



# The **l warp** package

## $\text{\LaTeX}$ to HTML

v0.42 — 2017/10/30

© 2016–2017 Brian Dunn  
bd@BDTechConcepts.com

### Abstract

The **l warp** 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 **l warp** 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 **l warp**, and some things may change in response to user feedback and further project development.

#### License:

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

## 1 Supporting TeX development

TeX 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 TeX 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 TeXLive, the major TeX distribution.

Donations may be directed towards individual projects:

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

**CTAN:** The Comprehensive TeX Archive Network — Central storage for TeX.

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

**EduTeX:** Teaching and using TeX 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 pdfTeX engine and the Lua language.

**MetaPost:** Postscript graphics.

**MacTeX:** TeX 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 MiK<sup>T</sup>E<sub>X</sub>:

<https://miktex.org/donations.html>

For country-specific TeX 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>39</b>
3.1	Supported packages and features . . . . .	40
<b>4</b>	<b>Alternatives</b>	<b>44</b>
4.1	Internet class . . . . .	44
4.2	TeX4ht . . . . .	44
4.3	Translators . . . . .	44
4.4	AsciiDoc . . . . .	45
4.5	Pandoc . . . . .	45
4.6	Word processors . . . . .	45
4.7	Commercial systems . . . . .	46
4.8	Comparisons . . . . .	46
<b>5</b>	<b>Installation</b>	<b>47</b>
5.1	Installing the <code>l warp</code> package . . . . .	47
5.2	Installing the <code>l warpmk</code> utility . . . . .	50
5.2.1	Using a local copy of <code>l warpmk</code> . . . . .	51
5.3	Installing additional utilities . . . . .	52
<b>6</b>	<b>Tutorial</b>	<b>54</b>
6.1	Starting a new project . . . . .	54
6.2	Compiling the print version with <code>l warpmk</code> . . . . .	58
6.3	Compiling the HTML version with <code>l warpmk</code> . . . . .	59
6.4	Generating the SVG images . . . . .	60
6.5	Using MathJax for math . . . . .	61
6.6	Changing the CSS style . . . . .	62
6.7	Customizing the HTML output . . . . .	63
6.8	Using <code>latexmk</code> . . . . .	68
6.9	Using XeLaTeX or LuaLaTeX . . . . .	69
6.10	Using a glossary . . . . .	70
6.11	Cleaning auxiliary files . . . . .	71
6.12	Cleaning auxiliary and output files . . . . .	71
6.13	Processing multiple projects in the same directory . . . . .	71
6.14	Using the <code>make</code> utility . . . . .	71
<b>7</b>	<b>Additional details</b>	<b>72</b>
7.1	Font and UTF-8 support . . . . .	72

7.1.1	Indexes and UTF-8 . . . . .	73
7.2	lwarf package loading and options . . . . .	73
7.3	Selecting the operating system . . . . .	74
7.4	Selecting actions for print or HTML output . . . . .	75
7.5	Commands to be placed into the <code>warpprint</code> environment . . . . .	76
7.6	Title page . . . . .	76
7.7	HTML page meta descriptions . . . . .	77
7.8	HTML page meta author . . . . .	77
7.9	CSS . . . . .	78
7.10	Modifying <code>xindy</code> index processing . . . . .	78
<b>8</b>	<b>Special cases and limitations</b>	<b>79</b>
8.1	Formatting . . . . .	79
8.1.1	Text formatting . . . . .	79
8.1.2	Horizontal space . . . . .	79
8.1.3	Text alignment . . . . .	80
8.1.4	Accents . . . . .	80
8.1.5	Textcomp . . . . .	80
8.1.6	Superscripts and other non-math uses of math mode . . . . .	80
8.1.7	Empty <code>\item</code> followed by a new line of text or a nested list: . . . . .	80
8.1.8	Filenames in lists . . . . .	80
8.1.9	<code>relsize</code> package . . . . .	80
8.2	Boxes and minipages . . . . .	81
8.2.1	Save Boxes . . . . .	81
8.2.2	Minipages . . . . .	81
8.2.3	Side-by-side minipages . . . . .	82
8.2.4	Framed minipages and other environments . . . . .	82
8.2.5	<code>fancybox</code> package . . . . .	83
8.2.6	<code>mdframed</code> package . . . . .	85
8.3	Cross-references . . . . .	85
8.3.1	Page references . . . . .	85
8.3.2	<code>cleveref</code> and <code>varioref</code> packages . . . . .	86
8.3.3	Hyperlinks, <code>hyperref</code> , and <code>url</code> . . . . .	86
8.3.4	Footnotes and page notes . . . . .	86
8.4	Front and back matter . . . . .	86
8.4.1	Starred chapters and sections . . . . .	86
8.4.2	<code>abstract</code> package . . . . .	87
8.4.3	<code>titling</code> and <code>authblk</code> . . . . .	87
8.4.4	<code>tocloft</code> package . . . . .	88
8.4.5	<code>appendix</code> package . . . . .	88
8.4.6	<code>pagenote</code> package . . . . .	88
8.4.7	<code>endnotes</code> package . . . . .	88
8.4.8	<code>glossaries</code> package . . . . .	88
8.4.9	Index and the <code>tocbibind</code> package . . . . .	89
8.5	Math . . . . .	90
8.5.1	Rendering tradeoffs . . . . .	90

8.5.2	SVG option . . . . .	91
8.5.3	MathJax option . . . . .	91
8.5.4	ntheorem package . . . . .	93
8.5.5	siunitx package . . . . .	93
8.5.6	units and nicefrac packages . . . . .	93
8.5.7	newtxmath package . . . . .	94
8.6	Graphics . . . . .	94
8.6.1	grffile package . . . . .	95
8.6.2	color package . . . . .	95
8.6.3	xcolor package . . . . .	95
8.6.4	overpic package . . . . .	96
8.7	Tabular . . . . .	96
8.7.1	longtable package . . . . .	99
8.7.2	supertabular and xtab packages . . . . .	99
8.7.3	bigdelim package . . . . .	99
8.8	Floats . . . . .	100
8.8.1	float, trivfloat, and/or algorithmicx together . . . . .	100
8.8.2	caption and subcaption packages . . . . .	100
8.8.3	subfig package . . . . .	100
8.8.4	floatrow package . . . . .	100
8.8.5	keyfloat package . . . . .	101
8.9	Miscellaneous . . . . .	101
8.9.1	verse and memoir . . . . .	101
8.9.2	newclude package . . . . .	102
8.9.3	babel package . . . . .	102
<b>9</b>	<b>EPUB conversion</b>	<b>103</b>
<b>10</b>	<b>Word-processor conversion</b>	<b>105</b>
10.1	Activating word-processor conversion . . . . .	105
10.2	Additional modifications . . . . .	107
10.3	Recommendations . . . . .	108
10.4	Limitations . . . . .	109
<b>11</b>	<b>Modifying l warp</b>	<b>111</b>
11.1	Creating an l warp version of a package . . . . .	111
11.2	Adding a package to the l warp.dtx file . . . . .	112
11.3	Testing l warp . . . . .	112
11.4	Modifying l warpmk . . . . .	112
<b>12</b>	<b>Troubleshooting</b>	<b>114</b>
12.1	Using the l warp.sty package . . . . .	114
12.1.1	Debug tracing output . . . . .	116
12.2	Compiling the l warp.dtx file . . . . .	116

<b>1</b>	<b>l warp.sty</b>	<b>117</b>
<b>13</b>	<b>Implementation</b>	<b>117</b>
<b>14</b>	<b>Section depths and HTML headings</b>	<b>118</b>
<b>15</b>	<b>Source Code</b>	<b>119</b>
<b>16</b>	<b>Detecting the TeX Engine — pdflatex, lualatex, xelatex</b>	<b>120</b>
<b>17</b>	<b>Unicode Input Characters</b>	<b>120</b>
<b>18</b>	<b>Early package requirements</b>	<b>121</b>
<b>19</b>	<b>Operating-System portability</b>	<b>122</b>
19.1	Common portability code . . . . .	122
19.2	Unix, Linux, and Mac OS . . . . .	122
19.3	MS-Windows . . . . .	122
<b>20</b>	<b>Package options</b>	<b>123</b>
20.1	Conditional compilation . . . . .	126
<b>21</b>	<b>Misplaced packages</b>	<b>127</b>
<b>22</b>	<b>Required packages</b>	<b>132</b>
<b>23</b>	<b>Loading packages</b>	<b>136</b>
<b>24</b>	<b>File handles</b>	<b>139</b>
<b>25</b>	<b>Include a file</b>	<b>139</b>
<b>26</b>	<b>Copying a file</b>	<b>140</b>
<b>27</b>	<b>Debugging messages</b>	<b>141</b>
<b>28</b>	<b>HTML-conversion output modifications</b>	<b>142</b>
28.1	User-level controls . . . . .	142
28.2	Heading adjustments . . . . .	144
<b>29</b>	<b>Remembering original formatting macros</b>	<b>146</b>
<b>30</b>	<b>Accents</b>	<b>147</b>
<b>31</b>	<b>Configuration Files</b>	<b>149</b>
31.1	project_html.tex . . . . .	149
31.2	lwarpmk.conf . . . . .	149
31.3	project.lwarpmkconf . . . . .	150
31.4	l warp.css . . . . .	151

31.5	lwarf_sagebrush.css . . . . .	178
31.6	lwarf_formal.css . . . . .	183
31.7	sample_project.css . . . . .	187
31.8	lwarf.xdy . . . . .	188
31.9	lwarf_mathjax.txt . . . . .	188
31.10	lwarfmk option . . . . .	190
<b>32</b>	<b>Stacks</b>	<b>201</b>
32.1	Assigning depths . . . . .	202
32.2	Closing actions . . . . .	202
32.3	Closing depths . . . . .	203
32.4	Pushing and popping the stack . . . . .	203
<b>33</b>	<b>Data arrays</b>	<b>205</b>
<b>34</b>	<b>HTML entities</b>	<b>205</b>
<b>35</b>	<b>HTML filename generation</b>	<b>206</b>
<b>36</b>	<b>Homepage link</b>	<b>209</b>
<b>37</b>	<b>\PrintStack diagnostic tool</b>	<b>209</b>
<b>38</b>	<b>Closing stack levels</b>	<b>210</b>
<b>39</b>	<b>PDF pages and styles</b>	<b>210</b>
<b>40</b>	<b>HTML tags, spans, divs, elements</b>	<b>212</b>
40.1	Mapping L <sup>A</sup> T <sub>E</sub> X Sections to HTML Sections . . . . .	212
40.2	Babel-French . . . . .	212
40.3	HTML tags . . . . .	213
40.4	Block tags and comments . . . . .	216
40.5	Div class and element class . . . . .	217
40.6	Single-line elements . . . . .	218
40.7	HTML5 semantic elements . . . . .	218
40.8	High-level block and inline classes . . . . .	219
40.9	Closing HTML tags . . . . .	221
<b>41</b>	<b>Paragraph handling</b>	<b>221</b>
<b>42</b>	<b>Paragraph start/stop handling</b>	<b>225</b>
<b>43</b>	<b>Page headers and footers</b>	<b>227</b>
<b>44</b>	<b>CSS</b>	<b>228</b>
<b>45</b>	<b>HTML meta description and author</b>	<b>229</b>

<b>46</b>	<b>Footnotes</b>	<b>230</b>
46.1	Regular page footnotes . . . . .	230
46.2	Minipage footnotes . . . . .	230
46.3	Titlepage thanks . . . . .	230
46.4	Regular page footnote implementation . . . . .	231
46.5	Minipage footnote implementation . . . . .	232
46.6	Printing pending footnotes . . . . .	233
<b>47</b>	<b>Marginpars</b>	<b>234</b>
<b>48</b>	<b>Splitting HTML files</b>	<b>234</b>
<b>49</b>	<b>Sectioning</b>	<b>240</b>
49.1	User-level starred section commands . . . . .	240
49.2	Book class commands . . . . .	241
49.3	Sectioning support macros . . . . .	241
49.4	\section and friends . . . . .	248
<b>50</b>	<b>Starting a new file</b>	<b>250</b>
<b>51</b>	<b>Starting HTML output</b>	<b>252</b>
<b>52</b>	<b>Ending HTML output</b>	<b>255</b>
<b>53</b>	<b>Title page</b>	<b>257</b>
53.1	Setting the title, etc. . . . .	257
53.2	Changes for \affiliation . . . . .	259
53.3	Printing the thanks . . . . .	260
53.4	Printing the title, etc. in HTML . . . . .	260
53.5	Printing the title, etc. in print form . . . . .	261
53.6	\maketitle for HTML output . . . . .	262
53.7	\published and \subtitle . . . . .	265
<b>54</b>	<b>Abstract</b>	<b>266</b>
<b>55</b>	<b>Quote and verse</b>	<b>267</b>
55.1	Citations and attributions . . . . .	267
55.2	Quotes, quotations . . . . .	267
55.3	Verse . . . . .	268
<b>56</b>	<b>Verbatim and tabbing</b>	<b>269</b>
<b>57</b>	<b>Theorems</b>	<b>271</b>
<b>58</b>	<b>Lists</b>	<b>272</b>
58.1	Itemize . . . . .	273
58.2	Enumerate . . . . .	274
58.3	Description . . . . .	274

<b>59 Tabular</b>	<b>276</b>
59.1 Limitations . . . . .	276
59.2 Token lookahead . . . . .	279
59.3 Booleans . . . . .	279
59.4 Handling &, @, !, and bar . . . . .	280
59.4.1 Localizing & catcodes . . . . .	282
59.4.2 Handling & . . . . .	283
59.4.3 Filling an unfinished row . . . . .	284
59.5 Handling \\ . . . . .	285
59.6 Variables . . . . .	286
59.7 Parsing @, >, <, !, bar columns . . . . .	287
59.8 Parsing 'l', 'c', or 'r' columns . . . . .	290
59.9 Parsing 'p', 'm', or 'b' columns . . . . .	290
59.10 Parsing 'D' columns . . . . .	291
59.11 Parsing the column specifications . . . . .	291
59.12 Starting a new row . . . . .	295
59.13 Printing vertical bar tags . . . . .	296
59.14 Printing at or bang tags . . . . .	296
59.15 Data opening tag . . . . .	297
59.16 Midrules . . . . .	299
59.17 Multicolumns . . . . .	304
59.17.1 Parsing multicolumns . . . . .	304
59.17.2 Multicolumn factored code . . . . .	307
59.17.3 Multicolumn . . . . .	308
59.17.4 Longtable captions . . . . .	309
59.17.5 Counting HTML tabular columns . . . . .	311
59.18 Multicolumnrow . . . . .	312
59.19 Utility macros inside a table . . . . .	314
59.20 Special-case tabular markers . . . . .	314
59.21 Checking for a new table cell . . . . .	315
59.22 \mrowcell . . . . .	317
59.23 \mcolrowcell . . . . .	318
59.24 New \tabular definition . . . . .	318
59.25 Array . . . . .	323
<b>60 Cross-references</b>	<b>323</b>
60.1 Setup . . . . .	323
60.2 Zref setup . . . . .	325
60.3 Labels . . . . .	327
60.4 References . . . . .	328
60.5 Hyper-references . . . . .	330
<b>61 Floats</b>	<b>333</b>
61.1 Float captions . . . . .	334
61.1.1 Caption inside a float environment . . . . .	336
61.1.2 Caption and LOF linking and tracking . . . . .	336

<b>62</b>	<b>Table of Contents, LOF, LOT</b>	<b>339</b>
62.1	Reading and printing the TOC . . . . .	340
62.2	High-level TOC commands . . . . .	343
62.3	Side TOC . . . . .	343
62.4	Low-level TOC line formatting . . . . .	344
<b>63</b>	<b>Index and glossary</b>	<b>347</b>
<b>64</b>	<b>Restoring original formatting</b>	<b>349</b>
<b>65</b>	<b>Math</b>	<b>351</b>
65.1	Limitations . . . . .	351
65.1.1	Rendering tradeoffs . . . . .	351
65.1.2	SVG option . . . . .	352
65.1.3	MathJax option . . . . .	352
65.2	Inline and display math . . . . .	353
65.3	MathJax support . . . . .	356
65.4	Equation environment . . . . .	359
65.5	AMS Math environments . . . . .	360
65.5.1	Support macros . . . . .	360
65.5.2	Environment patches . . . . .	362
<b>66</b>	<b>Lateximages</b>	<b>369</b>
66.1	Description . . . . .	369
66.2	Support counters and macros . . . . .	369
66.3	Font size . . . . .	370
66.4	Sanitizing math expressions for HTML . . . . .	370
66.5	Equation numbers . . . . .	371
66.6	HTML <alt> tags . . . . .	373
66.7	lateximage . . . . .	374
<b>67</b>	<b>center, flushleft, flushright</b>	<b>377</b>
<b>68</b>	<b>Pre-loaded packages</b>	<b>378</b>
<b>69</b>	<b>Siunitx</b>	<b>379</b>
<b>70</b>	<b>Graphics print-mode modifications</b>	<b>379</b>
70.1	General limitations . . . . .	379
70.2	Print-mode modifications . . . . .	380
<b>71</b>	<b>Xcolor boxes</b>	<b>381</b>
<b>72</b>	<b>Cleveref</b>	<b>384</b>
<b>73</b>	<b>Picture</b>	<b>386</b>
<b>74</b>	<b>Boxes and Minipages</b>	<b>387</b>

