	601
\fbox, \framebox, \raisebox,		xcolor: Added \fcolorboxBlock,	
\scalebox, \reflectbox. . . . .	382	\colorboxBlock. . . . .	605
BlockClass: Moved optional		xcolor: Fix: Background none in	
argument in front of mandatory. . . . .	224	print mode. . . . .	605
fminipage: Added. . . . .	404	xcolor: Refactored	
LWR@nestspan: Fix: Minipages,		\LWR@colorstyle. . . . .	608
BlocksClass, and lists inside a		xcolor: Uses \fboxrule and	
span. . . . .	219	\fboxsep. . . . .	605
LWR@tabular: booktabs: Works		xcolor: \fcolorbox etc. now work	
inside lateximage. . . . .	327	inside lateximage. . . . .	605
v0.35		Docs: Reorganized: Special cases	
General: 2017/08/08 . . . . .	1	and limitations. . . . .	82
Fix: \textbf and related. . . . .	406	Source: Improved formatting. . . . .	1
v0.36		lateximage: Footnotes appear in	
\LWR@HTMLsanitize: Fix for		regular text instead of the	
babel-french. . . . .	379	lateximage minipage. . . . .	382
\LWR@HTMLsanitizeexpand: Fix for		LWR@tabular: Fix for babel-french. . . . .	327
babel-french. . . . .	379	v0.37	
\LWR@closeparagraph: Extra HTML		\include: Maintains independent	
source space after paragraphs. . . . .	228	.aux files for HTML. . . . .	145
\LWR@currenttextcolor: Fix for		General: 2017/08/19 . . . . .	1
\rule when xcolor not loaded. . . . .	417	$\TeX$ accents: Added. . . . .	152
\LWR@footnotetext: Extra HTML		babel-french: Adjustment for load	
source space after paragraphs. . . . .	236	order. . . . .	217
Force HTML superscripts. . . . .	236	color: Prevented. . . . .	441
\LWR@nullfonts: Fix: Filenames		comment: Maintains independent	
while using MathJax. . . . .	409	cutfiles for print, HTML. . . . .	131

siunitx: Improved symbol support. . . . .	553	Supports authblk with <div>s of	
textcomp: Improved support. . . . .	567	class oneauthor instead of	
lwarpmk: Removes additional		tabular. . . . .	269, 587
HTML aux files. . . . .	195	\AddSubtitlePublished: Added. . . . .	271
File handles reorganized. . . . .	144	\LWR@domulticolumn: Add: Optional	
v0.38		vpos and # rows. . . . .	314
\@secntformat: Added for		\LWR@restoreorigformatting:	
appendix. . . . .	248	Appended with \appto instead of	
\ForceHTMLPage: Added. . . . .	245	calling various macros. . . . .	357
\ForceHTMLTOC: Added. . . . .	246	\LWR@tabledatacolumnntag: Don't	
\LWR@section: \part* starts a new		start a data cell if see	
HTML page, for appendix. . . . .	248	\TabularMacro. . . . .	323
Modified spacing, uses		\ResumeTabular: Added. . . . .	322
\numberline. . . . .	248	\TabularMacro: Added. . . . .	322
\numberline: Added trailing \quad. . . . .	352	\multicolumnrow: Added. . . . .	320, 523
\part: Fix with article class. . . . .	253	\printauthor: Removed minipages. . . . .	266
General: 2017/08/27 . . . . .	1	Supports authblk with <div>s of	
appendix: Added. . . . .	434	class oneauthor instead of	
arabicfront: Added. . . . .	434	tabular. . . . .	266
caption2: Prevented. . . . .	439	\thanksmarkseries: Removed	
chappg: Added. . . . .	440	minipage footnotes. . . . .	588
color: Forces xcolor as well. . . . .	441	General: 2017/09/05 . . . . .	1
fix2col: Added. . . . .	460	a4wide: Added. . . . .	425
fncychap: Added. . . . .	468	a4: Added. . . . .	425
grffile: Added. . . . .	490	a5comb: Added. . . . .	425
metalogo: Added. . . . .	516	addlines: Added. . . . .	428
nonumonpart: Added. . . . .	526	anysize: Added. . . . .	433
nopageno: Added. . . . .	526	authblk: Added. . . . .	434
pagenote: Option page disabled. . . . .	539	bigdelim: Added. . . . .	436
realscripts: Added. . . . .	544	bigstrut: Added. . . . .	437
relsize: Added. . . . .	545	chngepage: Prevented. . . . .	440
romanbarpagenumber: Added. . . . .	546	ebook: Added. . . . .	444
romanbar: Added. . . . .	546	fullwidth: Added. . . . .	476
scalefnt: Added. . . . .	548	supertabular: Added. . . . .	563
siunitx: Removed from lwarp core. . . . .	553	textarea: Added. . . . .	566
textcomp: Removed from lwarp		titling: Improved compatibility. . . . .	584
core. . . . .	567	titling: Removed extraneous	
tocbibind: Added. . . . .	590	center environments. . . . .	585
xltxtra: Added. . . . .	615	typearea: Added. . . . .	599
lwarpmk: Added print1 and html1		xtabular: Added. . . . .	616
actions. . . . .	195	zwpagelayout: Added. . . . .	618
Added \markboth, \sloppy, etc. . . . .	215	Add: New optional vpos	
Docs: Enhanced <i>Supported</i>		argument. . . . .	522
<i>Functions</i> table. . . . .	41	Add: Supports left/right border for	
Docs: Index, tocbibind. . . . .	92	bigdelim. . . . .	522
Docs: Starred sections. . . . .	90	Docs: Reorganized tabular	
v0.39		discussion. . . . .	99
\@maketitle: titling version. . . . .	587	Fix: Long text argument. . . . .	522
Native $\LaTeX$ version. . . . .	269	Titlepage \published and	
Removed minipages. . . . .	269, 587	\subtitle removed.	