74.1	Counters and lengths . . . . .	387
74.2	Footnote handling . . . . .	388
74.3	Minipage handling . . . . .	388
74.4	Parbox, makebox, framebox, fbox, raisebox . . . . .	392
75	<b>Direct formatting</b>	<b>397</b>
76	<b>Skips, spaces, font sizes</b>	<b>403</b>
77	<b>\phantomsection</b>	<b>411</b>
78	<b>\LaTeX{} and other logos</b>	<b>411</b>
78.1	HTML logos . . . . .	412
78.2	Print logos . . . . .	414
79	<b>\AtBeginDocument, \AtEndDocument</b>	<b>414</b>
80	<b>Trademarks</b>	<b>415</b>
<b>2</b>	<b>l warp-a4.sty</b>	<b>416</b>
81	<b>a4</b>	<b>416</b>
<b>3</b>	<b>l warp-a4wide.sty</b>	<b>416</b>
82	<b>a4wide</b>	<b>416</b>
<b>4</b>	<b>l warp-a5comb.sty</b>	<b>416</b>
83	<b>a5comb</b>	<b>416</b>
<b>5</b>	<b>l warp-abstract.sty</b>	<b>416</b>
84	<b>abstract</b>	<b>416</b>
<b>6</b>	<b>l warp-adjmulticol.sty</b>	<b>418</b>
85	<b>adjmulticol</b>	<b>418</b>
<b>7</b>	<b>l warp-addlines.sty</b>	<b>419</b>
86	<b>addlines</b>	<b>419</b>

<b>8 l warp-afterpage.sty</b>	<b>419</b>
87 afterpage	419
<b>9 l warp-algorithmicx.sty</b>	<b>420</b>
88 algorithmicx	420
<b>10 l warp-alltt.sty</b>	<b>421</b>
89 alltt	421
<b>11 l warp-amsthm.sty</b>	<b>421</b>
90 amsthm	421
<b>12 l warp-anonchap.sty</b>	<b>424</b>
91 anonchap	424
<b>13 l warp-any size.sty</b>	<b>424</b>
92 anysize	424
<b>14 l warp-appendix.sty</b>	<b>424</b>
93 appendix	424
<b>15 l warp-arabicfront.sty</b>	<b>425</b>
94 arabicfront	425
<b>16 l warp-authblk.sty</b>	<b>425</b>
95 authblk	425
<b>17 l warp-balance.sty</b>	<b>426</b>
96 balance	426

<b>18 l warp-bigdelim.sty</b>	<b>426</b>
97 <b>bigdelim</b>	<b>426</b>
<b>19 l warp-bigstrut.sty</b>	<b>428</b>
98 <b>bigstrut</b>	<b>428</b>
<b>20 l warp-bookmark.sty</b>	<b>428</b>
99 <b>bookmark</b>	<b>428</b>
<b>21 l warp-booktabs.sty</b>	<b>429</b>
100 <b>booktabs</b>	<b>429</b>
<b>22 l warp-boxedminipage.sty</b>	<b>429</b>
101 <b>boxedminipage</b>	<b>429</b>
<b>23 l warp-boxedminipage2e.sty</b>	<b>429</b>
102 <b>boxedminipage2e</b>	<b>429</b>
<b>24 l warp-caption2.sty</b>	<b>430</b>
103 <b>caption2</b>	<b>430</b>
<b>25 l warp-ccaption.sty</b>	<b>430</b>
104 <b>ccaption</b>	<b>430</b>
<b>26 l warp-changepage.sty</b>	<b>430</b>
105 <b>changepage</b>	<b>430</b>

<b>27 lwarf-chngpage.sty</b>	<b>431</b>
<b>106 chngpage</b>	<b>431</b>
<b>28 lwarf-chappg.sty</b>	<b>431</b>
<b>107 chappg</b>	<b>431</b>
<b>29 lwarf-color.sty</b>	<b>431</b>
<b>108 color</b>	<b>431</b>
<b>30 lwarf-crop.sty</b>	<b>431</b>
<b>109 crop</b>	<b>431</b>
<b>31 lwarf-cuted.sty</b>	<b>432</b>
<b>110 cuted</b>	<b>432</b>
<b>32 lwarf-cutwin.sty</b>	<b>432</b>
<b>111 cutwin</b>	<b>432</b>
<b>33 lwarf-dblfnote.sty</b>	<b>433</b>
<b>112 dblfnote</b>	<b>433</b>
<b>34 lwarf-dcolumn.sty</b>	<b>433</b>
<b>113 dcolumn</b>	<b>433</b>
<b>35 lwarf-draftwatermark.sty</b>	<b>434</b>
<b>114 draftwatermark</b>	<b>434</b>
<b>36 lwarf-ebook.sty</b>	<b>434</b>
<b>115 ebook</b>	<b>434</b>

<b>37 lwarf-ellipsis.sty</b>	<b>435</b>
116 ellipsis	435
<b>38 lwarf-emptypage.sty</b>	<b>435</b>
117 emptypage	435
<b>39 lwarf-endnotes.sty</b>	<b>435</b>
118 endnotes	435
<b>40 lwarf-enumerate.sty</b>	<b>436</b>
119 enumerate	436
<b>41 lwarf-enumitem.sty</b>	<b>436</b>
120 enumitem	436
<b>42 lwarf-epigraph.sty</b>	<b>437</b>
121 epigraph	437
<b>43 lwarf-eso-pic.sty</b>	<b>438</b>
122 eso-pic	438
<b>44 lwarf-everypage.sty</b>	<b>438</b>
123 everypage	438
<b>45 lwarf-everyshi.sty</b>	<b>439</b>
124 everyshi	439
<b>46 lwarf-extramarks.sty</b>	<b>439</b>
125 extramarks	439

<b>47 lwarf-fancybox.sty</b>	<b>440</b>
126 fancybox	440
<b>48 lwarf-fancyhdr.sty</b>	<b>444</b>
127 fancyhdr	444
<b>49 lwarf-fancyvrb.sty</b>	<b>444</b>
128 fancyvrb	444
<b>50 lwarf-figcaps.sty</b>	<b>450</b>
129 figcaps	450
<b>51 lwarf-fix2col.sty</b>	<b>450</b>
130 fix2col	450
<b>52 lwarf-float.sty</b>	<b>450</b>
131 float and \newfloat	450
<b>53 lwarf-floatflt.sty</b>	<b>452</b>
132 floatflt	452
<b>54 lwarf-floatpag.sty</b>	<b>453</b>
133 floatpag	453
<b>55 lwarf-floatrow.sty</b>	<b>453</b>
134 floatrow	453
<b>56 lwarf-flushend.sty</b>	<b>459</b>
135 flushend	459

<b>57 l warp-fncychap.sty</b>	<b>459</b>
<b>136 fncychap</b>	<b>459</b>
 <b>58 l warp-fnpos.sty</b>	<b>460</b>
<b>137 fnpos</b>	<b>460</b>
 <b>59 l warp-fontenc.sty</b>	<b>460</b>
<b>138 fontenc</b>	<b>460</b>
 <b>60 l warp-fontspec.sty</b>	<b>460</b>
<b>139 fontspec</b>	<b>460</b>
 <b>61 l warp-footmisc.sty</b>	<b>461</b>
<b>140 footmisc</b>	<b>461</b>
 <b>62 l warp-footnote.sty</b>	<b>462</b>
<b>141 footnote</b>	<b>462</b>
 <b>63 l warp-footnotehyper.sty</b>	<b>463</b>
<b>142 footnotehyper</b>	<b>463</b>
 <b>64 l warp-framed.sty</b>	<b>463</b>
<b>143 framed</b>	<b>463</b>
 <b>65 l warp-ftnright.sty</b>	<b>466</b>
<b>144 ftnright</b>	<b>466</b>
 <b>66 l warp-fullpage.sty</b>	<b>466</b>
<b>145 fullpage</b>	<b>466</b>

<b>67 lwarf-fullwidth.sty</b>	<b>466</b>
<b>146 fullwidth</b>	<b>466</b>
<b>68 lwarf-geometry.sty</b>	<b>467</b>
<b>147 geometry</b>	<b>467</b>
<b>69 lwarf-glossaries.sty</b>	<b>467</b>
<b>148 glossaries</b>	<b>467</b>
<b>70 lwarf-graphics.sty</b>	<b>469</b>
<b>149 graphics</b>	<b>469</b>
149.1 Graphics extensions . . . . .	469
149.2 Length conversions and graphics options . . . . .	469
149.3 Printing HTML styles . . . . .	472
149.4 \includegraphics . . . . .	472
149.5 Boxes . . . . .	477
<b>71 lwarf-graphicx.sty</b>	<b>480</b>
<b>150 graphicx</b>	<b>480</b>
<b>72 lwarf-grffile.sty</b>	<b>480</b>
<b>151 grffile</b>	<b>480</b>
<b>73 lwarf-hyperref.sty</b>	<b>480</b>
<b>152 hyperref</b>	<b>480</b>
<b>74 lwarf-hyperxmp.sty</b>	<b>485</b>
<b>153 hyperxmp</b>	<b>485</b>

<b>75 l warp-idxlayout.sty</b>	<b>485</b>
<b>154 idxlayout</b>	<b>485</b>
<b>76 l warp-indentfirst.sty</b>	<b>486</b>
<b>155 indentfirst</b>	<b>486</b>
<b>77 l warp-inputenc.sty</b>	<b>486</b>
<b>156 inputenc</b>	<b>486</b>
<b>78 l warp-keyfloat.sty</b>	<b>486</b>
<b>157 keyfloat</b>	<b>486</b>
<b>79 l warp-layout.sty</b>	<b>488</b>
<b>158 layout</b>	<b>488</b>
<b>80 l warp-letterspace.sty</b>	<b>488</b>
<b>159 letterspace</b>	<b>488</b>
<b>81 l warp-lettrine.sty</b>	<b>488</b>
<b>160 lettrine</b>	<b>488</b>
<b>82 l warp-lips.sty</b>	<b>489</b>
<b>161 lips</b>	<b>489</b>
<b>83 l warp-listings.sty</b>	<b>490</b>
<b>162 listings</b>	<b>490</b>

<b>84 l warp-longtable.sty</b>	<b>493</b>
<b>163 longtable</b>	<b>493</b>
<b>85 l warp-lscape.sty</b>	<b>495</b>
<b>164 lscape</b>	<b>495</b>
<b>86 l warp-ltcaption.sty</b>	<b>495</b>
<b>165 ltcaption</b>	<b>495</b>
<b>87 l warp-marginfit.sty</b>	<b>496</b>
<b>166 marginfit</b>	<b>496</b>
<b>88 l warp-marginfix.sty</b>	<b>496</b>
<b>167 marginfix</b>	<b>496</b>
<b>89 l warp-marginnote.sty</b>	<b>497</b>
<b>168 marginnote</b>	<b>497</b>
<b>90 l warp-mcaption.sty</b>	<b>497</b>
<b>169 mcaption</b>	<b>497</b>
<b>91 l warp-mdframed.sty</b>	<b>497</b>
<b>170 mdframed</b>	<b>497</b>
170.1 Package loading . . . . .	498
170.2 Limitations . . . . .	498
170.3 Color and length HTML conversion . . . . .	499
170.4 Environment encapsulation . . . . .	499
170.5 Titles and subtitles . . . . .	501
170.6 New environments . . . . .	503

<b>92 l warp-metalogo.sty</b>	<b>506</b>
<b>171 metalogo</b>	<b>506</b>
<b>93 l warp-microtype.sty</b>	<b>507</b>
<b>172 microtype</b>	<b>507</b>
<b>94 l warp-midfloat.sty</b>	<b>508</b>
<b>173 midfloat</b>	<b>508</b>
<b>95 l warp-moreverb.sty</b>	<b>508</b>
<b>174 moreverb</b>	<b>508</b>
<b>96 l warp-morewrites.sty</b>	<b>509</b>
<b>175 morewrites</b>	<b>509</b>
<b>97 l warp-mparhack.sty</b>	<b>510</b>
<b>176 mparhack</b>	<b>510</b>
<b>98 l warp-multicol.sty</b>	<b>510</b>
<b>177 multicol</b>	<b>510</b>
<b>99 l warp-multirow.sty</b>	<b>511</b>
<b>178 multirow</b>	<b>511</b>
178.1 Multirow . . . . .	512
178.2 Combined multicolumn and multirow . . . . .	513
<b>100 l warp-nameref.sty</b>	<b>514</b>
<b>179 nameref</b>	<b>514</b>

<b>101 l warp-needspace.sty</b>	<b>514</b>
<b>180 needspace</b>	<b>514</b>
<b>102 l warp-newinclude.sty</b>	<b>515</b>
<b>181 newinclude</b>	<b>515</b>
<b>103 l warp-newunicodechar.sty</b>	<b>515</b>
<b>182 newunicodechar</b>	<b>515</b>
<b>104 l warp-nextpage.sty</b>	<b>515</b>
<b>183 nextpage</b>	<b>515</b>
<b>105 l warp-nonumonpart.sty</b>	<b>516</b>
<b>184 nonumonpart</b>	<b>516</b>
<b>106 l warp-nopageno.sty</b>	<b>516</b>
<b>185 nopageno</b>	<b>516</b>
<b>107 l warp-nowidow.sty</b>	<b>516</b>
<b>186 nowidow</b>	<b>516</b>
<b>108 l warp-ntheorem.sty</b>	<b>517</b>
<b>187 ntheorem</b>	<b>517</b>
187.1 Limitations . . . . .	517
187.2 Options . . . . .	517
187.3 Remembering the theorem style . . . . .	518
187.4 HTML cross-referencing . . . . .	520
187.5 \newtheoremstyle . . . . .	520
187.6 Standard styles . . . . .	521
187.7 Additional objects . . . . .	523
187.8 Renewed standard configuration . . . . .	523
187.9 amsthm option . . . . .	524

187.10 Ending a theorem . . . . .	527
187.11 \NoEndMark . . . . .	527
187.12 List-of . . . . .	527
187.13 Symbols . . . . .	528
187.14 Cross-referencing . . . . .	528
<b>109 l warp-overpic.sty</b>	<b>528</b>
188 overpic	528
<b>110 l warp-pagenote.sty</b>	<b>529</b>
189 pagenote	529
<b>111 l warp-paralist.sty</b>	<b>529</b>
190 paralist	529
<b>112 l warp-parskip.sty</b>	<b>530</b>
191 parskip	530
<b>113 l warp-pdflandscape.sty</b>	<b>530</b>
192 pdflandscape	530
<b>114 l warp-pdfsync.sty</b>	<b>530</b>
193 pdfsync	530
<b>115 l warp-pfnote.sty</b>	<b>531</b>
194 pfnote	531
<b>116 l warp-placeins.sty</b>	<b>531</b>
195 placeins	531

<b>117 l warp-prelim2e.sty</b>	<b>531</b>
<b>196 prelim2e</b>	<b>531</b>
 <b>118 l warp-quotchap.sty</b>	<b>532</b>
<b>197 quotchap</b>	<b>532</b>
 <b>119 l warp-ragged2e.sty</b>	<b>533</b>
<b>198 ragged2e</b>	<b>533</b>
 <b>120 l warp-realscripts.sty</b>	<b>534</b>
<b>199 realscripts</b>	<b>534</b>
 <b>121 l warp-relsize.sty</b>	<b>534</b>
<b>200 relsize</b>	<b>534</b>
 <b>122 l warp-romanbar.sty</b>	<b>536</b>
<b>201 romanbar</b>	<b>536</b>
 <b>123 l warp-romanbarpagenumber.sty</b>	<b>536</b>
<b>202 romanbarpagenumber</b>	<b>536</b>
 <b>124 l warp-rotating.sty</b>	<b>536</b>
<b>203 rotating</b>	<b>536</b>
 <b>125 l warp-rotfloat.sty</b>	<b>537</b>
<b>204 rotfloat</b>	<b>537</b>
 <b>126 l warp-savetrees.sty</b>	<b>538</b>
<b>205 savetrees</b>	<b>538</b>

<b>127 l warp-scalefnt.sty</b>	<b>538</b>
206 scalefnt	538
<b>128 l warp-sectsty.sty</b>	<b>538</b>
207 sectsty	538
<b>129 l warp-setspace.sty</b>	<b>539</b>
208 setspace	539
<b>130 l warp-shadow.sty</b>	<b>540</b>
209 shadow	540
<b>131 l warp-showidx.sty</b>	<b>540</b>
210 showidx	540
<b>132 l warp-showkeys.sty</b>	<b>541</b>
211 showkeys	541
<b>133 l warp-sidecap.sty</b>	<b>541</b>
212 sidecap	541
<b>134 l warp-sidenotes.sty</b>	<b>542</b>
213 sidenotes	542
<b>135 l warp-siunitx.sty</b>	<b>543</b>
214 siunitx	543
<b>136 l warp-soul.sty</b>	<b>545</b>
215 soul	545