\AddSubtitlePublished		quotchap: Added. ....	542
restores. ....	270	sectsty: Added. ....	548
titlepage: Clear pending footnotes. ....	265	stabular: Added. ....	557
Removed minipages. ....	265	tabs: Added. ....	565
titlingpage: Clear pending		textcomp: Additional symbols,	
footnotes. ....	585	improved XeLaTeX and LuaLaTeX	
v0.40		support. ....	567
\@chapcntformat: Added for		tocbibind: Improved for	
tocbibind, anonchap. ....	248	\simplechapter. ....	590
\LWR@hline: Added. ....	326	xltxtra: Fix for \showhyphens with	
\LWR@includegraphicsb: Add: Full		XeLaTeX. ....	615
\graphicspath support. ....	482	Improved bigdelim borders. ....	522
\LWR@nullfonts: Fix: Long		No longer preloads xfrac. ....	141
arguments for expandable			
command. ....	409	v0.41	
\LWR@restoreorigformatting:		\LWR@addcmidruletrim: Add:	
Improved $\TeX$ logos inside a		\cmidrule trims. ....	309
lateximage. ....	357	\LWR@clearmidrules: Add:	
Improved symbols inside a		\cmidrule trims. ....	307
lateximage. ....	357	\LWR@closetabledatacell: Add:	
Nullified \InlineClass, etc.		Mute > for \bottomrule. ....	287
inside a lateximage. ....	357	Fix: At/bang column with	
\LWR@tabledatacolumntag: Fix for		\multirow. ....	287
bigdelim: \ldelim, \rdelim. ..	323	Fix: Cancel < for \multicolumn. ....	287
\chapter: Added support for		\LWR@domulticolumn: Add:	
quotchap. ....	254	\cmidrule trims. ....	314
\multicolumnrow: Fix: Adapts to		Added vertical rules. ....	315
older multirow and xparse. ....	320	\LWR@nullifyNoAutoSpacing: Fix:	
\simplechapterdelim: Added for		\NoAutoSpacing in a tabular	
tocbibind, anonchap. ....	248	with babel-french. ....	326
\underline: Added. ....	411	\LWR@parsebarcolumn: Added	
General: 2017/09/25 ....	1	vertical rules. ....	296
adjmulticol: Added. ....	427	\LWR@printatbang: Add: \cmidrule	
anonchap: Added. ....	433	trims. ....	303
bigdelim: Improved		Add: Mute at and bang columns	
documentation. ....	436	for \bottomrule. ....	303
cuted: Added. ....	441	\LWR@printbartag: Added vertical	
dblfnote: Added. ....	442	rules. ....	303
fnpos: Added. ....	469	\LWR@subaddcmidruletrim: Added. ....	308
graphics: Moved out of the lwarp		\LWR@subcmidrule: Add: \cmidrule	
core. ....	478	trims. ....	307
graphics: Restores		\LWR@tabledatasinglecolumntag:	
\includegraphics and		Add: \cmidrule trims. ....	304
\DeclareGraphicsExtensions		Add: Mute < for \bottomrule. ..	304
in a lateximage. ....	478	\LWR@tabularfinishrow:	
graphicx: Moved out of the lwarp		Unfinished tabular rows	
core. ....	489	automatically filled. ....	290
grffile: Directly supported. ....	490	\mcolrowcell: Added for	
midfloat: Added. ....	518	\multicolumnrow cells. ....	325
pfnote: Added. ....	541	General: 2017/10/07 ....	1
		Add: \cmidrule trims. ....	522

Added vertical rules. . . . .	522	\LWR@singledollar: If FormatWP print LaTeX expression. . . . .	363
Fix: < spec. . . . .	523	\LWR@subaddcmidruletrim: Opt if no rule given. . . . .	308
\LWR@tabular: Fix: \NoAutoSpacing in a tabular with babel-french. . .	327	\LWR@tabledatasinglecolumn: If FormatWP add cell alignment. . . . .	305
Improved rules. . . . .	328	\LaTeX: If FormatWP use explicit style. . . . .	421
v0.42		\TeX: If FormatWP use explicit style. . . . .	420
\@ensuredmath: Improved		\listoffigures: Added boolean WPMarkLOFT. . . . .	350
\ensuremath. . . . .	364	\listoftables: Added boolean WPMarkLOFT. . . . .	350
\@textsubscript: Added. . . . .	411	\marginpar: If FormatWP emulate a wrapfig. . . . .	239
\@textsuperscript: Added. . . . .	411	\tableofcontents: Added boolean WPMarkTOC. . . . .	350
\LWR@HTMLtextstyle: Added. . . . .	406	\underline: If FormatWP, use explicit styles for \underline, etc. . . . .	411
\LWR@addformatwpalignment: If FormatWP add explicit style for cell alignment. . . . .	310	General: 2017/10/30 . . . . .	1
\LWR@addrulewidth: If FormatWP force explicit border. . . . .	309	\textbf and related: If FormatWP, use explicit styles for \textsc, etc. . . . .	406
\LWR@amsmathbody: Fix: Numbering and naming AMS math environments. . . . .	381	algorithmic: If FormatWP add \quads. . . . .	429
\LWR@amsmathbodynumbered: Fix: Numbering and naming AMS math environments. . . . .	382	epigraph: If FormatWP add HTML styles. . . . .	446
\LWR@domulticolumn: If FormatWP add cell alignment. . . . .	315	fancybox: If FormatWP add HTML styles. . . . .	450
\LWR@doubledollar: If FormatWP print LaTeX expression. . . . .	362	floatflt: Added width. . . . .	461
Improved \ensuremath. . . . .	362	includegraphics: Fix: Class key. . .	482
Improved line spacing with mathjax. . . . .	362	keyfloat: If FormatWP add explicit HTML style. . . . .	496
\LWR@floatbegin: If FormatWP add a text frame. . . . .	342	moreverb: Simplified formatting of listings. . . . .	518
\LWR@floatend: If FormatWP add a text frame. . . . .	343	morewrites: Added. . . . .	519
\LWR@hline: If FormatWP force explicit border. . . . .	326	multirow: If FormatWP add cell alignment. . . . .	522
\LWR@hspace: If FormatWP add \quads. . . . .	416	overpic: Added. . . . .	538
\LWR@htmlmathlabel: If FormatWP print LaTeX expression. . . . .	369	realscripts: Fix for subscripts in a lateximage. . . . .	544
\LWR@includegraphicsb: Fix: Filename expansion. . . . .	482	sidenotes: If FormatWP add explicit HTML style. . . . .	552
If FormatWP, use explicit size. . .	484	siunitx: Improved \ensuremath. . .	554
\LWR@remembertag: Fix: Numbering and naming AMS math environments. . . . .	381	soul: If FormatWP, add explicit styles. . . . .	555
\LWR@restoreorigformatting: Improved \ensuremath. . . . .	359	textcomp: Improved \interrobangdown. . . . .	567
\LWR@rule: If FormatWP add \quads. . . . .	419		

wrapfig: If FormatWP add explicit HTML style. . . . .	604	v0.43	
Added boolean WPMarkLOFT. . . .	148		\LWR@domulticolumn: Fix for vertical rules. . . . . 315
Added boolean WPMarkMath. . . .	149		Fix: Multicolumn trim. . . . 314, 315
Added boolean WPMarkMinipages. . . . .	148		\LWR@maybeprintpendingfootnotes: Added FootnoteDepth. . . . . 239
Added boolean WPMarkTOC. . . .	148		\LWR@nullfonts: Fix: Nullify dollar inside filenames. . . . . 409
Added boolean WPTitleHeading. . . .	149		\LWR@parsetablecols: Ignore spaces in col spec. . . . . 300
Docs: Added support page. . . . .	2		\LWR@printmccoltype: Added vertical rules. . . . . 311
Docs: Improper \prevdepth. . . .	118		\LWR@restoreorigformatting: Fix: \ref in math or lateximage. . . 357
Docs: Reorganized math limitations . . . . .	93		Fix: Improved math inside lateximages. . . . . 359
File: lwarp_mathjax.txt: Updated siunitx script. . . . .	193		\LWR@section: Fix: Expansion in comparison. . . . . 249
Fix: Numbering and naming AMS math environments. . . . .	380		Fix: Math in section name. 250, 252
If FormatWP, shift section headings. . . . .	149		Fix: Nullify fonts inside HTML comment. . . . . 250
tabbing: Added. . . . .	277		\TabularMacro: \newcommand instead of \relax to fix supertabular and xtab. . . . . 322
lateximage: Fix: Numbering and naming AMS math environments. . . . .	382		\href: Made robust. . . . . 339
center: If FormatWP use explicit text-align. . . . .	386		\nameref: Made robust. . . . . 338
minipage: Added boolean WPMarkMinipages. . . . .	400		\nolinkurl: Made robust. . . . . 339
If FormatWP add a text frame. . .	399		\url: Made robust. . . . . 340
eqnarray: Fix: Numbering and naming AMS math environments. . . . .	372		General: 2017/11/08 . . . . . 1
If FormatWP print LaTeX expression. . . . .	371		breakurl: Added. . . . . 439
equation: If FormatWP print LaTeX expression. . . . .	367		hyperref: Made robust. 490, 491, 493
LWR@BlockClassWP: Added to factor code. . . . .	225		hyperref: \Gauge added. . . . . 493
LWR@figcaption: If FormatWP forces italic captions. . . . .	345		luatodonotes: Added. . . . . 506
LWR@tabular: If FormatWP force explicit border. . . . .	328		todonotes: Added. . . . . 597
			LWR@currentautosec: Added. . . 248
			Added FootnoteDepth. . . . . 236
			Docs: HTML settings table. . . . 72
			Docs: Reorganized HTML customization. . . . . 72

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>362</i>
$\backslash \&$ .....	<i>211</i> , <i>5012</i>
$\backslash ($ .....	<i>6613</i>
$\backslash )$ .....	<i>6613</i>
$\backslash ,$ .....	<i>83</i>
$\backslash @\setcpageref$ .....	<i>7339</i>
$\backslash @\setcref$ .....	<i>7335</i>
$\backslash @\setcrefrange$ .....	<i>7336</i>
$\backslash @author$ .....	<i>263</i>
$\backslash @begintheorem$ .....	<i>4839</i>
$\backslash @capytype$ .....	<i>6183</i>
$\backslash @chapcntformat$ .....	<i>4208</i>
$\backslash @currentlabelname$ .....	<i>5956</i>
$\backslash @date$ .....	<i>263</i>
$\backslash @dlbfloat$ .....	<i>6159</i>
$\backslash @endtheorem$ .....	<i>4849</i>
$\backslash @ensuredmath$ .....	<i>6621</i>
$\backslash @float$ .....	<i>6159</i>
$\backslash @fnsymbol$ .....	<i>4649</i>
$\backslash @footnotetext$ .....	<i>3967</i>
$\backslash @include$ .....	<i>460</i>
$\backslash @makecaption$ .....	<i>6185</i>
$\backslash @makefnmark$ .....	<i>3939</i>
$\backslash @makefntext$ .....	<i>3938</i>
$\backslash @maketitle$ .....	<i>49</i> , <i>4672</i>
$\backslash @mpfootnotetext$ .....	<i>3968</i>
$\backslash @opargbegintheorem$ .....	<i>4844</i>
$\backslash @seccntformat$ .....	<i>4206</i>
$\backslash @starttoc$ .....	<i>6299</i>
$\backslash @textsubscript$ .....	<i>7726</i>
$\backslash @textsuperscript$ .....	<i>7722</i>
$\backslash @title$ .....	<i>263</i>
$\backslash @wrglossary$ .....	<i>6463</i>
$\backslash @wrindex$ .....	<i>6456</i>
$\backslash \backslash$ .....	<i>413</i>
$\$$ .....	<i>362</i>
$\$ \$$ .....	<i>362</i>
$\sim$ .....	<i>83</i>
A	
a4 (package) .....	<i>425</i>
a4wide (package) .....	<i>425</i>
a5comb (package) .....	<i>425</i>
abstract (environment) .....	<i>4735</i>
abstract (package) .....	<i>425</i>
$\backslash abstractname$ .....	<i>74</i> , <i>4734</i>
accents .....	<i>245</i>
$\backslash addcontentsline$ .....	<i>6249</i>
addlines (package) .....	<i>428</i>
$\backslash AddSubtitlePublished$ .....	<i>4697</i>
adjmulticol (package) .....	<i>427</i>
Adobe (program) .....	<i>47</i>
$\backslash affiliation$ .....	<i>4559</i>
afterpackage (package) .....	<i>139</i>
afterpage (package) .....	<i>428</i>
algorithmic	
with newfloat, trivfloat .....	<i>598</i>
algorithmicx (package) .....	<i>429</i>
align (environment) .....	<i>6888</i>
align* (environment) .....	<i>6911</i>
alignment tab character &, misplaced	
.....	<i>99</i> , <i>282</i> , <i>288</i>
alltt (package) .....	<i>430</i>
$\backslash AmS$ .....	<i>7953</i>
AMSMath	
split miss-numbered .....	<i>96</i> , <i>527</i>
amsmath (package) .....	<i>141</i>
amsthm (package) .....	<i>430</i>
$\backslash and$ .....	<i>263</i>
anonchap (package) .....	<i>433</i>
anysize (package) .....	<i>433</i>
appendix (package) .....	<i>433</i>
arabicfront (package) .....	<i>434</i>
array (package) .....	<i>331</i>
AsciiDoc (program) .....	<i>46</i>
AsciiDoctor (program) .....	<i>46</i>
Asciidoctor-LaTeX (program) .....	<i>46</i>
$\backslash attrib$ .....	<i>104</i> , <i>273</i> , <i>602</i>
$\backslash attribution$ .....	<i>4745</i>
authblk (package) .....	<i>434</i>
author	
HTML meta tag .....	<i>81</i> , <i>234</i>
$\backslash author$ .....	<i>81</i> , <i>263</i>
autosec .....	<i>247</i>