<b>137 l warp-stabular.sty</b>	<b>547</b>
<b>216 stabular</b>	<b>547</b>
<b>138 l warp-subfig.sty</b>	<b>547</b>
<b>217 subfig</b>	<b>547</b>
<b>139 l warp-supertabular.sty</b>	<b>553</b>
<b>218 supertabular</b>	<b>553</b>
<b>140 l warp-syntonly.sty</b>	<b>554</b>
<b>219 syntonly</b>	<b>554</b>
<b>141 l warp-tables.sty</b>	<b>555</b>
<b>220 tables</b>	<b>555</b>
<b>142 l warp-tabularx.sty</b>	<b>555</b>
<b>221 tabularx</b>	<b>555</b>
<b>143 l warp-tabulary.sty</b>	<b>556</b>
<b>222 tabulary</b>	<b>556</b>
<b>144 l warp-textarea.sty</b>	<b>556</b>
<b>223 textarea</b>	<b>556</b>
<b>145 l warp-textcomp.sty</b>	<b>557</b>
<b>224 textcomp</b>	<b>557</b>
224.1 Limitations . . . . .	557
224.2 Package loading . . . . .	557
224.3 Remembering original definitions . . . . .	557
224.4 HTML symbols . . . . .	558
224.4.1 pdf $\backslash$ TeX symbols . . . . .	558

224.4.2 Xe <sup>L</sup> T <sub>E</sub> X and Lua <sup>L</sup> T <sub>E</sub> X symbols . . . . .	558
224.5 HTML dicritics . . . . .	559
224.6 Inside a <i>lateximage</i> . . . . .	560
<b>146 l warp-textpos.sty</b>	<b>560</b>
225 <i>textpos</i>	560
<b>147 l warp-theorem.sty</b>	<b>561</b>
226 <i>theorem</i>	561
226.1 Remembering the theorem style . . . . .	561
226.2 CSS patches . . . . .	563
<b>148 l warp-threeparttable.sty</b>	<b>565</b>
227 <i>threeparttable</i>	565
<b>149 l warp-tikz.sty</b>	<b>566</b>
228 <i>tikz</i>	566
<b>150 l warp-titleps.sty</b>	<b>567</b>
229 <i>titleps</i>	567
<b>151 l warp-titleref.sty</b>	<b>570</b>
230 <i>titleref</i>	570
<b>152 l warp-titlesec.sty</b>	<b>570</b>
231 <i>titlesec</i>	570
<b>153 l warp-toclet.sty</b>	<b>572</b>
232 <i>titletoc</i>	572

<b>154 l warp-titling.sty</b>	<b>574</b>
<b>233 titling</b>	<b>574</b>
<b>155 l warp-toctbibind.sty</b>	<b>579</b>
<b>234 toctbibind</b>	<b>579</b>
<b>156 l warp-tocloft.sty</b>	<b>581</b>
<b>235 tocloft</b>	<b>581</b>
<b>157 l warp-transparent.sty</b>	<b>586</b>
<b>236 transparent</b>	<b>586</b>
<b>158 l warp-trivfloat.sty</b>	<b>587</b>
<b>237 trivfloat</b>	<b>587</b>
237.1 Combining \newfloat, \trivfloat, and algorithmicx . . . . .	588
<b>159 l warp-typearea.sty</b>	<b>588</b>
<b>238 typearea</b>	<b>588</b>
<b>160 l warp-ulem.sty</b>	<b>589</b>
<b>239 ulem</b>	<b>589</b>
<b>161 l warp-upref.sty</b>	<b>591</b>
<b>240 upref</b>	<b>591</b>
<b>162 l warp-verse.sty</b>	<b>591</b>
<b>241 verse</b>	<b>591</b>

<b>163 l warp-wallpaper.sty</b>	<b>593</b>
<b>242 wallpaper</b>	<b>593</b>
<b>164 l warp-wrapfig.sty</b>	<b>593</b>
<b>243 wrapfig</b>	<b>593</b>
<b>165 l warp-xcolor.sty</b>	<b>595</b>
<b>244 xcolor</b>	<b>595</b>
244.1 Limitations . . . . .	595
244.2 Xcolor definitions: location and timing . . . . .	595
244.3 Package loading . . . . .	597
244.4 Remembering and restoring original definitions . . . . .	598
244.5 HTML color style . . . . .	598
244.6 HTML border . . . . .	599
244.7 High-level macros . . . . .	599
<b>166 l warp-xfrac.sty</b>	<b>603</b>
<b>245 xfrac</b>	<b>603</b>
<b>167 l warp-xltextra.sty</b>	<b>605</b>
<b>246 xltextra</b>	<b>605</b>
<b>168 l warp-xmpincl.sty</b>	<b>606</b>
<b>247 xmpincl</b>	<b>606</b>
<b>169 l warp-xtab.sty</b>	<b>606</b>
<b>248 xtab</b>	<b>606</b>
<b>170 l warp-zwpagelayout.sty</b>	<b>608</b>
<b>249 zwpagelayout</b>	<b>608</b>

**Change History and Index** **610**

## List of Figures

1	tutorial.tex listing	55
---	----------------------	----

## List of Tables

1	l <sup>A</sup> T <sub>E</sub> X–HTML generation — l warp package — Supported functions	40
2	Required software programs	48
3	Files created along with the print version	57
4	Package options	74
5	Section HTML headings for word-processor conversion	109
6	Section depths and HTML headings	118
7	Tabular baseline	292
8	Tabular HTML column conversions	292
9	Cross-referencing data structures	324
10	Float data structures	333
11	AMSthm package — CSS styling of theorems and proofs	421
12	Ntheorem package — CSS styling of theorems and proofs	517
13	Theorem package — CSS styling of theorems and proofs	561

## 2 Updates

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

For a detailed list of changes, see the Change History on page 610.

### v0.42:

#### word-processor conversion

- ⚠ name change
- ⚠ name change

- 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 WPMarkFloats.
  - 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 L<sup>A</sup>T<sub>E</sub>X source for copy/paste into the LibreOffice Writer TeXMaths extension.
  - Improved formatting for numerous objects. See section 10.

#### tabbing

- Add: tabbing environment.

#### overpic

- Add: overpic package. See section 188.

#### math

- Fix: Text copy/paste of *AMS* math environment numbers and names.

- Improved \ensuremath.

- MathJax with siunitx: Updated script and documentation.

#### symbols

- textcomp: Improved \interrobangdown.

- realscripts: Fix for subscripts in a lateximage.

#### load order

- morewrites: Enforces loading before lwarf.

### 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 178.2 on page 513.

- Fix: Unfinished tabular rows are automatically filled.

- Fix for tabular column specifiers while using babel-french. (\NoAutoSpacing is activated then nullified inside the tabular, due to a conflict with the tabular column parsing code.)

**v0.40:****graphics, graphicx**

- graphics and graphicx have been moved from the l warp core, and are only loaded if requested with \usepackage.

**\includegraphics path**  
⚠️ image file extensions

- 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.

**bigdelim**  
**symbols**  
**fixes**

- grffile is now directly supported instead of emulated.

- Fix for bigdelim, and improved documentation. See section 97.

- Improved L<sup>A</sup>T<sub>E</sub>X and textcomp symbols.

- Fix for L<sup>A</sup>T<sub>E</sub>X logos and \InLineClass, etc. inside a lateximage.

- Fix for xltextra with X<sup>A</sup>T<sub>E</sub>X.

- Fixes for tocbibind with \simplechapter, etc.

- Fixes for \multicolumnrow and \nullfonts with older versions of multirow and xpars.

- Added \underline.

- Added adjmulticol.

- Added cuted, midfloat.

- Added pfnote, fnpos, dblfnote.

- Added stabular, tabls.

- Added sectsty, anonchap, quotchap.

**v0.39:****title pages**

- 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 178.2.

- 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.

⚠️ \published and  
\subtitle

⚠️ load order

tabular

multi column/row cell

⚠️ macros inside tabular

⚠️ tabular defined inside  
another environment

tabular

margins

- [page layout](#) • Added addlines, anysize, a4, a4wide, a5comb, textarea, zwpagelayout, typearea, ebook.

**v0.38:**

- [forced single-pass compile](#) • Added lwarfmk print1 and lwarfmk html1 actions to force a compile of the project a single time. Useful when multiple passes are not needed, or changes were not detected.
- [starred sections](#) • 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](#).
- [updated tutorial](#) • Modified the tutorial to use the new \ForceHTMLPage and \ForceHTMLTOC macros.
- [packages](#) • Added appendix, tocbibind, fncychap, fix2col.
- [font size](#) • Added relsize, scalefnt.
- [page numbering](#) • Added realscripts, metalogo, xltxtra.
- [front & back matter](#) • Added arabicfront, chappg, nonumonpart, nopageno, romanbarpagenumber.
- 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](#) • \include now maintains independent .aux files for HTML versions.
- [\texmkmk](#) • comment, used by lwarf, now maintains independent cut files for print and HTML versions, helping \texmkmk to better know whether to recompile.
- [accents and symbols](#) • Improved support for L<sup>A</sup>T<sub>E</sub>X accents, textcomp, siunitx symbols.
- [babel-french](#) • Improved babel-french handling for load order and ~ tilde.

**v0.36:**

- [boxes and frames](#) • Recorganized the documentation section regarding special cases and limitations. (Section [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 \teximage.
- [babel-french](#) • babel-french: Improvements for French variants, load order, footnotes, ellipses.

- footnotes**
    - Improved footnote numbering. `lateximage` footnotes now appear as regular footnotes to match the numbering of the print version. Also fixed a regression with MathJax.
  - siunitx**
    - Improved `siunitx` units.
    - Fix for filenames while using MathJax.
    - Fix for `\rule` when `xcolor` is not loaded.
    - Added `transparent`, `upref`.
- v0.35:** Fix: `\textbf` and related.
- v0.34:**
- ⚠ Optional arguments**
    - `BlockClass`'s optional argument has been moved in front of the mandatory argument:  
`BlockClass[style]{class}` (NEW)  
instead of:  
`BlockClass{class}[style]` (OLD)
    - This change makes it more consistent with `LATEX` standards, and avoids problems with space between arguments.
  - ⚠ Optional arguments**
    - Likewise, `\InlineClass`'s optional argument now comes before the mandatory arguments:  
`\InlineClass[style]{class}{text}`
- spans with minipages**
    - Improved compatibility between spans, minipages, lists, frames, and math. Handles minipages and lists inside an HTML span, such as an `\fbox` containing a minipage, although with minimal HTML fomating. 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 `lateximagees` and SVG math.
  - eqnarray**
    - Added the `eqnarray` environments.
  - verbatim packages**
    - `fancyvrb` is no longer required (preloaded), but is still supported.
    - Added `verbatim` and `moreverb`.
  - framing packages**
    - Added `fancybox`, `boxedminipage2e` and `shadow`.
  - list packages**
    - `enumitem` is no longer required, but is still supported.
    - Added `enumerate` and `paralist`.

**babel-french**

- `titleps` is no longer required, but is still supported.
- Added `crop`.
- Added `rotfloat`, `marginfit`, and several minor packages; see the change log.
- Adds fixed-width HTML spaces around punctuation when using `babel-french`. LuaTeX does not yet use the extra punctuation spacing.

**v0.33:**

- Tabular @ and ! columns now have their own HTML columns.
- & catcode changes are localized, perhaps causing errors about the tab alignment character &, so any definitions of macros or environments which themselves contain `tabular` and & must be enclosed within `\StartDefiningTabulars` and `\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:****⚠ l warp-newproject**

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

To modify existing documents, remove from the document source:

```
\usepackage{l warp-newproject}
```

The `l warp` package now produces the configuration files during print output, and also accepts the option `l warpmk` if desired.

**⚠ HTML setup changes.**

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

Old Macro	New Package Option
\HomeHTMLFileName	HomeHTMLfilename
\HTMLFileName	HTMLfilename
\useLatexmk	latexmk
\warpOSwindows	OSWindows
Old Package Option	New Package Option
lwarpmklang (new)	IndexLanguage xdyFilename
Old Macro	New Macro
\MetaLanguage	\HTMLLanguage
\HTMLAuthor	\HTMLAuthor
\NewHTMLDescription	\HTMLDescription
\SetFirstPageTop	\HTMLFirstPageTop
\SetPageTop	\HTMLPageTop
\SetPageBottom	\HTMLPageBottom
\NewCSS	\CSSfilename

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

```
\usepackage[
    HomeHTMLfilename=index,
    HTMLfilename={},
    IndexLanguage=english
]{lwarf}
```

- The file lwarf\_html.xdy has been renamed lwarf.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 lwarf.xdy.
  3. The old file lwarf\_html.xdy may be deleted.
- The new lwarf package option xdyFilename may be used to tell lwarpmk to use a custom .xdy file instead of lwarf.xdy. See section 7.10.
- Improvements in index processing:
  - xindy's language is now used for index processing as well as glossary.
  - Print mode without latexmk now uses xindy instead of makeindex.
  - texindy/xindy usage depends on pdflatex vs xelatex, lualatex.
  - For pdflatex and texindy, the -C utf8 option is used. This is supported in modern distributions, but a customized lwarpmk.lua may need to be created for use with older distributions.

**v0.29:**

- Add: `lwarpmklang` option for `lwarf-newproject` and `lwarf`. 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  $\text{\TeX}$  version. These `lwarp-pagename.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 40.)

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  $\text{\TeX}$  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  $\text{\TeX}$  code.

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

Category	Status
Engines:	<code>pdf<math>\text{\TeX}</math></code> , <code>X<math>\text{\TeX}</math></code> , <code>Lua<math>\text{\TeX}</math></code>
Classes:	book, report, or article. memoir is planned.
Page layout:	Emulates geometry, fancyhdr, titleps, addlines, any-size, a4, a4wide, a5comb, textarea, zwpagelayout, typearea, ebook.

## lwarf 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>anonchap</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> .
Languages:	<code>babel</code> . ( <code>polyglossia</code> is untested.)
Margin notes:	<code>marginfit</code> , <code>marginfix</code> .
Footnotes:	<code>footnote</code> , <code>footmisc</code> , <code>marginnote</code> , <code>sidenote</code> , <code>pagenote</code> , <code>endnotes</code> .
Math:	Supported. Converted to SVG images with <code>HTML</code> <code>&lt;alt&gt;</code> tags containing the <code>\TeX</code> source for the math expression. MathJax supported as an alternative. <code>\AMS</code> environments are supported. User-defined macros are available during conversion, due to native <code>\TeX</code> processing.
Theorems:	Support for native <code>\TeX</code> 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>

## lwarf 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, floatft.
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 mixing. \textcolor, \colorbox, \fcolorbox. Enhanced for HTML compatibility.
Lists:	Standard L <sup>A</sup> T <sub>E</sub> X environments are supported, along with enumitem, enumerate, paralist. Spacing is still controlled by css.
Environments:	Standard L <sup>A</sup> T <sub>E</sub> X 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.

## l warp Supported Functions — continued

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

## 4 Alternatives

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

### 4.1 Internet class

Cls **internet** The closest to lwarf in design principle is the `internet` class by Andrew Stacey (<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  $\text{\LaTeX}$  processing to produce a DVI file containing special commands, and then uses additional post-processing for the HTML conversion by way of numerous configuration files. In some cases, lwarf 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  $\text{\LaTeX}$  syntax into HTML. Search for each on CTAN (<http://ctan.org>).

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

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

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

Prog **LaTeXML** **LT<sub>E</sub>XML:** <http://dlmf.nist.gov/LaTeXML/>

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

Prog **LaTeX2HTML** **LT<sub>E</sub>X2HTML:** <http://www.latex2html.org/>  
and <http://ctan.org/pkg/latex2html>.

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

Finally, Glad $\text{\TeX}$  may used to directly insert  $\text{\TeX}$  math into HTML:

Prog Glad $\text{\TeX}$  **Glad $\text{\TeX}$ :** <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  $\text{\TeX}$  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  $\text{\TeX}$ -related features.

### Asciidoctor-Latex:

<http://www.noteshare.io/book/asciidoctor-latex-manual>

Prog Asciidoctor-LaTeX <https://github.com/asciidoctor/asciidoctor-latex>

## 4.5 Pandoc

Prog Pandoc

A markup system which also reads and writes  $\text{\TeX}$  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

Prog LibreOffice years in their abilities to represent math with a  $\text{\TeX}$ -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 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 InDesign

Prog 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 lwarf, 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 lwarf package

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

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

```
Enter ⇒ kpsewhich lwarf.sty
```

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

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

```
Enter ⇒ tlmgr install lwarf
```

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

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

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

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

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

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

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

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

This should be something like:

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

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

```
Enter ⇒ mktexlsr
```

— or —

```
Enter ⇒ texhash
```

Table 2: Required software programs

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

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

**$\text{\LaTeX}$ :** pdflatex, xelatex, or lualatex.

**The `lwarf` package:** This package.

**The `lwarpmk` utility:** Provided along with this package. This should be an operating-system executable in the same way that pdflatex or latexmk is. It is possible to have the lwarf package generate a local copy of `lwarpmk` called `lwarpmk.lua`. See table 3.