B	
babel (package) . . . . .	105
\backmatter . . . . .	4190
balance (package) . . . . .	435
\BaseJobname . . . . .	3352
BaseJobname (option) . . . . .	71, 129
baseline	
tabular . . . . .	298
\bfseries . . . . .	7711
\BibTeX . . . . .	7948
bigdelim (package) . . . . .	435
bigstrut (package) . . . . .	437
BlockClass (environment) . . . . .	3683
\BlockClassSingle . . . . .	3695
bookmark (package) . . . . .	437
booktabs (package) . . . . .	438
booleans:	
CombineHigherDepths . . . . .	74, 240
FileSectionNames . . . . .	74, 212
FormatEPUB . . . . .	107, 147
FormatWP . . . . .	109, 147
HTMLDebugComments . . . . .	147
LWR@amsmultline . . . . .	369
LWR@doingapar . . . . .	227
LWR@doingcmidrule . . . . .	285
LWR@doinghline . . . . .	285
LWR@doingstartpars . . . . .	227
LWR@doingtbrule . . . . .	285
LWR@emptyatbang . . . . .	286
LWR@exitingtabular . . . . .	323
LWR@freezethisfloat . . . . .	343
LWR@infloatrow . . . . .	482
LWR@intabularmetadata . . . . .	286
LWR@isstartingequation . . . . .	380
LWR@minipagefullwidth . . . . .	397
LWR@minipagethispar . . . . .	398
LWR@skipatbang . . . . .	286
LWR@skippingmcolrowcell . . . . .	286
LWR@skippingmrowcell . . . . .	286
LWR@starredlongtable . . . . .	316
LWR@startedrow . . . . .	285
LWR@tableparcell . . . . .	286
LWR@tabularcelladded . . . . .	285
LWR@tabularmutemods . . . . .	323
LWR@tracinglwarp . . . . .	146
LWR@validtablecol . . . . .	298
LWR@verbtags . . . . .	275
mathjax . . . . .	128
usingOSWindows . . . . .	127
warpingHTML . . . . .	128
warpingprint . . . . .	128
WPMarkFloats . . . . .	111, 147
WPMarkLOFT . . . . .	112, 148
WPMarkMath . . . . .	112, 149
WPMarkMinipages . . . . .	111, 148
WPMarkTOC . . . . .	111, 148
WPTitleHeading . . . . .	112, 149
boxedminipage (package) . . . . .	438
boxedminipage2e (package) . . . . .	438
\boxframe . . . . .	137
breakurl (package) . . . . .	439
bugs . . . . .	118
BVerbatim (environment) . . . . .	183
C	
calc (package) . . . . .	140
Calibre . . . . .	107
caption (package) . . . . .	140
caption2 (package) . . . . .	439
\caption@begin . . . . .	6228
\caption@end . . . . .	6228
\captionlistentry . . . . .	6232
\captionof . . . . .	6262
\CaptionSeparator . . . . .	6184
ccaption (package) . . . . .	440
center (environment) . . . . .	7172
\centering . . . . .	259
changepage (package) . . . . .	440
chappg (package) . . . . .	440
\chapter . . . . .	4351
chnpage (package) . . . . .	440
\citetitle . . . . .	4752
classes:	
internet . . . . .	45
cleveref (package) . . . . .	393
cmap (package) . . . . .	69
\color . . . . .	609
color (package) . . . . .	441
\colorbox . . . . .	45
\colorboxBlock . . . . .	55, 7231
CombineHigherDepths (boolean) . . . . .	74, 240
comment (package) . . . . .	131
Computer Modern . . . . .	68
\ConTeXt . . . . .	7945
counters:	
FileDepth . . . . .	73, 240
FootnoteDepth . . . . .	74, 236
lofdepth . . . . .	353
lotdepth . . . . .	353
LWR@currentautosec . . . . .	248

LWR@externalfilecnt	362		
LWR@htmlfilenumber	212		
LWR@latestautopage	344		
LWR@lateximagedepth	378		
LWR@lateximagenumber	378		
LWR@LIpage	378		
LWR@midrulecounter	306		
LWR@minipagedepth	396		
LWR@nextautofloat	344		
LWR@nextautopage	344		
LWR@nextequation	364		
LWR@prevFileDepth	248		
LWR@spandepth	227		
LWR@startingequation	380		
LWR@tablecolindex	293		
LWR@tablecolspecindex	293		
LWR@tablecolspecwidth	293		
LWR@tabletotalcols	293		
LWR@tabletotalcolsnext	293		
LWR@thisfloat	343		
SideTOCDepth	73, 351		
tocdepth	73		
\cpagerefFor	7338		
crop (package)	441		
cross-references			
missing or incorrect	118		
CSS			
file selection	77		
lwrap.css	78		
per HTML page	77		
project-specific changes	78		
\CSSFilename	74, 77, 3905		
cuted (package)	441		
cutwin (package)	442		
<b>D</b>			
danger icon	124		
\date	81		
dblfnote (package)	442		
dcolumn (package)	443		
debugging	118		
HTML debug comments	147		
\DeclareGraphicsExtensions	2		
Deja Vu	68		
description			
HTML meta tag	81, 235		
description (environment)	4903		
displaymath (environment)	6635		
\dotfill	7738		
draftwatermark (package)	443		
<b>E</b>			
ebook (package)	444		
ellipsis (package)	444		
\emph	7611		
emptypage (package)	444		
\end@dlbfloat	6175		
\end@float	6175		
\EndDefiningTabulars	5006		
endnotes (package)	445		
\enlargethispage	7850		
\enskip	414, 7803		
enumerate (environment)	4875		
enumerate (package)	445		
enumitem (package)	446		
environ (package)	140		
environments:			
abstract	4735		
align	6888		
align*	6911		
BlockClass	3683		
BVerbatim	183		
center	7172		
description	4903		
displaymath	6635		
enumerate	4875		
eqnarray	6743		
equation	6682		
equation*	6706		
fcolorminipage	121, 7273		
flalign	6934		
flalign*	6957		
flushleft	7188		
flushright	7180		
fminipage	85, 7554, 7602		
gather	6842		
gather*	6865		
itemize	4861		
lateximage	377, 7090		
longtable	2		
LWR@BlockClassWP	3711		
LWR@figcaption	6194		
LWR@nestspan	3527		
LWR@tabular	5841		
LWRrprint@fminipage	7578		
math	6634		
minipage	7404		
multiline	6793		
multiline*	6817		
picture	395, 7359		
quote	4760		