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

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

**latexmk:** Optionally used by `lwarpmk` to compile  $\text{\LaTeX}$  code. On a MiK<sub>T</sub>E<sub>X</sub> system, Perl may need to be installed first.

**pdfcrop:** Used to pull images out of the  $\text{\LaTeX}$  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 MiK<sub>T</sub>E<sub>X</sub>, start the program called MiK<sub>T</sub>E<sub>X</sub> 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 l warp package.
2. Download the zip archive l warp.zip into your own l warp directory.
3. Unpack l warp.zip.
4. Locate the contents l warp.dtx and l warp.ins
5. Create the documentation:

Enter ⇒ pdflatex l warp.dtx  
(several times)

6. Create the .sty files:

Enter ⇒ pdflatex l warp.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/l warp

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

<texlocal>/doc/local/l warp

9. Renew the cache:

Enter ⇒ mktexlsr  
— or —

Enter ⇒ texhash

Or, for Windows MiK<sub>T</sub>E<sub>X</sub>, start the program called MiK<sub>T</sub>E<sub>X</sub> Settings (Admin) and click on the button called Refresh FNDB.

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

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

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

## 5.2 Installing the lwarpmk utility

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

After the `lwarf` 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 MiK $\text{\TeX}$ , 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  $\text{\TeX}$  Live or MiK $\text{\TeX}$  system you may use

Enter  $\Rightarrow$  `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  $\text{\TeX}$  Live binaries:

Enter  $\Rightarrow$  `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  $\Rightarrow$  `ln -s <path to lwarpmk.lua> lwarpmk`

- (c) Make the link executable:

Enter  $\Rightarrow$  `chmod 0755 lwarpmk`

**Windows  $\text{\TeX}$  Live:** Create a new `lwarpmk.exe` file:

- (a) Locate the  $\text{\TeX}$  Live binaries as shown above for Unix.

- (b) In the binaries directory, make a *copy* of `runscript.exe` and call it `lwarpmk.exe`. This will call the copy of `lwarpmk.lua` which is in the `scripts` directory of the distribution.

**Windows MiK $\text{\TeX}$ :** 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 l warpmk.bat containing:

```
texlua "C:\Program Files\MiKTeX 2.9\scripts\l warp\l warpmk.texlua" %*
```

This will call the copy of l warpmk.lua which is in the scripts directory of the distribution.

### 5.2.1 Using a local copy of l warpmk

It is also possible to use a local version of l warpmk:

1. When compiling the tutorial in section 6, use the l warpmk option for the l warp package:

```
\usepackage[l warpmk]{l warp}
```

2. When the tutorial is compiled with pdflatex, the file l warpmk.lua will be generated along with the other configuration files.

3. l warpmk.lua may be used for this project:

#### Unix:

- (a) Make l warpmk.lua executable:

Enter ⇒ chmod 0755 l warpmk.lua

- (b) Compile documents with

Enter ⇒ ./l warpmk.lua html

Enter ⇒ ./l warpmk.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 l warpmk.lua html

Enter ⇒ texlua l warpmk.lua print

etc.

Or:

Enter ⇒ l warpmk html

Enter ⇒ l warpmk 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 48.

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

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

The TeX utilities xindy, latexmk, and pdfcrop may be 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/pdfcrop
```

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

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

**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 48.
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 lwarf document.

## 6.1 Starting a new project

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

#### **Copy from the documentation PDF:**

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

## **Copy from the lwarps documentation directory:**

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

```
Enter ⇒ texdoc -l lwarptutorial.txt
```

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

## ⚠️ Bad formatting!

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

3. Compile the project:  
Enter ⇒ `pdflatex tutorial.tex`  
(several times)  
(`xelatex` or `lualatex` may be used as well.)
  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 `lwrap` 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,  $\text{\LaTeX}$  program (`pdflatex`, `xelatex`, or `lualatex`), the filenames of the source code and `HTML` output, and whether the additional helper program `latexmk` will be used to compile the document.)

Figure 1: tutorial.tex listing

Note: There are two pages!

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

\documentclass{book}

\usepackage{iftex}

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

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

% --- 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{l warp_sagebrush.css}

\begin{document}

\maketitle % Or titlepage/titlingpage environment.

% An article abstract would go here.

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

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

\section{Some math}

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

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

\end{document}

```

---

Table 3: Files created along with the print version

**tutorial.pdf:** The PDF output from L<sup>A</sup>T<sub>E</sub>X. The print version of the document.

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

**Auxiliary files:** The usual L<sup>A</sup>T<sub>E</sub>X files .aux, .log, .out, .toc, .lof, .idx. When an HTML version of the document is created, \_html versions of the auxiliary files will also be generated.

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

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

**.css files:** l warp.css, l warp\_formal.css, l warp\_sagebrush.css These files are standard for l warp, and are not meant to be modified by the user.

**sample\_project.css:** An example of a user-customized css file, which may be used for project-specific changes to the l warp defaults.

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

**l warp\_mathjax.txt:** Inserted into the HTML files when MathJax is used to display math. This file should not be modified by the user.

**comment.cut:** A temporary file used by l warp to conditionally process blocks of text. This file may be ignored.

---

When the lwarpmk option is given to the l warp package:

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

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

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

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

## 6.2 Compiling the print version with `lwarpmk`

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

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

1. Re-compile the print version:

Enter ⇒ `lwarpmk print`

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

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

Enter ⇒ `lwarpmk print`

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

4. Force a recompile to occur.

Enter ⇒ `lwarpmk again`

Enter ⇒ `lwarpmk print`

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

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

Enter ⇒ `lwarpmk printindex`

6. Recompile again to include the index.

Enter ⇒ `lwarpmk print`

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

Enter ⇒ `lwarpmk print1`

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

---

<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 lwarf 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

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

1. Create the svg images:

Enter  $\Rightarrow$  `lwarfpmk limages`

Enter  $\Rightarrow$  `lwarfpmk 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 `lwarfpmk 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.

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

### Lots of files!

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

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

The styles provided by lwarf include:

`lwarf.css`: A default style if `\CSSFilename` is not used. This style is comparable to a plain L<sup>A</sup>T<sub>E</sub>X document. To set this style, you may use `\CSSFilename{lwarf.css}`, or no `\CSSFilename` call at all.

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

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

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

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

 **Rename it!** Note that `sample_project.css` is overwritten whenever lwarf 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 `lwarf.css`, and this entry may be changed to load `lwarf_formal.css` or `lwarf_sagebrush.css` if desired. Additional changes to the css may be made by making entries later in the `<project>.css` file.

## 6.7 Customizing the HTML output

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

**⚠ Placement!** 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}] \`

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.

Opt `latexmk`

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

Opt `mathsvg`

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

Opt `mathjax`

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

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

Ctr `tocdepth`

**tocdepth:** Sectioning depth of the table of contents. See section 14 for a list of `LATEX` stack depths.

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

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

It is recommended to set:

`SideTOCDepth = FileDepth`

or

`SideTOCDepth = FileDepth+1`



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

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

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



Bool **CombineHigherDepths**

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

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

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

**HTML filenames** Example HTML filenames:

**Numbered HTML nodes:**

Example: Homepage index.html, and node-1, node-2.<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, etc.

---

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

---

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

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

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

---

<sup>10</sup>See \SetHTMLFileName to number grouped by chapter, for example.

The css styles of the web pages are set by the \CSSFilename command. If \CSSFilename is not used, a default plain style is used to mimic printed L<sup>A</sup>T<sub>E</sub>X output. lwarp\_sagebrush.css is a semi-fancy colored style as shown in this tutorial. Change it to lwarp\_formal.css for a more formal look, or comment out the \CSSFilename command to see the default. \CSSFilename may be used before each file break to set the css for individual pages of the website.

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

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

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

**Placed in the document wherever necessary:**

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

Env \warpHTML	<b>warpHTML:</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.
\warpprintonly	\warpprintonly: { <i>contents</i> } A macro version of the warpprint environment.
\warpHTMLonly	\warpHTMLonly: { <i>contents</i> } A macro version of the warpHTML environment.

## 6.8 Using `latexmk`

`latexmk` is a L<sup>A</sup>T<sub>E</sub>X 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

Xe $\text{\TeX}$  or Lua $\text{\TeX}$  may be used instead of  $\text{\TeX}$ .

1. Remove the auxiliary files for the project:

Enter  $\Rightarrow$  lwarfmk 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 lwarfmk are modified to remember which  $\text{\TeX}$  engine was used. Xe $\text{\TeX}$  or Lua $\text{\TeX}$  will be used for future runs of lwarfmk.

3. To recompile the document:

Enter  $\Rightarrow$  lwarfmk print

-and-

Enter  $\Rightarrow$  lwarfmk html

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

## 6.10 Using a glossary

lwarf 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]{lwarf}
```

---

To process the glossary for the print version:

Enter ⇒ `lwarfmk printglossary`

To process the glossary for the HTML version:

Enter ⇒ `lwarfmk 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

lwarf uses pdftotext to convert PDF output into UTF-8-encoded text. This process requires that UTF-8 information be embedded in the PDF file, which 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>E</sub>T<sub>E</sub>X and Lua<sub>E</sub>T<sub>E</sub>X 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{lwarf}:

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

- For X<sub>E</sub>T<sub>E</sub>X or Lua<sub>E</sub>T<sub>E</sub>X:

#### Pkg fontspec

- fontspec and font choices

#### ligatures

lwarf sets the following to turn off T<sub>E</sub>X ligatures during the generation of HTML tags, and turn off common ligatures in regular text, since older browsers may not display them correctly and newer browsers can automatically re-create them.

---

```
\defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
\defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
\defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
```

---

- For pdflatex:

#### Pkg lmodern

- lmodern or other font-related packages

#### Pkg fontenc

- fontenc

#### Pkg inputenc

- inputenc

#### Pkg newunicodechar

- newunicodechar

File	<code>glyptounicode</code>	<ul style="list-style-type: none"> <li>- <code>\input glyptounicode.tex</code></li> <li>- <code>\input glyptounicode-cmr.tex%</code> from the <code>pdfx</code> package</li> <li>- <code>\pdfgentounicode=1</code></li> </ul>
Pkg	<code>cmap</code>	<ul style="list-style-type: none"> <li>- <code>cmap</code></li> </ul>
Pkg	<code>textcomp</code>	<ul style="list-style-type: none"> <li>- <code>textcomp</code></li> </ul>
Pkg	<code>microtype</code> <code>ligatures</code>	<ul style="list-style-type: none"> <li>- <code>microtype</code> is automatically used by lwarf to turn off f,q,t,TQ ligatures for the same browser-related reasons shown above. Also, the monospaced font is used during HTML tag generation to turn off <math>\TeX</math> ligatures.</li> </ul>

3. `\usepackage{lwarf}` (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

lwarf uses the `xindy` program to processes indexes.

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

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

For an older distribution of  $\TeX$ , 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.

## 7.2 lwarf package loading and options

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

Pkg	<code>lwarf</code>	Load the lwarf package immediately after the font and UTF-8 setup commands.
Opt	<code>warpprint</code>	Select the <code>warpprint</code> option to generate print output (default), or the <code>warpHTML</code> option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual <code>pdflatex</code> , etc. When lwarf 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{lwarf}</code> without any options to create a printed version, while <code>&lt;project&gt;.html.tex</code> will create an HTML version.
Opt	<code>mathsvg</code>	For math display, select <code>mathsvg</code> (default), or <code>mathjax</code> . For more information about
Opt	<code>mathjax</code>	

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 <code>xindy</code> language option used for index and glossary generation.
<code>latexmk</code>	Boolean for <code>lwarpmk</code> to use <code>latexmk</code> for compiling documents. Otherwise, <code>lwarpmk</code> attempts to recompile several times by itself.
<code>lwarpmk</code>	Generate a local copy of <code>lwarpmk.lua</code> .
<code>xdyFilename</code>	Tells <code>lwarpmk</code> to use a custom filename for <code>xindy</code> , instead of <code>lwarf.xdy</code> .

the math options, see section 8.5.

See table 4 for the full list of options.

### 7.3 Selecting the operating system

Prog Unix	<code>lwarf</code> 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 <code>lwarf</code> option <code>OSWindows</code> .
Prog Windows	
Opt OSWindows	When detected or specified, the operating-system path separator used by <code>lwarf</code> is modified, the boolean <code>usingOSWindows</code> is set true. This boolean may be tested by the user for later use.

## 7.4 Selecting actions for print or HTML output

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

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

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

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

---

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

---

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

---

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

---

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

---

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

---

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

`\warpprintonly {<actions>}`

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

`\warpHTMLonly {<actions>}`

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

## 7.5 Commands to be placed into the warpprint environment

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

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

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

Also see section 12: Troubleshooting.

## 7.6 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 lwarf.

\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.

## 7.7 HTML page meta descriptions

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

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

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

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

## 7.8 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.

## 7.9 CSS

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

---

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

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

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

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

---

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

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

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

## 7.10 Modifying xindy index processing

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

To customize index processing:

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

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

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

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

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

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

## 8 Special cases and limitations

Also see section 12: Troubleshooting.

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

### 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 \qquad.
- \, \, 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  $\text{\LaTeX}$  accents such as `\^o` will work, but many more kinds of accents are available when using Unicode-aware  $\text{\XeLaTeX}$  and  $\text{\LuaLaTeX}$ .

### 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

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

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

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

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

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

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

 **text alignment**

Nested minipages adopt their parent's text alignment in HTML, whereas in regular L<sup>A</sup>T<sub>E</sub>X PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

### 8.2.3 Side-by-side minipages

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

### 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 placed on their own line with a frame surrounding them. The contents preserve their HTML formatting, so lists and minipages look nicer, and valid HTML is created for a `tabular`. While an `\fbox` containing a `tabular` is valid L<sup>A</sup>T<sub>E</sub>X 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 inline `minipage` and lists:

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 244 for `xcolor`’s `\colorbox` and `\fcolorbox`, and `l warp`’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 `l warp` documentation for the `fancybox` package.

**To frame equations:** See section 126 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.

---

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

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

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

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

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

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

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

**indented alignment** LVerbatim, \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, lwarf loads mdframed in HTML with `framemethod=none`.

**font** For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

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

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

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

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

## 8.3 Cross-references

 **empty link** `\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.

### 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 `\pageref{pageFor}`, which defaults to “see”, followed by a hyperlink to the appropriate object.

Ex:

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

\pageref{PageFor} may be redefined to “page for”, empty, etc. See page 330.

### 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 386 to redefine the message which is printed for page number references.

### 8.3.3 Hyperlinks, hyperref, and url

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

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

### 8.3.4 Footnotes and page notes

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

#### ⚠ pfnote numbers

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

## 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, tocbibind, and the index. See the following sections where

applicable. Continue here if interested in the reason for adding these commands to lwarf.

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.

lwarf 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*`. lwarf 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.

`\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` lwarf supports the native L<sup>A</sup>T<sub>E</sub>X 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

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

#### 8.4.5 **appendix** package

`\usepackage{appendix}` 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

`\usepackage{pagenote}` `pagenote` works as-is, but the `page` option is disabled.

#### 8.4.7 **endnotes** package

`\usepackage{endnotes}` To place the endnotes in the TOC, use:

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

`\usepackage[HTML]{endnotes}` To additionally have the endnotes on their own HTML page, if `FileDepth` allows:

```
\ForceHTMLPage
\theendnotes
```

#### 8.4.8 **glossaries** package

`\usepackage[glossaries]{xindy}` `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 `\nameref`s in HTML, which could become a very long string if many items are referenced. For now, the number list is simply turned off.

`\usepackage[placement=bottom, toc=none]{glossaries}` 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 lwarf 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`.

**8.4.9 Index and the `tocbibind` package**

**Opt IndexLanguage** The lwarf 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  $\text{\TeX}$  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  $\text{\TeX}$  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      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 234 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 **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 **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 **LATEX** code which generated the math, allowing copy/paste of the **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 **LATEX** code for the math expression. For **AMS** environments with multiple numbers in the same environment, only the first and last is copy/pasted, as a range. No tags are listed inside a starred **AMS** environment, although the `\tag` macro will still appear inside the **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.

Prog MathJax

When MathJax is enabled, math is rendered twice:

- As regular **LATEX** PDF output placed inside an **HTML** comment, allowing equation numbering and cross referencing to be almost entirely under the control of **LATEX**, and

---

<sup>11</sup>See section 245 regarding fonts and fractions.

2. As detokenized printed  $\text{\LaTeX}$  commands placed directly into the  $\text{HTML}$  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,  $\backslash\tag{}$  tagged equations have the chapter number prepended in  $\text{HTML}$  output, unlike  $\text{\LaTeX}$ .  $\backslash\tag*$  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.

- 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:



**siunitx inside an equation**



**other macros and packages**

---

```
\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  $\text{\TeX}$  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** 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{lwarf}
...
\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:**

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

For print output, lwarf 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

⚠ 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 L<sup>A</sup>T<sub>E</sub>X, 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 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.