tabbing	4827	files:	
theindex	6434	glyphtounicode	69
tikzpicture	8	lwarp.css	78, 156
titlepage	80, 4576	lwarp.xdy	82, 193
titlingpage	14, 80	lwarp_formal.css	188
Verbatim	13	lwarp_mathjax.txt	193
verbatim	4817	lwarp_sagebrush.css	183
verse	2	lwarp_tutorial.txt	55
warpall	79, 100	lwarpmk.conf	154
warpHTML	50, 76, 79, 101	lwarpmk.lua	116
warpprint	75, 79, 101	project.css	78
epigraph (package)	446	project.lwarpmkconf	155
EPUB		project_html.tex	154
conversion software	107	sample_project.css	78, 192
HTML conversion settings	107, 147	tutorial.tex	55
eqnarray (environment)	6743	FileSectionNames (boolean)	74, 212
equation		fix2col (package)	460
miss-numbered	96, 527	\flagverse	602
equation (environment)	6682	flalign (environment)	6934
equation* (environment)	6706	flalign* (environment)	6957
error messages	118	Flare (program)	47
eso-pic (package)	447	float (package)	460
etoolbox (package)	126	floatflt (package)	461
everyhook (package)	139	\floatname	9
everypage (package)	448	floatpag (package)	462
everyshi (package)	448	\floatplacement	12
expl3 (package)	139	floatrow (package)	462
export		\floatstyle	15
to word processor	109	\flushbottom	3446
extramarks (package)	449	flushend (package)	468
<b>F</b>		flushleft (environment)	7188
fancybox (package)	449	flushright (environment)	7180
fancyhdr (package)	453	fminipage (environment)	85, 7554, 7602
fancyvrb (package)	454	fncychap (package)	468
FAQ	118	fnpos (package)	469
fbox		font	
around a minipage	85	bitmapped	68
\fbox	85, 7537	Computer Modern	68
\fboxBlock	85, 7546, 7601	Deja Vu	68
\fcolorbox	67, 7237	ligatures	68, 69
\fcolorboxBlock	87, 7272	packages	68
fcolorminipage (environment)	121, 7273	selection	68
figcaps (package)	459	size — lateximage	94, 360, 378
file		size — math, SVG	94, 360, 378
inaccessible	73, 74	size — xfrac	613
underscore	73, 83	fontenc (package)	68, 469
filecontents (package)	139	fontspec (package)	68, 470
FileDepth (counter)	73, 240	footmisc (package)	470
		footnote (package)	471
		FootnoteDepth (counter)	74, 236





lwrap_formal.css (file) . . . . .	188	LWR@doingstartpars (boolean) . . . . .	227
lwrap_mathjax.txt (file) . . . . .	193	LWR@doingtbrule (boolean) . . . . .	285
lwrap_sagebrush.css (file) . . . . .	183	\LWR@domulticolumn . . . . .	5554
lwrap_tutorial.txt (file) . . . . .	55	\LWR@doubledollar . . . . .	6579
lwrapmk		LWR@emptyatbang (boolean) . . . . .	286
customizing . . . . .	116	\LWR@endhideamsmath . . . . .	6736
[lwrapmk]:		\LWR@endminipage . . . . .	7368
htmlglossary (option) . . . . .	92, 477	\LWR@endoffline . . . . .	7786
printglossary (option) . . . . .	92, 477	\LWR@endsubminipage . . . . .	7389
lwrapmk (option) . . . . .	71, 129	\LWR@ensuredoingapar . . . . .	3770
lwrapmk (program) . . . . .	116, 195	\LWR@equationtag . . . . .	7049
lwrapmk.conf (file) . . . . .	154	LWR@exitingtabular (boolean) . . . . .	323
lwrapmk.lua (file) . . . . .	116	LWR@externalfilecnt (counter) . . . . .	362
\LWR@addcmidruletrim . . . . .	5428	LWR@figcaption (environment) . . . . .	6194
\LWR@addcmidrulewidth . . . . .	5453	\LWR@filenamenoblanks . . . . .	4035
\LWR@addformatwppalignment . . . . .	5460	\LWR@filestart . . . . .	4417
\LWR@addleftmostbartag . . . . .	5309	\LWR@findword . . . . .	389
\LWR@addmathjax . . . . .	6671	\LWR@floatbegin . . . . .	6138
\LWR@addrulewidth . . . . .	5433	\LWR@floatend . . . . .	6161
\LWR@afterendverbatim . . . . .	4801	\LWR@footnotetext . . . . .	3940
\LWR@amsmathbody . . . . .	7077	\LWR@forceminwidth . . . . .	7521
\LWR@amsmathbodynumbered . . . . .	7083	\LWR@forcenewpage . . . . .	3438
LWR@amsmultline (boolean) . . . . .	369	LWR@freezethisfloat (boolean) . . . . .	343
\LWR@atbeginverbatim . . . . .	4787	\LWR@futurenonspacelct . . . . .	4929
\LWR@beginhideamsmath . . . . .	6728	\LWR@FVstyle . . . . .	17
\LWR@blackborderpadding . . . . .	7532	\LWR@getexparray . . . . .	3331
LWR@BlockClassWP (environment) . . . . .	3711	\LWR@getmynexttoken . . . . .	4936
\LWR@botnavigation . . . . .	3412	\LWR@heavyrulewidth (length) . . . . .	306
\LWR@caption@begin . . . . .	6209	\LWR@hidelatexequation . . . . .	6657
\LWR@caption@end . . . . .	6219	\LWR@hline . . . . .	5811
\LWR@clearmidrules . . . . .	5374	\LWR@hspace . . . . .	7811
\LWR@closeparagraph . . . . .	3797	\LWR@htmlblockcomment . . . . .	3623
\LWR@closeprevious . . . . .	3432	\LWR@htmlblocktag . . . . .	3631
\LWR@closetabledatacell . . . . .	4966	\LWR@htmlclosecomment . . . . .	3595
\LWR@cmidrulewidth (length) . . . . .	306	\LWR@htmlcomment . . . . .	3616
LWR@colafterspec . . . . .	293	\LWR@htmldivclass . . . . .	3660
LWR@colatspec . . . . .	293	\LWR@htmldivclassend . . . . .	3663
LWR@colbangspec . . . . .	293	\LWR@htmllelement . . . . .	3673
LWR@colbarspec . . . . .	293	\LWR@htmllelementclass . . . . .	3647
LWR@colbeforespec . . . . .	293	\LWR@htmllelementclassend . . . . .	3652
\LWR@copyfile . . . . .	495	\LWR@htmllelementclassline . . . . .	3666
\LWR@createautosec . . . . .	4196	\LWR@htmllelementend . . . . .	3676
LWR@currentautosec (counter) . . . . .	248	LWR@htmlfilenumber (counter) . . . . .	212
\LWR@currentcss . . . . .	3904	\LWR@htmlfileref . . . . .	5982
\LWR@currenttextcolor . . . . .	7851	\LWR@htmlmathlabel . . . . .	6713
\LWR@descitem . . . . .	4889	\LWR@htmlmulticolumn . . . . .	5590
\LWR@docmidrule . . . . .	5399	\LWR@htmlopencomment . . . . .	3595
LWR@doingapar (boolean) . . . . .	227	\LWR@htmlrefsectionfilename . . . . .	3397
LWR@doingcmidrule (boolean) . . . . .	285	\LWR@HTMLsanitize . . . . .	6994
LWR@doinghline (boolean) . . . . .	285	\LWR@HTMLsanitizeexpand . . . . .	7015

\LWR@htmlsectionfilename	3364	\LWR@newlabel	6016
\LWR@htmlspan	3568	\LWR@newref	6066
\LWR@htmlspanclass	3576	LWR@nextautofloat (counter)	344
\LWR@htmltag	3586	LWR@nextautopage (counter)	344
\LWR@htmltagc	3517	LWR@nextequation (counter)	364
\LWR@HTMLtextstyle	7606	\LWR@nohspace	7840
\LWR@includegraphicsb	105	\LWR@nullfonts	7676
\LWR@indexitem	6444, 6448, 6452	\LWR@nullifyNoAutoSpacing	5828
LWR@infloatrow (boolean)	482	\LWR@openparagraph	3775
\LWR@instertatbangcols	4958	\LWR@origcolspec	5087
LWR@intabularmetadata (boolean)	286	\LWR@originname	244
LWR@isstartingequation (boolean)	380	\LWR@originnames	251
\LWR@itemizeitem	4854	\LWR@parseaftercolumn	5148
LWR@latestautopage (counter)	344	\LWR@parseatcolumn	5094
LWR@lateximagedepth (counter)	378	\LWR@parsebangcolumn	5118
\LWR@lateximagedepthref	5986	\LWR@parsebarcolumn	5157
LWR@lateximagenumber (counter)	378	\LWR@parsebeforecolumn	5139
\LWR@lateximagenumberref	5989	\LWR@parseDcolumn	5184
\LWR@lateximagesfile	457	\LWR@parsenormalcolumn	5168
\LWR@lightrulewidth (length)	306	\LWR@parsepcolumn	5180
LWR@LPage (counter)	378	\LWR@parsetablecols	5189
\LWR@loadafter	111	\LWR@patchlists	4917
\LWR@loadbefore	122	LWR@prevFileDepth (counter)	248
\LWR@loadnever	131	\LWR@printatbang	5286
\LWR@longtabledatacaptiontag	5601	\LWR@printbartag	5278
\LWR@lookforpackagename	392	\LWR@printmccoldata	5513
\LWR@LwarpEnd	4536, 7975	\LWR@printmccoltype	5478
\LWR@LwarpStart	4467, 7975	\LWR@printmccolpendingfootnotes	3995
\LWR@maketitlesetup	33, 4641	\LWR@printthetitle	4600
\LWR@maybeinthisfloat	6180	\LWR@ProvidesPackageDrop	444
\LWR@maybeinnewtablerow	5256	\LWR@ProvidesPackagePass	432
\LWR@maybeprintpendingfootnotes	4005	\LWR@pushhonestclose	4199
LWR@midrulecounter (counter)	306	\LWR@quickfile	454
LWR@midrules	306	\LWR@remembertag	7053
LWR@minipagedepth (counter)	396	\LWR@requesttoc	4529
LWR@minipagefullwidth (boolean)	397	\LWR@requirepackagenames	388
LWR@minipageheight (length)	396	\LWR@restoreorigaccents	704
\LWR@minipagestartpars	7791	\LWR@restoreorigformatting	6478
\LWR@minipagestoppars	7794	\LWR@restoreorigprintxcolor	7311
LWR@minipagethispar (boolean)	398	\LWR@rotboxorigin	243
LWR@minipagewidth (length)	396	\LWR@rotstyle	91
\LWR@multicolother	5504	\LWR@rule	7852
\LWR@multicolparttext	5499	\LWR@scalestyle	94
\LWR@multicolskip	5510	\LWR@section	4210
\LWR@multirowborder	3	\LWR@sectionnumber	4193
\LWR@mynexttoken	4928	\LWR@setexparray	3326
\LWR@myshorttoc	6274	\LWR@setlatestname	5958
\LWR@nameref	5979	\LWR@setOSWindows	29
LWR@nestspan (environment)	3527	\LWR@sidetoc	6359
\LWR@newhtmlfile	4101	\LWR@singledollar	6597