### 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 `\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.

#### 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, lwarf's emulation of the `tabular` does not automatically resume when the containing environment ends, resulting in corrupted HTML rows. To fix this, use `\ResumeTabular` as follows. This is ignored in print mode.

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

#### Cell contents:

⚠ `paragraphs`

- 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 `l warp`, so `\warpprintonly` and `\warpHTMLonly` must be used to make versions for print and HTML.

- See section [178.2](#) for `\multicolumnrow`.

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

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

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}[0pt]{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`

- `*` 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.

\multicolumn

- In \multicolumn cells, the print version may have extra instances of <, >, @, and ! cells on the second and later rows in the \multicolumn which do not appear in the HTML version.
- \newcolumntype is ignored; unknown column types are set to l.

⚠ \newcolumntype**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{}.

- For S columns (from the siunitx package), while producing print output, anything non-numeric must be placed inside {} braces, including commands such as \multicolumn. 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 & {\multicolumn{2}{2cm}{Text}} & 3 \\}
\warpHTMLonly{1 & 2 & \multicolumn{2}{2cm}{Text} & 3 \\}
```

⚠ S columns

### 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
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 237.1.

### 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.

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

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

### 8.8.5 keyfloat package

**⚠ keywrap** If placing a \keyfig[H] inside a keywrap, use an absolute width for \keyfig, instead of 1w-proportional widths. (The [H] option forces the use of a minipage, which internally adjusts for a virtual 6-inch wide minipage, which then corrupts the 1w option.)

## 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. l warp provides \attribution, which works for both print and HTML output. To combine the two so that \attrib is used for print and \attribution is used for HTML:

---

```
\begin{warpHTML}
\let\attrib\attribution
\end{warpHTML}
```

---

Len \leftskip  
 Len \leftmargini  
 Len \TMLleftskip  
 Len \TMLleftmargini 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 \HTMLleftskip 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.

### 8.9.2 newclude package

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

*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 the user's changes do not affect the HTML output.

⚠ `customized spacing`

## 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:

*FormatEPUB*

---

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

---

To help convert lwarp HTML output to EPUB, add

\booltrue{FormatEPUB}

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

Enter ⇒ lwarpmk cleanall

Enter ⇒ lwarpmk html

Enter ⇒ lwarpmk limages

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

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

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

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

Preferences → Plugins → File type plugins → HTML to Zip

Check the box Add linked files in breadth first order.

### section breaks

The EPUB-conversion program must also know where the section breaks are located. For a list of lwarp's section headings, see table 6. For example, an article class document would break at \section, which is mapped to HTML heading level <h4>, whereas a book class document would break at \chapter, which is HTML heading level <h3>. For Calibre, this option is found in

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

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

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:

---

*FormatWP*

---

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

---

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

`\booltrue{FormatWP}`

to the project's source preamble after lwarp is loaded. The following changes are 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 file split. These will have to be manually moved to their proper place in the document. lwarp does not know where the page breaks will be in the word processor's document, so the footnotes are simply moved to the end of each sectional break.
- Forces single-file output.

- Turns off HTML debugging comments. These are comments appearing inside the HTML code, marking the opening/closing of sections and <div>s, but they are no longer useful when the document has been imported into a word processor.
- An additional <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, floatfl, and wrapfig.
  - fancybox \shadowbox, etc.
  - The L<sup>A</sup>T<sub>E</sub>X and T<sub>E</sub>X logos.
- Honors several booleans:

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

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

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

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

**WPMarkMath:** Prints L<sup>A</sup>T<sub>E</sub>X 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*

---

Default false. Adds

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

Bool WPMarkFloats

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

```

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

---

---

### *WPMarkMinipages*

---

Default false. Adds

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

```

Bool WPMarkMinipages

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

---

---

### *WPMarkTOC*

---

Default true. While formatting for word processors, adds

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

```

Bool WPMarkTOC

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*

Default **false**. While formatting for word processors, adds

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

Bool WPMarkLOFT

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*

Default **false**. While formatting for word processors, prints math as  $\text{\LaTeX}$  code instead of creating SVG images or MathJax. This is useful for cut/paste into the LibreOffice Writer TeXMaths extension.

Bool WPMarkMath

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*

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

See table 5 on page 109.

### 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 5: Section HTML headings for word-processor conversion

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

\* For default depths when not FormatWP, see table 6 on page 118.

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

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  $\text{HTML}$  versions.

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

As of when last checked, LibreOffice ignores the following:

- Minipage alignment.
- Tabular cell vertical alignment.
- Image rotation and scaling.
- Rounded border corners, which are also used by:
  - $\text{\texttt{\textbackslash textcircled{}}}$
  - booktabs trim
- $\text{\texttt{\textbackslash hspace}}$  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 l warp

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 `\latextimage` then displayed with an image of the resulting  $\text{\TeX}$  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 l warp version of a package

When creating HTML, l warp redefines the `\usepackage` and `\RequirePackage` macros such that it first looks to see if a `l warp-<packagename>.sty` version exists. If so, the l warp version is used instead. This modular system allows users to create their own versions of packages for l warp to use for HTML, simply by creating a new package with a `l warp-` prefix. If placed in the local directory along with the source code, it will be seen by that project alone. If placed alongside the other `l warp-` packages where  $\text{\TeX}$  can see it, then the user's new package will be seen by any documents using l warp. (Remember `mktexlsr` or `texhash`.)

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

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

Each `l warp-` package should first call either

```
\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 187 (ntheorem) for an example of selectively disabling user options for a package. Use this when HTML output only requires some modifications of the original package. For a case where the original package is usable without changes, there is no need to create a lwarp- version.

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

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

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

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 l warp . sty package

Also see:

Section 7.5: 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 6.7.

**Tabular problems:** See section 8.7.

#### Obscure error messages:

- Be sure that a print version of the document compiles and that your document's L<sup>A</sup>T<sub>E</sub>X code is correct, before attempting to generate an HTML version.

**"Leaders not followed by proper glue":** This can be caused by a missing `\@floattype` or `\@sectiontype` definition. See l warp's definitions for examples.

**"Improper \prevdepth":** Something tried to use `\ensuremath` where l warp then tries to create a `\teximage`. If you can locate what used `\ensuremath`, have it temporarily set:  
`\LetLtxMacro{\ensuredmath}{\LWR{\origensuredmath}}`  
inside a group first.

#### Missing sections:

- See section 6.7 regarding the `FileDepth` and `SideTOCDepth` counters, and the use of `\tableofcontents` in the home page.

#### Missing HTML files:

- See the warning regarding changes to the HTML settings at section 6.7.

#### Missing / incorrect cross-references:

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

 empty link

 cleveref page numbers

- `\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 386 to redefine the message which is printed for page number references.

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

Use X<sub>EL</sub>T<sub>E</sub>X or Lua<sub>EL</sub>T<sub>E</sub>X.

**FLOATS OUT OF SEQUENCE:**

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

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

**Print document contains HTML tags:**

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

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

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

**Images are appearing in strange places:**

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

**SVG images:**

 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.

Expressing math as `svg` images has the advantage of representing the math exactly as `EL`T<sub>E</sub>X would, but has the disadvantage of requiring an individual

- ⚠️ Lots of files! 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 6.7.
- Verify that the proper css is actually being used.
- The browser may compensate for some subtle changes, such as automatically generating ligatures, reflowing text, etc.

**Un-matched conditional compiles:**

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

### 12.1.1 Debug tracing output

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

Place `\tracingl warp` just after `\usepackage{l warp}` to activate tracing.

## 12.2 Compiling the l warp .dtx file

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

**Illogical error messages caused by an out-of-sync l warp .sty file:**

1. Delete the `lwarf.sty` file.
2. `pdflatex lwarf.ins` to generate a new `lwarf.sty` file.
3. `pdflatex lwarf.dtx` to recompile the `lwarf.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 **lwarf.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 erude haëks clever tricks. Reference sources are given for many of the solutions, and a quick internet search will provide additional possibilities.

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

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

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

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

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

Table 6: Section depths and HTML headings

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

\* If `FormatWP` is true, section headings may be adjusted, depending on `WPTitleHeading`. See table 5 on page 109.

## 14 Section depths and HTML headings

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

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

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

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

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

## 15 Source Code

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

The following sections document the actual implementation of the lwarf package.

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

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

**objects** Black-colored tags in the left margin are used to identify programming objects such as files, packages, environments, booleans, and counters. Items without a tag are

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

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

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

## 16 Detecting the $\text{\TeX}$ Engine — pdflatex, lualatex, xelatex

```

1 \RequirePackage{iftex}
2
3 \ifLuaTeX
4 \RequirePackage{lualatex85}% until the geometry package is updated
5 \fi

```

## 17 Unicode Input Characters

**for HTML & PRINT:** If using pdflatex, convert a minimal set of Unicode characters. Additional characters may be defined by the user, as needed.

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

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

```

6
7 \RequirePackage{newunicodechar}
8
9 \newunicodechar{x}{\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 PDFT $\text{\TeX}$ , 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 <code>OSWindows</code> .
Prog Windows	
Opt OSWindows	When detected or specified, the operating-system path separator used by lwarp is modified, the boolean <code>usingOSWindows</code> is set true. This boolean may be tested by the user for later use.

### 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:

```
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{l warp}{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{l warp}{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 `l warp.xdy`. A customized file should be based on `l warp.xdy`, and must retain the line

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

```
65 \DeclareStringOption[l warp.xdy]{xdyFilename}
```

Opt **lwarpmk** Tells `l warp` 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.

```
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.

```
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 \ifcsempty{\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 l warpmk. 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 l warp, and others should only be loaded after.

Packages which should only be loaded before l warp have their own

`l warp-<packagename>.sty`

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

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

```

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

```

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

```

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 {\langle badpackagename\rangle} {\langle replacementpkgnname\rangle}

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

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

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

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

Detects XeTeX and LuaTeX:

```
304 \RequirePackage{ifutex}
305 \newif\ifxetexorluatex
306 \ifXeTeX
307     \xetexorluatextrue
308 \else
309     \ifLuaTeX
310         \xetexorluatextrue
311     \else
312         \xetexorluatexfalse
313     \fi
314 \fi
```

```
315 \end{warpall}
```

**for HTML output:** 316 `\begin{warpHTML}`

```
317 \ifxetexorluatex
318 % ^~A \usepackage[no-math]{fontspec}
```

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

```
319 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
320 \defaultfontfeatures[\sfamily]{Ligatures={NoCommon,TeX}}
321 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
322 \else
```

**pdflatex only:** Only pre-loaded if `pdflatex` is being used.

Pkg `microtype`

**ligatures** Older browsers don't display ligatures. Turn off letter ligatures, keeping L<sup>A</sup>T<sub>E</sub>X dash and quote ligatures, which may fail on older browsers but at least won't corrupt written words.

```

323 \RequirePackage {microtype}
324
325 \microtypesetup{
326   protrusion=false,
327   expansion=false,
328   tracking=false,
329   kerning=false,
330   spacing=false}
331
332 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}

333 \fi

334 \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:** 335 \begin{warpHTML}
336 \RequirePackage[paperheight=190in,paperwidth=20in,%
337 left=2in,right=12in,%
338 top=1in,bottom=1in,%
339 ]{geometry}
340 \twosidefalse
341 \mparswitchfalse
342 \end{warpHTML}

**for HTML & PRINT:** 343 \begin{warpall}

Pkg `xparse`

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

```
344 \RequirePackage{xparse}
```

Pkg **afterpackage** Used to patch titling to add \AddSubtitlePublished.

345 \RequirePackage{afterpackage}

346 \end{warpall}

**for HTML output:** 347 \begin{warpHTML}

Pkg **expl3**

LaTeX3 programming

348 \RequirePackage{expl3}

Pkg **gettitlestring**

Used to emulate \nameref.

349 \RequirePackage{gettitlestring}

Pkg **everyhook**

everyhook is used to patch paragraph handling.

350 \RequirePackage{everyhook}

351 \end{warpHTML}

**for HTML & PRINT:** 352 \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>

353 \RequirePackage{filecontents}

354

355 \@ifpackagelater{filecontents}{2011/10/09} %

356 {}

357 {

358 \newwrite\fcwrite

359 \let\LWR@origfilec@ntents\filec@ntents

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

361 }

362 \end{warpall}

**for HTML output:** 363 \begin{warpHTML}

Pkg xifthen

364 \RequirePackage{xifthen}

Pkg xstring

365 \RequirePackage{xstring}

Pkg xstring

366 \RequirePackage{verbatim}

Pkg makeidx

367 \RequirePackage{makeidx}

368 \makeindex

Pkg calc

369 \RequirePackage{calc}

Pkg refcount

370 \RequirePackage{refcount}

Pkg newfloat

371 \RequirePackage{newfloat}

Pkg caption

372 \RequirePackage{caption}

373 \end{warpHTML}

**for HTML & PRINT:** 374 \begin{warpall}

Pkg environ

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

375 \RequirePackage{environ}

376 \end{warpall}

**for HTML output:** 377 `\begin{warpHTML}`

Pkg `zref`

Used for cross-references.

378 `\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.

379 `\PassOptionsToPackage{leqno}{amsmath}`

380 `\RequirePackage{amsmath}`

Used to convert lengths for image width/height options.

381 `\RequirePackage{printlen}`

382 `\end{warpHTML}`

## 23 Loading packages

**for HTML output:** 383 `\begin{warpHTML}`

Remember the original `\RequirePackage`:

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

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

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

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

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

386 `\newcommand*\LWR@findword[3][,]{%`  
 387       `\StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%`  
 388       `}`

\LWR@lookforpackagename {*index*} If this is a package name, re-direct it to the l warp version by renaming it l warp- followed by the original name.

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

Find the n'th package name from the list:

```
390 \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.

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

See if the package name was found:

```
392 \IfStrEq{\LWR@strresulttwo}{}%
393 {}% no filename
394 {}% yes filename
```

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

```
395 \IfFileExists{l warp-\LWR@strresulttwo.sty}%
396 {}% l warp-* file found
397     \StrSubstitute%
398     {\LWR@requirepackagenames}%
399     {\LWR@strresult}%
400     {l warp-\LWR@strresulttwo}[\LWR@requirepackagenames]%
401 }%
402 {}% no l warp-* file
403 {}% yes filename
404 }
```

\RequirePackage [*1: options*] [*2: package names*] [*3: version*]

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

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

Redirect up to nine names:

```
406 \renewcommand*{\LWR@requirepackagenames}{#2}
407 \LWR@lookforpackagename{1}
408 \LWR@lookforpackagename{2}
409 \LWR@lookforpackagename{3}
410 \LWR@lookforpackagename{4}
```

```

411 \LWR@lookforpackagename{5}
412 \LWR@lookforpackagename{6}
413 \LWR@lookforpackagename{7}
414 \LWR@lookforpackagename{8}
415 \LWR@lookforpackagename{9}

```

\RequirePackage depending on the options and version:

```

416 \IfValueTF{#1}
417 {%
418   \IfValueTF{#3}{%
419     {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}{#3}}
420     {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}
421   }
422 {%
423   \IfValueTF{#3}{%
424     {\LWR@origRequirePackage{\LWR@requirepackagenames}{#3}}
425     {\LWR@origRequirePackage{\LWR@requirepackagenames}}
426   }
427 }
428 \LetLtxMacro{\usepackage}{\RequirePackage}

```

\LWR@ProvidesPackagePass {*<pkgname>*} [*<version>*]

Uses the original package, including options.

```

429 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}%
430 {%
431   \PackageInfo{lwarf}{Using package '#1' and adding lwarf modifications, including options,}%
432   {\ProvidesPackage{lwerp-#1}[#2]}
433   {\ProvidesPackage{lwerp-#1}}
434   \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
435   \ProcessOptions\relax
436
437 \IfValueTF{#2}
438 {\LWR@origRequirePackage[#1][#2]}
439 {\LWR@origRequirePackage[#1]}
440 }

```

\LWR@ProvidesPackageDrop {*<pkgname>*} [*<version>*]

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

```

441 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}%
442 {%
443   \PackageInfo{lwarf}{Replacing package '#1' with the lwarf version, discarding options,}%
444   {\ProvidesPackage{lwerp-#1}[#2]}

```

```
445 {\ProvidesPackage{lwarp-\#1}}
446 \DeclareOption*{}
447 \ProcessOptions\relax
448 }

449 \end{warpHTML}
```

## 24 File handles

Defines file handles for writes.