OSWindows (option) . . . . . 70, 78, 127, 130  
overpic (package) . . . . . 538

## P

packages  
    required . . . . . 137  
packages:  
    a4 . . . . . 425  
    a4wide . . . . . 425  
    a5comb . . . . . 425  
    abstract . . . . . 425  
    addlines . . . . . 428  
    adjmulticol . . . . . 427  
    afterpackage . . . . . 139  
    afterpage . . . . . 428  
    algorithmicx . . . . . 429  
    alltt . . . . . 430  
    amsmath . . . . . 141  
    amsthm . . . . . 430  
    anonchap . . . . . 433  
    anysize . . . . . 433  
    appendix . . . . . 433  
    arabicfront . . . . . 434  
    array . . . . . 331  
    authblk . . . . . 434  
    babel . . . . . 105  
    balance . . . . . 435  
    bigdelim . . . . . 435  
    bigstrut . . . . . 437  
    bookmark . . . . . 437  
    booktabs . . . . . 438  
    boxedminipage . . . . . 438  
    boxedminipage2e . . . . . 438  
    breakurl . . . . . 439  
    calc . . . . . 140  
    caption . . . . . 140  
    caption2 . . . . . 439  
    ccaption . . . . . 440  
    changepage . . . . . 440  
    chappg . . . . . 440  
    chnpage . . . . . 440  
    cleveref . . . . . 393  
    cmap . . . . . 69  
    color . . . . . 441  
    comment . . . . . 131  
    crop . . . . . 441  
    cuted . . . . . 441  
    cutwin . . . . . 442  
    dblfnote . . . . . 442  
    dcolumn . . . . . 443

draftwatermark . . . . . 443  
ebook . . . . . 444  
ellipsis . . . . . 444  
emptypage . . . . . 444  
endnotes . . . . . 445  
enumerate . . . . . 445  
enumitem . . . . . 446  
environ . . . . . 140  
epigraph . . . . . 446  
eso-pic . . . . . 447  
etoolbox . . . . . 126  
everyhook . . . . . 139  
everypage . . . . . 448  
everyshi . . . . . 448  
expl3 . . . . . 139  
extramarks . . . . . 449  
fancybox . . . . . 449  
fancyhdr . . . . . 453  
fancyvrb . . . . . 454  
figcaps . . . . . 459  
filecontents . . . . . 139  
fix2col . . . . . 460  
float . . . . . 460  
floatflt . . . . . 461  
floatpag . . . . . 462  
floatrow . . . . . 462  
flushend . . . . . 468  
fncychap . . . . . 468  
fnpos . . . . . 469  
fontenc . . . . . 68, 469  
fontspec . . . . . 68, 470  
footmisc . . . . . 470  
footnote . . . . . 471  
footnotehyper . . . . . 472  
framed . . . . . 473  
ftnright . . . . . 475  
fullpage . . . . . 475  
fullwidth . . . . . 476  
geometry . . . . . 138, 476  
getttitlestring . . . . . 139  
glossaries . . . . . 91, 477  
graphics . . . . . 478  
graphicx . . . . . 489  
grffile . . . . . 489  
hyperref . . . . . 338, 490  
hyperxmp . . . . . 494  
idxlayout . . . . . 494  
ifplatform . . . . . 126  
indentfirst . . . . . 495  
inputenc . . . . . 68, 495

keyfloat	496	quotchap	542
kvoptions	128	ragged2e	543
layout	497	realscripts	544
letltxmacro	126	refcount	140
letterspace	497	relsize	544
lettrine	498	romanbar	546
lips	499	romanbarpagenumber	546
listings	499	rotating	546
lmodern	68	rotfloat	547
longtable	503	savetrees	548
lscap	505	scalefnt	548
ltcaption	505	sectsty	548
luatodonotes	505	setspace	549
lwrap	70	shadow	550
makeidx	140	showidx	550
marginfit	506	showkeys	551
marginfix	506	sidecap	551
marginnote	507	sidenotes	552
mcaption	507	siunitx	96, 387, 553
mdframed	507	soul	555
memoir	602	stabular	557
metalogo	516	subfig	557
microtype	69, 137, 517	supertabular	563
midfloat	518	syntonly	564
moreverb	518	tabls	565
morewrites	519	tabularx	565
mparhack	520	tabulary	566
multicol	520	textarea	566
multirow	521	textcomp	69, 567
nameref	524	textpos	570
needspace	524	theorem	571
newclude	105, 525	threeparttable	575
newfloat	140	tikz	576
newtxmath	97	titleps	577
newunicodechar	68, 525	titleref	580
nextpage	525	titlesec	580
nicefrac	96	titletoc	582
nonumonpart	526	titling	584
nopageno	526	tocbibind	93, 589
nowidow	526	tocloft	591
ntheorem	527	todonotes	106, 596
overpic	538	transparent	597
pagenote	539	trivfloat	597
paralist	539	typearea	599
parskip	540	ulem	599
pdfscape	540	units	96
pdfsync	540	upref	601
pfnote	541	verse	602
placeins	541	wallpaper	603
prelim2e	541	wrapfig	604

xcolor	389, 605	\printtitle	263, 4592, 4626
xfrac	613	problems	118
xifthen	140	programs	
xltextra	615	utility	48
xmpincl	616	programs:	
xparse	138	[requirement]:	
xstring	140	LuaLaTeX	49
xtab	616	MathJax	49
zref	141	pdfcrop	49
zwpagelayout	618	pdfLaTeX	49
page		pdfseparate	49, 53
inaccessible	73, 74	pdftocairo	49, 53
\pagebreak	7846	pdftotext	49, 53
\pagecolor	43	perl	54
pagenote (package)	539	XeLaTeX	49
\pagenumbering	3449	Adobe	47
\pageref	6073	AsciiDoc	46
\pagerefPageFor	6072	AsciiDoctor	46
\pagestyle	3441	Asciidoctor-LaTeX	46
Pandoc (program)	46	Flare	47
\paragraph	4392	FrameMaker	47
paralist (package)	539	GELLMU	45
\parbox	7486	GladTeX	46
\parsemulticolumnalignment	5540	Hevea	45
parskip (package)	540	InDesign	47
\part	4343	LaTeX2HTML	45
pdfcrop (program) [requirement]	49	LaTeXML	45
pdfLaTeX (program) [requirement]	49	LibreOffice	46
pdfscape (package)	540	Linux	78, 127
pdfseparate (program) [requirement]	49, 53	lwarpmk	116, 195
pdfsync (package)	540	Mac OS	78, 127
pdftocairo (program) [requirement]	49, 53	Madcap	47
pdftotext (program) [requirement]	49, 53	MathJax	94, 95, 360
Perl	54	MS-Windows	78, 127
perl (program) [requirement]	54	OpenOffice	46
pfnote (package)	541	Pandoc	46
\phantomsection	7904	Plastex	45
picture (environment)	395, 7359	TeX2page	45
placeins (package)	541	TeX4ht	45
Plastex (program)	45	TeXMaths	112
\popclose	3299	TtH	45
Poppler	49, 53	Unix	78, 127
prelim2e (package)	541	Windows	78, 127
\printauthor	263, 4608, 4627	Word	46
\printdate	263, 4619, 4629	xindy	82
printglossary (option) [lwarpmk]	92, 477	project.css (file)	78
\printindex	6427	project.lwarpmkconf (file)	155
\PrintStack	3417	project_html.tex (file)	154
\printthanks	263, 265	published	270
		\pushclose	3272