**for HTML & PRINT:** 450 \begin{warpall}

\LWR@quickfile For quick temporary use only. This is reused in several places.

```
451 \newwrite\LWR@quickfile%
452 \end{warpall}
```

**for HTML output:** 453 \begin{warpHTML}

\LWR@lateximagesfile For `lateximages.txt`.

```
454 \newwrite\LWR@lateximagesfile
455 \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 L<sup>A</sup>T<sub>E</sub>X.

**for HTML output:** 456 \begin{warpHTML}

```
\include {\langle filename\rangle}

\@include {\langle filename\rangle} Modified to load _html_inc files.

457 \def\@include#1 {%
458 \immediate\openout\LWR@quickfile #1_html_inc.tex% new
459 \immediate\write\LWR@quickfile{\string\input{\#1.tex}}% new
460 \immediate\closeout\LWR@quickfile% new
461 \LWR@origclearpage% \changed
462 \if@filesw
463   \immediate\write\@mainaux{\string\@input{\#1_html_inc.aux}}% changed
464 \fi
465 \@tempswatru
466 \if@partsw
467   \tempswafalse
468   \edef\reserved@b{\#1}%
469   \for\reserved@a:=\partlist\do
470     {\ifx\reserved@a\reserved@b\@tempswatru\fi}%
471 \fi
472 \if@tempswa
473   \let\@auxout\@partaux
474   \if@filesw
475     \immediate\openout\@partaux #1_html_inc.aux % changed
476     \immediate\write\@partaux{\relax}%
477   \fi
478   \input{\#1_html_inc.tex}% changed
479   \LWR@origclearpage% changed
480   \writeckpt{\#1}%
481   \if@filesw
482     \immediate\closeout\@partaux
483   \fi
484 \else
485   \deadcycles{z@}
486   \nameuse{cp@\#1}%
487 \fi
488 \let\@auxout\@mainaux%
489 }

490 \end{warpHTML}
```

## 26 Copying a file

**for HTML output:** 491 \begin{warpHTML}

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

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

```

492 \newwrite\LWR@copyoutfile % open the file to write to
493 \newread\LWR@copyinfile    % open the file to read from
494
495 \newcommand*\LWR@copyfile}[2]{%
496   \immediate\openout\LWR@copyoutfile=#2
497   \openin\LWR@copyinfile=#1
498   \begingroup\endlinechar=-1
499   \makeatletter
500   \loop\unless\ifeof\LWR@copyinfile
501     \read\LWR@copyinfile to\LWR@fileline % Read one line and store it into \LWR@fileline
502     \LWR@fileline\par % print the content into the pdf
503   % print the content:
504   \immediate\write\LWR@copyoutfile{\unexpanded\expandafter{\LWR@fileline}}%
505   \repeat
506   \closeout\LWR@copyoutfile
507 \endgroup
508 }

509 \end{warpHTML}

```

## 27 Debugging messages

**for HTML & PRINT:** 510 \begin{warpall}

Bool LWR@tracinglwarp True if tracing is turned on.

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

\tracinglwarp Turns on the debug tracing messages.

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

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

```

513 \newcommand{\LWR@traceinfo}[1]{%
514   \ifbool{LWR@tracinglwarp}{%
515     \typeout{*** lwarp: #1}%
516     \PackageInfo{lwarp}{#1 : }%
517   }{%
518   }%
519 }%
520 }
```

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

```
521 \newbool{HTMLDebugComments}
522 \boolfalse{HTMLDebugComments}
```

If `\tracinglwarp`, show where preamble hooks occur:

```
523 \AfterEndPreamble{
524 \LWR@traceinfo{AfterEndPreamble}
525 }
526
527 \AtBeginDocument{
528 \LWR@traceinfo{AtBeginDocument}
529 }

530 \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:** 531 `\begin{warpall}`

### 28.1 User-level controls

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

```
532 \newbool{FormatEPUB}
533 \boolfalse{FormatEPUB}
```

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.

```
534 \newbool{FormatWP}
535 \boolfalse{FormatWP}
```

Bool `WPMarkFloats` Default false. Adds

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

```

or

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

around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames and captions.<sup>12</sup>

```
536 \newbool{WPMarkFloats}
537 \boolfalse{WPMarkFloats}
```

Bool WPMarkMinipages Default false. Adds

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

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

```
538 \newbool{WPMarkMinipages}
539 \boolfalse{WPMarkMinipages}
```

Bool WPMarkTOC Default true. While formatting for word processors, adds

```
==== 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.*

```
540 \newbool{WPMarkTOC}
541 \booltrue{WPMarkTOC}
```

Bool WPMarkLOFT Default false. While formatting for word processors, adds

```
==== 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.*

```
542 \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.

---

543 \boolfalse{WPMarkLOFT}

Bool WPMarkMath Default false. While formatting for word processors, prints math as L<sup>E</sup>T<sub>E</sub>X code instead of creating SVG images or MathJax. This is useful for cut/paste into the LibreOffice Writer TeXMaths extension.

544 \newbool{WPMarkMath}  
545 \boolfalse{WPMarkMath}

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

See table 5 on page 109.

546 \newbool{WPTitleHeading}  
547 \boolfalse{WPTitleHeading}

548 \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:**

549 \begin{warpHTML}  
  
550 \AtBeginDocument{  
551 \ifbool{FormatWP}{  
552 \@ifundefined{chapter}{  
553 \ifbool{WPTitleHeading}{% part and section starting at h2  
554 \renewcommand\*\{\LWR@tagtitle\}{h1}  
555 \renewcommand\*\{\LWR@tagtitleend\}{/h1}  
556 \renewcommand\*\{\LWR@tagpart\}{h2}  
557 \renewcommand\*\{\LWR@tagpartend\}{/h2}  
558 \renewcommand\*\{\LWR@tagsection\}{h3}  
559 \renewcommand\*\{\LWR@tagsectionend\}{/h3}  
560 \renewcommand\*\{\LWR@tagsubsection\}{h4}  
561 \renewcommand\*\{\LWR@tagsubsectionend\}{/h4}  
562 \renewcommand\*\{\LWR@tagsubsubsection\}{h5}  
563 \renewcommand\*\{\LWR@tagsubsubsectionend\}{/h5}  
564 \renewcommand\*\{\LWR@tagparagraph\}{h6}  
565 \renewcommand\*\{\LWR@tagparagraphend\}{/h6}}

```
566 \renewcommand*{\LWR@tag subparagraph}{span class="subparagraph"{}}
567 \renewcommand*{\LWR@tag subparagraphend}{/span}
568 }% WPTitleHeading
569 {% not WPTitleHeading, part and section starting at h1
570 \renewcommand*{\LWR@tag title}{div class="title"}
571 \renewcommand*{\LWR@tag titleend}{/div}
572 \renewcommand*{\LWR@tag part}{h1}
573 \renewcommand*{\LWR@tag partend}{/h1}
574 \renewcommand*{\LWR@tag section}{h2}
575 \renewcommand*{\LWR@tag sectionend}{/h2}
576 \renewcommand*{\LWR@tag subsection}{h3}
577 \renewcommand*{\LWR@tag subsectionend}{/h3}
578 \renewcommand*{\LWR@tag subsubsection}{h4}
579 \renewcommand*{\LWR@tag subsubsectionend}{/h4}
580 \renewcommand*{\LWR@tag paragraph}{h5}
581 \renewcommand*{\LWR@tag paragraphend}{/h5}
582 \renewcommand*{\LWR@tag subparagraph}{h6}
583 \renewcommand*{\LWR@tag subparagraphend}{/h6}
584 }% not WPTitleHeading
585 }% chapter undefined
586 {% chapter defined
587 \ifbool{WPTitleHeading}{}%
588 {% not WPTitleHeading, part and chapter starting at h1
589 \renewcommand*{\LWR@tag title}{div class="title"}
590 \renewcommand*{\LWR@tag titleend}{/div}
591 \renewcommand*{\LWR@tag part}{h1}
592 \renewcommand*{\LWR@tag partend}{/h1}
593 \renewcommand*{\LWR@tag chapter}{h2}
594 \renewcommand*{\LWR@tag chapterend}{/h2}
595 \renewcommand*{\LWR@tag section}{h3}
596 \renewcommand*{\LWR@tag sectionend}{/h3}
597 \renewcommand*{\LWR@tag subsection}{h4}
598 \renewcommand*{\LWR@tag subsectionend}{/h4}
599 \renewcommand*{\LWR@tag subsubsection}{h5}
600 \renewcommand*{\LWR@tag subsubsectionend}{/h5}
601 \renewcommand*{\LWR@tag paragraph}{h6}
602 \renewcommand*{\LWR@tag paragraphend}{/h6}
603 \renewcommand*{\LWR@tag subparagraph}{span class="subparagraph"{}}
604 \renewcommand*{\LWR@tag subparagraphend}{/span}
605 }% not WPTitleHeading
606 }% chapter defined
607 }% FormatWP
608 }% AtBeginDocument
609 \end{warpHTML}
```

## 29 Remembering original formatting macros

**for HTML output:** 610 \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.

```
611 \LetLtxMacro{\LWR@origtextrm}{\textrm}
612 \LetLtxMacro{\LWR@origtextsf}{\textsf}
613 \LetLtxMacro{\LWR@origtexttt}{\texttt}
614 \LetLtxMacro{\LWR@origtextnormal}{\textnormal}
615 \LetLtxMacro{\LWR@origtextbf}{\textbf}
616 \LetLtxMacro{\LWR@origtextmd}{\textmd}
617 \LetLtxMacro{\LWR@origtextit}{\textit}
618 \LetLtxMacro{\LWR@origtextsl}{\textsl}
619 \LetLtxMacro{\LWR@origtextsc}{\textsc}
620 \LetLtxMacro{\LWR@origtextup}{\textup}
621 \LetLtxMacro{\LWR@origemph}{\emph}
622
623 \LetLtxMacro{\LWR@origrmfamily}{\rmfamily}
624 \LetLtxMacro{\LWR@origsffamily}{\sffamily}
625 \LetLtxMacro{\LWR@origttfamily}{\ttfamily}
626 \LetLtxMacro{\LWR@origbfseries}{\bfseries}
627 \LetLtxMacro{\LWR@origmdseries}{\mdseries}
628 \LetLtxMacro{\LWR@origupshape}{\upshape}
629 \LetLtxMacro{\LWR@origslshape}{\slshape}
630 \LetLtxMacro{\LWR@origscshape}{\scshape}
631 \LetLtxMacro{\LWR@origitshape}{\itshape}
632 \LetLtxMacro{\LWR@origem}{\em}
633 \LetLtxMacro{\LWR@orignormalfont}{\normalfont}
634
635 \let\LWR@origraggedright\raggedright
636 \let\LWR@origonecolumn\onecolumn
637
638 \let\LWR@origsp\sp
639 \let\LWR@origsb\sb
640 \LetLtxMacro{\LWR@origtextsuperscript}{\textsuperscript}
641 \LetLtxMacro{\LWR@orig@textsuperscript}{\textsuperscript}
642 \AtBeginDocument{
643 \LetLtxMacro{\LWR@origtextsubscript}{\textsubscript}
644 \LetLtxMacro{\LWR@orig@textsubscript}{\textsubscript}
645 }
646 \LetLtxMacro{\LWR@origunderline}{\underline}
647
648 \let\LWR@origscriptsize\scriptsize
649
650 \let\LWR@orignewpage\newpage
```

```

651
652 \let\LWR@origpagestyle\pagestyle
653 \let\LWR@origthispagestyle>thispagestyle
654 \let\LWR@origpagenumbering\pagenumbering
655
656 \LetLtxMacro{\LWR@origminipage}{\minipage}
657 \let\LWR@origendminipage\endminipage
658 \LetLtxMacro{\LWR@origparbox}{\parbox}
659
660 \let\LWR@orignewline\newline
661
662 \LetLtxMacro{\LWR@origitem}{\item}
663
664 \LetLtxMacro{\LWR@origitemize}{\itemize}
665 \LetLtxMacro{\LWR@endorigitemize}{\enditemize}
666 \LetLtxMacro{\LWR@origenumerate}{\enumerate}
667 \LetLtxMacro{\LWR@endorigenumerate}{\endenumerate}
668 \LetLtxMacro{\LWR@origdescription}{\description}
669 \LetLtxMacro{\LWR@endorigdescription}{\enddescription}
670
671 \let\LWR@origpar\par
672
673 \LetLtxMacro{\LWR@origfootnote}{\footnote}
674 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
675
676 \let\LWR@origclearpage\clearpage
677 \let\clearpage\relax
678 \let\cleardoublepage\relax

679 \end{warpHTML}

```

## 30 Accents

Native  $\text{\TeX}$  accents such as  $\text{\^n}$  will work, but many more kinds of accents are available when using Unicode-aware  $\text{X}\text{\TeX}$  and  $\text{Lua}\text{\TeX}$ .

**for HTML output:** 680 \begin{warpHTML}

Without  $\text{\AtBeginDocument}$ ,  $\text{\t}$  was being re-defined somewhere.

681 \AtBeginDocument{

The following are restored for print when inside a  $\text{lateximage}$ .

For Unicode engines, only  $\text{\t}$  needs to be redefined:

```
682 \LetLtxMacro{\LWR@origt}{\t}
```

For pdf $\text{\TeX}$ , additional work is required:

```
683 \ifPDFTeX
684 \LetLtxMacro{\LWR@origequalaccent}{\=}
685 \LetLtxMacro{\LWR@origdotaccent}{\.\.}
686 \LetLtxMacro{\LWR@origu}{\u}
687 \LetLtxMacro{\LWR@origv}{\v}
688 \LetLtxMacro{\LWR@origc}{\c}
689 \LetLtxMacro{\LWR@origd}{\d}
690 \LetLtxMacro{\LWR@origb}{\b}
```

The HTML redefinitions follow.

For pdf $\text{\TeX}$ , Unicode diacritical marks are used:

```
691 \renewcommand*{\=}[1]{\#1\HTMLunicode{0305}}
692 \renewcommand*{\.\.}[1]{\#1\HTMLunicode{0307}}
693 \renewcommand*{\u}[1]{\#1\HTMLunicode{0306}}
694 \renewcommand*{\v}[1]{\#1\HTMLunicode{030C}}
695 \renewcommand*{\c}[1]{\#1\HTMLunicode{0327}}
696 \renewcommand*{\d}[1]{\#1\HTMLunicode{0323}}
697 \renewcommand*{\b}[1]{\#1\HTMLunicode{0331}}
698 \fi
```

For all engines, a Unicode diacritical tie is used:

```
699 \def\LWR@t{\#1\HTMLunicode{0361}\#2}
700 \renewcommand*{\t}[1]{\LWR@t\#1}
```

`\LWR@restoreorigaccents` Called from `\restoreoriginalformatting` when a `lateximage` is begun.

```
701 \ifPDFTeX
702 \newcommand*{\LWR@restoreorigaccents}{%
703 \LetLtxMacro{\=}{\LWR@origequalaccent}%
704 \LetLtxMacro{\.\.}{\LWR@origdotaccent}%
705 \LetLtxMacro{\u}{\LWR@origu}%
706 \LetLtxMacro{\v}{\LWR@origv}%
707 \LetLtxMacro{\t}{\LWR@origt}%
708 \LetLtxMacro{\c}{\LWR@origc}%
709 \LetLtxMacro{\d}{\LWR@origd}%
710 \LetLtxMacro{\b}{\LWR@origb}%
711 }
712 \else% XeLaTeX, LuaLaTeX:
713 \newcommand*{\LWR@restoreorigaccents}{%
714 \LetLtxMacro{\t}{\LWR@origt}%
715 }
716 \fi
```

---

```
717 }% AtBeginDocument
```

```
718 \end{warpHTML}
```

## 31 Configuration Files

```
719 \begin{warpprint}
720 \typeout{l warp: generating configuration files}
721 \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.

```
722 \begin{warpprint}
723 \immediate\openout\LWR@quickfile=\jobname_html.tex
724 \immediate\write\LWR@quickfile{%
725 \detokenize{\PassOptionsToPackage}{%
726 {warpHTML,BaseJobname=\jobname}{l warp}}%
727 }
728 \immediate\write\LWR@quickfile{%
729 \detokenize{\input}{\string{\jobname.tex}\string }}%
730 }
731 \immediate\closeout\LWR@quickfile
732 \end{warpprint}
```

### 31.2 lwarpmk.conf

File `lwarpmk.conf` `lwarpmk.conf` is automatically (re-)created by the `l warp` 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: 733 \begin{warpprint}
734 \ifcsdef{LWR@quickfile}{}{\newwrite{\LWR@quickfile}}
735 \immediate\openout\LWR@quickfile=lwarpmk.conf
736 \ifbool{usingOSWindows}{
737 \immediate\write{\LWR@quickfile{opsystem = "Windows"}}
738 }{
739 \immediate\write{\LWR@quickfile{opsystem = "Unix"}}
740 }
741 \ifPDFTeX
742 \immediate\write{\LWR@quickfile{latexname = "pdflatex"}}
743 \fi
744 \ifXeTeX
745 \immediate\write{\LWR@quickfile{latexname = "xelatex"}}
746 \fi
747 \ifLuaTeX
748 \immediate\write{\LWR@quickfile{latexname = "lualatex"}}
749 \fi
750 \immediate\write{\LWR@quickfile{sourcename = "\jobname"}}
751 \immediate\write{\LWR@quickfile{%
752 homehtmlfilename = "\HomeHTMLFilename"%
753 }}
754 \immediate\write{\LWR@quickfile{htmlfilename = "\HTMLFilename"}}
755 \immediate\write{\LWR@quickfile{latexmk = "\ifbool{LWR@latexm}{true}{false}"}}
756 \immediate\write{\LWR@quickfile{language = "\LWR@IndexLanguage"}}
757 \immediate\write{\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}}
758 \immediate\closeout{\LWR@quickfile
759 \end{warpprint}

```

### 31.3 project.lwarpmkconf

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

```

760 \begin{warpprint}
761 \ifcsdef{LWR@quickfile}{}{\newwrite{\LWR@quickfile}}
762 \immediate\openout{\LWR@quickfile=\jobname.lwarpmkconf
763 \ifbool{usingOSWindows}{
764 \immediate\write{\LWR@quickfile{opsystem = "Windows"}}
765 }{
766 \immediate\write{\LWR@quickfile{opsystem = "Unix"}}
767 }
768 \ifPDFTeX
769 \immediate\write{\LWR@quickfile{latexname = "pdflatex"}}
770 \fi
771 \ifXeTeX
772 \immediate\write{\LWR@quickfile{latexname = "xelatex"}}

```

```
773 \fi
774 \ifLuaTeX
775 \immediate\write\LWR@quickfile{latexname = "lualatex"}
776 \fi
777 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
778 \immediate\write\LWR@quickfile{%
779   homehtmlfilename = "\HomeHTMLFilename"%
780 }
781 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
782 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexm}{true}{false}"}
783 \immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
784 \immediate\write\LWR@quickfile{xdfyfile = "\LWR@xdfyFilename"}
785 \immediate\closeout\LWR@quickfile
786 \end{warpprint}
```

### 31.4 lwarf.css

File `lwarf.css` This is the base css layer used by lwarf.