<b>Q</b>	
\qqquad	414, 7802
\quad	414, 7797
quotchap (package)	542
quote (environment)	4760
<b>R</b>	
ragged2e (package)	543
\raggedbottom	3445
\raggedleft	259
\raggedright	259
\raisebox	7571
realscripts (package)	544
\ref	6066
refcount (package)	140
\reflectbox	297
relsize (package)	544
[requirement]:	
LuaLaTeX (program)	49
MathJax (program)	49
pdfcrop (program)	49
pdfLaTeX (program)	49
pdfseparate (program)	49, 53
pdftocairo (program)	49, 53
pdftotext (program)	49, 53
perl (program)	54
XeLaTeX (program)	49
\RequirePackage	408
\resizebox	302
\restylefloat	17
\ResumeTabular	5727
\rmfamily	7712
romanbar (package)	546
romanbarpagenumber (package)	546
\rotatebox	263
rotating (package)	546
rotfloat (package)	547
\rule	7901
<b>S</b>	
sample_project.css (file)	78, 192
savetrees (package)	548
\sb	7720
\scalebox	282
scalegnt (package)	548
\scshape	7717
section	
depths	122
heading, word processor	112
missing	73, 74
\section	4367
sectsty (package)	548
\SetHTMLFileNumber	3355
setspace (package)	549
settings	
css project-specific	78
CSS selection	77
HTML conversion	72
lwrap package options	70
selecting output	78
title page	80
\sffamily	7713
\sfrac	613
shadow (package)	550
showidx (package)	550
showkeys (package)	551
sidecap (package)	551
sidenotes (package)	552
SideTOCDepth (counter)	73, 351
\sitetocname	6356
\simplechapterdelim	4207
siunitx	
with TeXMaths	112
siunitx (package)	96, 387, 553
\sloppy	3447
soul (package)	555
\sp	7719
space	
horizontal	83
between minipages	413
split	
miss-numbered	96, 527
stabular (package)	557
stack depths	122
\StartDefiningTabulars	5002
subequations	
MathJax	95, 361
subfig (package)	557
\subparagraph	4400
\subsection	4375
\subsubsection	4383
subtitle	270
supertabular (package)	563
SVG	
images showing as HTML	61, 119
math summary	94, 360
mathsvg option	129
syntonly (package)	564



<code>\upshape</code> .....	7715	Word (program) .....	46
<code>\url</code> .....	6118	word processor	
<code>usingOSWindows</code> (boolean) .....	127	conversion recommendations .....	112
UTF-8		HTML conversion settings .....	109, 147, 148
enhanced coverage .....	68	section headings .....	112
file & section names .....	245	<code>WPMarkFloats</code> (boolean) .....	111, 147
index .....	69	<code>WPMarkLOFT</code> (boolean) .....	112, 148
selection .....	68	<code>WPMarkMath</code> (boolean) .....	112, 149
utility		<code>WPMarkMinipages</code> (boolean) .....	111, 148
programs .....	48	<code>WPMarkTOC</code> (boolean) .....	111, 148
		<code>WPTitleHeading</code> (boolean) .....	112, 149
		<code>wrapfig</code> (package) .....	604
<b>V</b>			
<code>Verbatim</code> (environment) .....	13	<b>X</b>	
<code>verbatim</code> (environment) .....	4817	<code>xcolor</code> (package) .....	389, 605
<code>\VerbatimHTMLWidth</code> (length) .....	274	<code>xdyFilename</code> (option) .....	71, 82, 129
<code>\verbatiminput</code> .....	4809	XeLaTeX	
<code>verse</code> (environment) .....	2	detection .....	137
<code>verse</code> (package) .....	602	file & section names .....	245
viewport		<code>\XeLaTeX</code> .....	7941
HTML meta tag .....	257	<code>XeLaTeX</code> (program) [requirement] .....	49
<code>vleftmargini</code> (length) .....	104, 274, 602	<code>\XeTeX</code> .....	7941
<code>vleftskip</code> (length) .....	104, 274, 602	<code>xfrac</code> (package) .....	613
<code>\vspace</code> .....	7843	<code>\xfracHTMLfontsize</code> .....	3
		<code>xifthen</code> (package) .....	140
<b>W</b>			
<code>wallpaper</code> (package) .....	603	xindy	
warning icon .....	124	customizing .....	82
<code>warpall</code> (environment) .....	79, 100	xindy (program) .....	82
<code>warpHTML</code> (environment) ..	50, 76, 79, 101	<code>xltextra</code> (package) .....	615
<code>warpHTML</code> (option) .....	70, 128	<code>xmpinl</code> (package) .....	616
<code>\warpHTMLonly</code> .....	76, 79, 90	<code>xparse</code> (package) .....	138
<code>warpingHTML</code> (boolean) .....	128	<code>xstring</code> (package) .....	140
<code>warpingprint</code> (boolean) .....	128	<code>xtab</code> (package) .....	616
<code>warpprint</code> (environment) .....	75, 79, 101	<b>Z</b>	
<code>warpprint</code> (option) .....	70, 128	<code>zref</code> (package) .....	141
<code>\warpprintonly</code> .....	76, 79, 89	<code>zwpagelayout</code> (package) .....	618
<code>Windows</code> (program) .....	78, 127		