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

```
787 \begin{warpprint}
788 \begin{filecontents*}{lwarf.css}
789 /*
790   CSS stylesheet for the LaTeX lwarf package
791   Copyright 2016-2017 Brian Dunn -- BD Tech Concepts LLC
792 */
793
794
795 /* a fix for older browsers: */
796 header, section, footer, aside, nav, main,
797   article, figure { display: block; }
798
799
800 A:link {color:#000080 ; text-decoration: none ; }
801 A:visited {color:#800000 ; }
802 A:hover {color:#000080 ; text-decoration: underline ;}
803 A:active {color:#800000 ; }
804
805 a.tocpart {display: inline-block ; margin-left: 0em ;
806   font-weight: bold ;}
807 a.tocchapter {display: inline-block ; margin-left: 0em ;
808   font-weight: bold ;}
809 a.tocsection {display: inline-block ; margin-left: 1em ;
810   text-indent: -.5em ; font-weight: bold ;}
811 a.tocsubsection {display: inline-block ; margin-left: 2em ;
```

```
812     text-indent: -.5em ; }
813 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
814     text-indent: -.5em ; }
815 a.tocparagraph {display: inline-block ; margin-left: 4em ;
816     text-indent: -.5em ; }
817 a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
818     text-indent: -.5em ; }
819 a.tocfigure {margin-left: 0em}
820 a.tocsubfigure {margin-left: 2em}
821 a.toctable {margin-left: 0em}
822 a.tocsubtable {margin-left: 2em}
823 a.toctheorem {margin-left: 0em}
824 a.toclstlisting {margin-left: 0em}
825
826
827 body {
828     font-family: "DejaVu Serif", "Bitstream Vera Serif",
829                 "Lucida Bright", Georgia, serif;
830     background: #FAF7F4 ;
831     color: black ;
832     margin:0em ;
833     padding:0em ;
834     font-size: 100% ;
835     line-height: 1.2 ;
836 }
837
838 p {margin: 1.5ex 0em 1.5ex 0em ;}
839
840 /* Holds a section number to add space between it and the name */
841 span.sectionnumber { margin-right: 0em }
842
843 /* Inserted in front of index lines */
844 span.indexitem {margin-left: 0em}
845 span.indexsubitem {margin-left: 2em}
846 span.indexsubsubitem {margin-left: 4em}
847
848 div.hidden, span.hidden { display: none ; }
849
850 kbd {
851     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
852                 "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
853                 "Courier New", monospace;
854     font-size: 100% ;
855 }
856
857 pre { padding: 3pt ; }
858
859 span.strong { font-weight: bold; }
860
861 span.textmd { font-weight: normal; }
```

```
862
863 span.textsc { font-variant: small-caps; }
864
865 span.textsl { font-style: oblique; }
866
867 span.textup { font-variant: normal; }
868
869 span.textrm {
870     font-family: "DejaVu Serif", "Bitstream Vera Serif",
871     "Lucida Bright", Georgia, serif;
872 }
873
874 span.textsf {
875     font-family: "DejaVu Sans", "Bitstream Vera Sans",
876     Geneva, Verdana, sans-serif ;
877 }
878
879 span.textcircled { border: 1px solid black ; border-radius: 1ex ; }
880
881 span.underline {
882     text-decoration: underline ;
883     text-decoration-skip ;
884 }
885
886
887 /* For realscripts */
888 .supsubscript {
889     display: inline-block;
890     text-align:left ;
891 }
892
893 .supsubscript sup,
894 .supsubscript sub {
895     position: relative;
896     display: block;
897     font-size: .5em;
898     line-height: 1;
899 }
900
901 .supsubscript sup {
902     top: .5em;
903 }
904
905 .supsubscript sub {
906     top: .5em;
907 }
908
909 span.attribution {
910     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
911 }
```

```
912
913 span.citetitle {
914   margin-left: 1em ; font-size: 80% ; font-style: oblique;
915 }
916
917 span.poemtitle {
918   font-size: 120% ; font-weight: bold;
919 }
920
921 pre.tabbing {
922   font-family: "Linux Libertine Mono O", "Lucida Console",
923   "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
924   "Liberation Mono", "FreeMono", "Andale Mono",
925   "Nimbus Mono L", "Courier New", monospace;
926 }
927
928 blockquote {
929   margin-left: 0px ;
930   margin-right: 0px ;
931 }
932
933 /* quotchap is for the quotchap package */
934 div.quotchap {
935   font-style: oblique ;
936   overflow-x: auto ;
937   margin-left: 2em ;
938   margin-right: 2em ;
939 }
940
941 blockquote p, div.quotchap p {
942   line-height: 1.5;
943   text-align: left ;
944   font-size: .85em ;
945   margin-left: 3em ;
946   margin-right: 3em ;
947 }
948
949 /* qauthor is for the quotchap package */
950 div.qauthor {
951   display: block ;
952   text-align: right ;
953   margin-left: auto ;
954   margin-right: 2em ;
955   font-size: 80% ;
956   font-variant: small-caps;
957 }
958
959 div.qauthor p {
960   text-align: right ;
961 }
```

```
962
963 blockquotation {
964     margin-left: 0px ;
965     margin-right: 0px ;
966 }
967
968 blockquotation p {
969     line-height: 1.5;
970     text-align: left ;
971     font-size: .85em ;
972     margin-left: 3em ;
973     margin-right: 3em ;
974 }
975
976 div.epigraph {
977     line-height: 1.2;
978     text-align: left ;
979     padding: 3ex 1em 0ex 1em ;
980 /*     margin: 3ex auto 3ex auto ; /* Epigraph centered */
981     margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
982 /*     margin: 3ex 1em 3ex 1em ; /* /* Epigraph to the left */
983     font-size: .85em ;
984     max-width: 27em ;
985 }
986
987
988
989 div.epigraphsource{
990     text-align:right ;
991     margin-left:auto ;
992 /*     max-width: 50% ; */
993     border-top: 1px solid #AOAOAO ;
994     padding-bottom: 3ex ;
995     line-height: 1.2;
996 }
997
998 div.epigraph p { padding: .5ex ; margin: 0ex ;}
999 div.epigraphsource p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
1000
1001
1002 /* lettrine package: */
1003 span.lettrine { font-size: 3ex ; float: left ; }
1004 span.lettrinetext { font-variant: small-caps ; }
1005
1006 /* ulem and soul packages: */
1007 span.uline {
1008     text-decoration: underline ;
1009     text-decoration-skip ;
1010 }
1011
```

```
1012 span.uunderline {  
1013     text-decoration: underline ;  
1014     text-decoration-skip ;  
1015     text-decoration-style: double ;  
1016 }  
1017  
1018 span.uwave {  
1019     text-decoration: underline ;  
1020     text-decoration-skip ;  
1021     text-decoration-style: wavy ;  
1022 }  
1023  
1024 span.sout {  
1025     text-decoration: line-through ;  
1026 }  
1027  
1028 span.xout {  
1029     text-decoration: line-through ;  
1030 }  
1031  
1032 span.dashuline {  
1033     text-decoration: underline ;  
1034     text-decoration-skip ;  
1035     text-decoration-style: dashed ;  
1036 }  
1037  
1038 span.dotuline {  
1039     text-decoration: underline ;  
1040     text-decoration-skip ;  
1041     text-decoration-style: dotted ;  
1042 }  
1043  
1044 span.letterspacing { letter-spacing: .2ex ; }  
1045  
1046 span.capsspacing {  
1047     font-variant: small-caps ;  
1048     letter-spacing: .1ex ;  
1049 }  
1050  
1051 span.highlight { background: #F8E800 ; }  
1052  
1053  
1054  
1055  
1056 html body {  
1057     margin: 0 ;  
1058     line-height: 1.2;  
1059 }  
1060  
1061
```

```
1062 body div {
1063   margin: 0ex;
1064 }
1065
1066
1067 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1068 {
1069   font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1070   "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1071   "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1072   "Times New Roman", serif;
1073   font-style: normal ;
1074   font-weight: bold ;
1075   text-align: left ;
1076 }
1077
1078 h1 { /* title of the entire website, used on each page */
1079   text-align: center ;
1080   font-size: 2.5em ;
1081   padding: .4ex 0em 0ex 0em ;
1082 }
1083 h2 { font-size: 2.25em }
1084 h3 { font-size: 2em }
1085 h4 { font-size: 1.75em }
1086 h5 { font-size: 1.5em }
1087 h6 { font-size: 1.25em }
1088 span.paragraph {font-size: 1em ; font-variant: normal ;
1089   margin-right: 1em ; }
1090 span.subparagraph {font-size: 1em ; font-variant: normal ;
1091   margin-right: 1em ; }
1092
1093
1094
1095 /* Title of the file */
1096 h1 {
1097   margin: 0ex 0em 0ex 0em ;
1098   line-height: 1.3;
1099   text-align: center ;
1100 }
1101
1102 /* Part */
1103 h2 {
1104   margin: 1ex 0em 1ex 0em ;
1105   line-height: 1.3;
1106   text-align: center ;
1107 }
1108
1109 /* Chapter */
1110 h3 {
1111   margin: 3ex 0em 1ex 0em ;
```

```
1112 line-height: 1.3;
1113 }
1114
1115 /* Section */
1116 h4 {
1117 margin: 3ex 0em 1ex 0em ;
1118 line-height: 1.3;
1119 }
1120
1121 /* Sub-Section */
1122 h5 {
1123 margin: 3ex 0em 1ex 0em ;
1124 line-height: 1.3;
1125 }
1126
1127 /* Sub-Sub-Section */
1128 h6 {
1129 margin: 3ex 0em 1ex 0em ;
1130 line-height: 1.3;
1131 }
1132
1133
1134 div.titlepage {
1135 text-align: center ;
1136 }
1137
1138 .footnotes {
1139 font-size: .85em ;
1140 margin: 3ex 1em 0ex 1em ;
1141 padding-bottom: 1ex ;
1142 border-top: 1px solid silver ;
1143 }
1144
1145 .marginpar {
1146 max-width:50%;
1147 float:right;
1148 text-align:left;
1149 margin: 1ex 0.5em 1ex 1em ;
1150 padding: 1ex 0.5em 1ex 0.5em ;
1151 font-size: 85% ;
1152 border-top: 1px solid silver ;
1153 border-bottom: 1px solid silver ;
1154 overflow-x: auto;
1155 }
1156
1157 .marginpar br { margin-bottom: 2ex ; }
1158
1159 div.marginblock {
1160 max-width:50%;
1161 float:right;
```

```
1162     text-align:left;
1163     margin: 1ex 0.5em 1ex 1em ;
1164     padding: 1ex 0.5em 1ex 0.5em ;
1165     overflow-x: auto;
1166 }
1167
1168 div.marginblock div.minipage {
1169     display: block ;
1170     margin: 0pt auto 0pt auto ;
1171 }
1172
1173 div.marginblock div.minipage p { font-size: 85%}
1174
1175 div.marginblock br { margin-bottom: 2ex ; }
1176
1177
1178 section.textbody div.footnotes{
1179     margin: 3ex 0em 0ex 0em ;
1180     border-bottom: 2px solid silver ;
1181 }
1182
1183 .footnoteheader {
1184     border-top: 2px solid silver ;
1185     margin-top: 3ex ;
1186     padding-top: 1ex ;
1187     font-weight: bold ;
1188 }
1189
1190 .mpfootnotes {
1191     text-align: left ;
1192     font-size: .85em ;
1193     margin-left: 1em ;
1194     border-top: 1px solid silver ;
1195 }
1196
1197 /* Remove footnote top border in the title page. */
1198 div.titlepage div.mpfootnotes {
1199     border-top: none ;
1200 }
1201
1202
1203
1204 ol {
1205     margin: 1ex 1em 1ex 0em;
1206     line-height: 1.2;
1207 }
1208
1209 ul, body dir, body menu {
1210     margin: 1ex 1em 1ex 0em;
1211     line-height: 1.2;
```

```
1212 }
1213
1214 li { margin: 0ex 0em 1ex 0em; }
1215
1216 html {
1217   margin: 0;
1218   padding: 0;
1219 }
1220
1221 .programlisting {
1222   font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1223             "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1224             "Courier New", monospace;
1225   margin: 1ex 0ex 1ex 0ex ;
1226   padding: .5ex 0pt .5ex 0pt ;
1227   overflow-x: auto;
1228 }
1229
1230 section.textbody>pre.programlisting {
1231 border-top: 1px solid silver ;
1232 border-bottom: 1px solid silver ;
1233 }
1234
1235
1236 .inlineprogramlisting {
1237   font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1238             "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1239             "Courier New", monospace;
1240   overflow-x: auto;
1241 }
1242
1243 span.listinglabel {
1244   display: inline-block ;
1245   font-size: 70% ;
1246   width: 4em ;
1247   text-align: right ;
1248   margin-right: 2em ;
1249 }
1250
1251 div.abstract {
1252   margin: 2em 5% 2em 5% ;
1253   padding: 1ex 1em 1ex 1em ;
1254 /*   font-weight: bold ; */
1255   font-size: 90% ;
1256   text-align: left ;
1257 }
1258
1259 div.abstract dl {line-height:1.5;}
1260 div.abstract dt {color:#304070;}
1261
```

```
1262 div.abstracttitle{  
1263     font-family: "URW Classico", Optima, "Linux Biolinum 0",  
1264         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",  
1265         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;  
1266     font-weight:bold;  
1267     font-size:1.25em;  
1268     text-align: center ;  
1269 }  
1270  
1271 span.abstractrunintitle{  
1272     font-family: "URW Classico", Optima, "Linux Biolinum 0",  
1273         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",  
1274         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;  
1275     font-weight:bold;  
1276 }  
1277  
1278  
1279 .verbatim {  
1280     overflow-x: auto ;  
1281 }  
1282  
1283 .alltt {  
1284     overflow-x: auto ;  
1285 }  
1286  
1287  
1288 .bverbatim {  
1289     margin: 1ex 0pt 1ex 0pt ;  
1290     padding: .5ex 0pt .5ex 0pt ;  
1291     overflow-x: auto ;  
1292 }  
1293  
1294 .lverbatim {  
1295     margin: 1ex 0pt 1ex 0pt ;  
1296     padding: .5ex 0pt .5ex 0pt ;  
1297     overflow-x: auto ;  
1298 }  
1299  
1300 .fancyvrb {  
1301     font-size:.85em ;  
1302     margin: 3ex 0pt 3ex 0pt  
1303 }  
1304  
1305 .fancyvrblabel {  
1306     font-weight:bold;  
1307     text-align: center ;  
1308 }  
1309  
1310  
1311 .verse {
```

```
1312     font-family: "Linux Libertine Mono O", "Lucida Console",
1313         "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1314         "Liberation Mono", "FreeMono", "Andale Mono",
1315         "Nimbus Mono L", "Courier New", monospace;
1316     margin-left: 1em ;
1317 }
1318
1319
1320 div.singlespace { line-height: 1.2 ; }
1321 div.onehalfspace { line-height: 1.5 ; }
1322 div.doublespace { line-height: 2 ; }
1323
1324
1325 /* Word processor format output: */
1326 div.wpfigure { border: 1px solid red ; margin: .5ex ; padding: .5ex ; }
1327 div.wptable { border: 1px solid blue ; margin: .5ex ; padding: .5ex ; }
1328 div.wpmminipage { border: 1px solid green ; margin: .5ex ; padding: .5ex ; }
1329
1330
1331
1332
1333 /* Minipage environments, vertically aligned to top, center, bottom: */
1334 .minipage, .fminipage, .fcollorminipage {
1335     /* display: inline-block ; */
1336     /* Mini pages which follow each other will be tiled. */
1337     margin: .25em .25em .25em .25em;
1338     padding: .25em .25em .25em .25em;
1339     display: inline-flex;
1340     flex-direction: column ;
1341     overflow: auto;
1342 }
1343
1344 /* Paragraphs in the flexbox did not collapse their margins. */
1345 /* Have not yet researched this. */
1346 .minipage p {margin: .75ex 0em .75ex 0em ;}
1347
1348 .fboxBlock .minipage, .colorbox .minipage, .colorboxBlock .minipage,
1349 .fcollorbox .minipage, .fcollorboxBlock .minipage
1350     {border: none ; background: none;}
1351
1352 .fbox, .fboxBlock { border: 1px solid black ; }
1353
1354 .fbox, .fboxBlock, .fcollorbox, .fcollorboxBlock, .colorbox, .colorboxBlock,
1355 .fminipage, .fcollorminipage
1356     {display: inline-block}
1357
1358 .shadowbox, .shabox {
1359     border: 1px solid black;
1360     box-shadow: 3px 3px 3px #808080 ;
1361     border-radius: 0px ;
```

```
1362     padding: .4ex .3em .4ex .3em ;
1363     margin: Opt .3ex Opt .3ex ;
1364     display: inline-block ;
1365 }
1366
1367 .doublebox {
1368     border: 3px double black;
1369     border-radius: 0px ;
1370     padding: .4ex .3em .4ex .3em ;
1371     margin: Opt .3ex Opt .3ex ;
1372     display: inline-block ;
1373 }
1374
1375 .ovalbox, .Ovalbox {
1376     border: 1px solid black;
1377     border-radius: 1ex ;
1378     padding: .4ex .3em .4ex .3em ;
1379     margin: Opt .3ex Opt .3ex ;
1380     display: inline-block ;
1381 }
1382
1383 .Ovalbox { border-width: 2px ; }
1384
1385 .framebox {
1386     border: 1px solid black;
1387     border-radius: 0px ;
1388     padding: .3ex .2em 0ex .2em ;
1389     margin: Opt .1ex Opt .1ex ;
1390     display: inline-block ;
1391 }
1392
1393
1394 .mdframed {
1395 /*     padding: 0ex ; */
1396 /*     border: 1px solid blafck; */
1397 /*     border-radius: 0px ; */
1398     padding: 0ex ;
1399     margin: 3ex 5% 3ex 5% ;
1400 /*     display: inline-block ; */
1401 }
1402
1403 .mdframed p { padding: 0ex .5em 0ex .5em ; }
1404
1405 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
1406
1407 .mdframedtitle {
1408     padding: .5em ;
1409     display: block ;
1410     font-size: 130%
1411 }
```

```
1412
1413 .mdframedsubtitle {
1414     padding: 0ex .5em 0ex .5em ;
1415     display: block ;
1416     font-size: 115% ;
1417 }
1418
1419 .mdframedsubsubtitle {
1420     padding: 0ex .5em 0ex .5em ;
1421     display: block ;
1422 }
1423
1424 .mdtheorem {
1425     padding: 0ex .5em 0ex .5em ;
1426     margin: 3ex 5% 3ex 5% ;
1427 /*   display: inline-block ; */
1428 }
1429
1430
1431 /* framed package */
1432 .framed, pre.boxedverbatim, fcolorbox {
1433     margin: 3ex 0em 3ex 0em ;
1434     border: 1px solid black;
1435     border-radius: 0px ;
1436     padding: .3ex 1em 0ex 1em ;
1437     display: block ;
1438 }
1439
1440 .snugframed {
1441     margin: 3ex 0em 3ex 0em ;
1442     border: 1px solid black;
1443     border-radius: 0px ;
1444     display: block ;
1445 }
1446
1447 .framedleftbar {
1448     margin: 3ex 0em 3ex 0em ;
1449     border-left: 3pt solid black;
1450     border-radius: 0px ;
1451     padding: .3ex .2em .3ex 1em ;
1452     display: block ;
1453 }
1454
1455 .framedtitle {
1456     margin: 0em ;
1457     padding: 0em ;
1458     font-size: 130%
1459 }
1460
1461 .framedtitle p { padding: .3em }
```

```
1462
1463
1464
1465 dl {
1466   margin: 1ex 2em 1ex 0em;
1467   line-height: 1.3;
1468 }
1469
1470 dl dt {
1471   margin-top: 1ex;
1472   margin-left: 1em ;
1473   font-weight: bold;
1474 }
1475
1476 dl dd p { margin-top: 0em; }
1477
1478
1479 nav.toc, nav.lof, nav.lot, nav.lol, nav.lothm {
1480   font-family: "URW Classico", Optima, "Linux Biolinum 0",
1481   "DejaVu Sans", "Bitstream Vera Sans",
1482   Geneva, Verdana, sans-serif ;
1483   margin-bottom: 4ex ;
1484 }
1485
1486 nav.toc p, nav.lof p, nav.lot p, nav.lol p, nav.lothm p {
1487   line-height: 1.2 ;
1488   margin-top:.5ex ;
1489   margin-bottom:.5ex;
1490   font-size: .9em ;
1491 }
1492
1493
1494
1495 img, img.hyperimage, img.borderimage {
1496   max-width: 600px;
1497   border: 1px solid silver;
1498   box-shadow: 3px 3px 3px #808080 ;
1499   padding: .5% ;
1500   margin: .5% ;
1501   background: none ;
1502 }
1503
1504 img.inlineimage{
1505   padding: 0px ;
1506   box-shadow: none ;
1507   border: none ;
1508   background: none ;
1509   margin: 0px ;
1510   display: inline-block ;
1511   border-radius: 0px ;
```

```
1512 }
1513
1514 img.logoimage{
1515     max-width: 300px ;
1516     box-shadow: 3px 3px 3px #808080 ;
1517     border: 1px solid black ;
1518     background:none ;
1519     padding:0 ;
1520     margin:.5ex ;
1521     border-radius: 10px ;
1522 }
1523
1524
1525 .section {
1526 /*
1527     To have each section float relative to each other:
1528 */
1529 /*
1530     display: block ;
1531     float: left ;
1532     position: relative ;
1533     background: white ;
1534     border: 1px solid silver ;
1535     padding: .5em ;
1536 */
1537     margin: 0ex .5em 0ex .5em ;
1538     padding: 0 ;
1539 }
1540
1541
1542 figure {
1543     margin: 3ex auto 3ex auto ;
1544     padding: 1ex 1em 1ex 1em ;
1545     overflow-x: auto ;
1546 }
1547
1548
1549 /* To automatically center images in figures: */
1550 /*
1551 figure img.inlineimage {
1552     margin: 0ex auto 0ex auto ;
1553     display: block ;
1554 }
1555 /*
1556
1557 /* To automatically center minipages in figures: */
1558 /*
1559 figure div.minipage, figure div.minipage div.minipage {
1560     margin: 1ex auto 1ex auto ;
1561     display: block ;
```

```
1562 }
1563 */
1564
1565 figure div.minipage p { font-size: 85% ; }
1566
1567 figure.subfigure, figure.subtable {
1568     display: inline-block ; margin: 3ex 1em 3ex 1em ;
1569 }
1570
1571 figcaption .minipage { margin:0 ; padding: 0 }
1572
1573 div.minipage figure { border: none ; box-shadow: none ; }
1574
1575 div.floatrow { text-align: center; }
1576
1577 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
1578
1579 div.floatfoot { font-size: .85em ;
1580     border-top: 1px solid silver ; line-height: 1.2 ; }
1581
1582 figcaption , .lstlistingtitle {
1583     font-size: .85em ;
1584     text-align: center ;
1585     font-weight: bold ;
1586     margin-top: 1ex ;
1587     margin-bottom: 1ex ;
1588 }
1589
1590 figure.subfigure figcaption, figure.subtable figcaption {
1591     border-bottom: none ; background: none ;
1592 }
1593
1594 div.nonfloatcaption {
1595     margin: 1ex auto 1ex auto ;
1596     font-size: .85em ;
1597     text-align: center ;
1598     font-weight: bold ;
1599 }
1600
1601 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
1602 figure div.floatrow div.minipage figcaption {
1603     border: none ;
1604     background: none ;
1605 }
1606
1607
1608 table {
1609     margin: 1ex auto 1ex auto ;
1610     border-collapse: separate ;
1611     border-spacing: 0px ;
```

```
1612     line-height: 1.3 ;
1613 }
1614
1615 tr.hline td {border-top: 1px solid #808080 ; margin-top: 0ex ;
1616     margin-bottom: 0ex ; } /* for \hline */
1617
1618 tr.tbrule td {border-top: 1px solid black ; margin-top: 0ex ;
1619     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
1620
1621 td {padding: 1ex .5em 1ex .5em ;}
1622
1623 table td.tdl { text-align: left ; vertical-align: middle ; }
1624 table td.tdc { text-align: center ; vertical-align: middle ; }
1625 table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
1626 table td.tdbang { text-align: center ; vertical-align: middle ; }
1627 table td.tdr { text-align: right ; vertical-align: middle ; }
1628 table td.tdp { text-align: left ; vertical-align: bottom ; }
1629 table td.tdm { text-align: left ; vertical-align: middle ; }
1630 table td.tdb { text-align: left ; vertical-align: top ; }
1631 table td.tdP { text-align: center ; vertical-align: bottom ; }
1632 table td.tdM { text-align: center ; vertical-align: middle ; }
1633 table td.tdB { text-align: center ; vertical-align: top ; }
1634
1635 table td.tvertbarl { border-left: 1px solid black }
1636 table td.tvertbarr { border-right: 1px solid black }
1637
1638
1639 /* for cmidrules: */
1640 table td.tdrule {
1641     border-top: 1px solid #AOAOAO ;
1642 }
1643
1644 table td.tdrulel {
1645     border-top-left-radius:.5em ;
1646     border-top: 1px solid #AOAOAO ;
1647 }
1648
1649 table td.tdruler {
1650     border-top-right-radius:.5em ;
1651     border-top: 1px solid #AOAOAO ;
1652 }
1653
1654 table td.tdrulelr {
1655     border-top-left-radius:.5em ;
1656     border-top-right-radius:.5em ;
1657     border-top: 1px solid #AOAOAO ;
1658 }
1659
1660
1661 /* Margins of paragraphs inside table cells: */
```

```
1662 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
1663     padding-bottom: 1ex ; margin: 0ex ; }
1664 td.tdm p , td.tmbrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
1665     padding-bottom: 1ex ; margin: 0ex ; }
1666 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
1667     padding-bottom: 1ex ; margin: 0ex ; }
1668
1669 td.tdp , td.tdprule , td.tdP , td.tdPrule
1670     { padding: 0ex .5em 0ex .5em ; }
1671 td.tdm , td.tdmrule , td.tdM , td.tdMrule
1672     { padding: 0ex .5em 0ex .5em ; }
1673 td.tdb , td.tdbrule , td.tdB , td.tdBrule
1674     { padding: 0ex .5em 0ex .5em ; }
1675
1676
1677 /* table notes: */
1678 .tnotes {
1679     margin: 0ex 5% 1ex 5% ;
1680     padding: 0.5ex 1em 0.5ex 1em;
1681     font-size:.85em;
1682     text-align: left ;
1683 }
1684
1685 .tnotes dl dt p {margin-bottom:0px;}
1686
1687 .tnoteitemheader {margin-right: 1em;}
1688
1689
1690 /* for bigdelim */
1691 .ldelim, .rdelim { font-size: 200% }
1692
1693
1694 /* center, flushleft, flushright environments */
1695 div.center{text-align:center;}
1696 div.center table {margin-left:auto;margin-right:auto;}
1697 div.flushleft{text-align:left;}
1698 div.flushleft table {margin-left:0em ; margin-right:auto;}
1699 div.flushright{text-align:right;}
1700 div.flushright table {margin-left:auto ; margin-right: 0em ;}
1701
1702
1703 /* Fancybox */
1704 div.Btrivlist table tr td { padding: .2ex 0em ; }
1705
1706
1707 /* program listing callouts: */
1708 span.callout {
1709     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1710             Geneva, Verdana, sans-serif ;
1711     border-radius: .5em;
```

```
1712     background-color:black;
1713     color:white;
1714     padding:0px .25em 0px .25em;
1715     margin: 0 ;
1716     font-weight: bold;
1717     font-size:.72em ;
1718 }
1719
1720 div.programlisting pre.verbatim span.callout{
1721     font-size: .85em ;
1722 }
1723
1724
1725
1726
1727
1728 div.published
1729 {
1730     text-align: center ;
1731     font-variant: normal ;
1732     font-style: italic ;
1733     font-size: 1em ;
1734     margin: 3ex 0em 3ex 0em ;
1735 }
1736
1737 div.subtitle
1738 {
1739     text-align: center ;
1740     font-variant: normal ;
1741     font-style: italic ;
1742     font-size: 1.25em ;
1743     margin: 3ex 0em 3ex 0em ;
1744 }
1745
1746 div.subtitle p { margin: 1ex ; }
1747
1748 div.author
1749 {
1750     font-variant: normal ;
1751     font-style: normal ;
1752     font-size: 1em ;
1753     margin: 3ex 0em 3ex 0em ;
1754 }
1755
1756 div.oneauthor {
1757     display: inline-block ;
1758     margin: 3ex 1em 0ex 1em ;
1759 }
1760
1761 /*
```

```
1762 div.author table {
1763     margin: 3ex auto 0ex auto ;
1764     background: none ;
1765 }
1766
1767 div.author table tbody tr td { padding: .25ex ; }
1768 */
1769
1770 span.affiliation {font-size: .85em ; font-variant: small-caps; }
1771
1772 div.titledate {
1773     text-align: center ;
1774     font-size: .85em ;
1775     font-style: italic;
1776     margin: 6ex 0em 6ex 0em ;
1777 }
1778
1779
1780 nav.topnavigation{
1781     text-align: left ;
1782     padding: 0.5ex 1em 0.5ex 1em ;
1783 /*     margin: 2ex 0em 3ex 0em ; */
1784     margin: 0 ;
1785     border-bottom: 1px solid silver ;
1786     border-top: 1px solid silver ;
1787     clear:right ;
1788 }
1789
1790 nav.botnavigation{
1791     text-align: left ;
1792     padding: 0.5ex 1em 0.5ex 1em ;
1793 /*     margin: 3ex 0em 2ex 0em ; */
1794     margin: 0 ;
1795     border-top: 1px solid silver ;
1796     border-bottom: 1px solid silver ;
1797     clear:right ;
1798 }
1799
1800
1801 header{
1802     line-height: 1.2 ;
1803     font-size: 1em ;
1804 /*     border-bottom: 2px solid silver ; */
1805     margin: 0px ;
1806     padding: 0ex 1em 0ex 1em ;
1807     text-align:center ;
1808 }
1809
1810 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
1811
```

```
1812
1813 footer{
1814     font-size: .85em ;
1815     line-height: 1.2 ;
1816     margin-top: 1ex ;
1817     border-top: 2px solid silver ;
1818     padding: 2ex 1em 2ex 1em ;
1819     clear:right ;
1820     text-align:left ;
1821 }
1822
1823
1824 a.linkhome { font-weight:bold ; font-size: 1em ;}
1825
1826
1827 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
1828
1829 img.lateximage{
1830     padding: 0px 0px 0px 0px ;
1831     box-shadow: none ;
1832     border: none ;
1833     background: none ;
1834     margin: 0px 0px -.15ex 0px ;
1835     /* pdfcrop leaves a slight margin, adjust to baseline */
1836     max-width: 100% ;
1837     border-radius: 0ex ;
1838     border: none ;
1839 }
1840
1841
1842
1843 nav.sidetoc {
1844     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1845         "Lucida Bright", Georgia, serif;
1846     float:right ;
1847     width: 20%;
1848     border-left: 1px solid silver;
1849     border-top: 1px solid silver;
1850     border-bottom: 1px solid silver;
1851 /*     border-top: 2px solid #808080 ; */
1852     background: #FAF7F4 ;
1853     padding: 2ex 0em 2ex 1em ;
1854     margin: 0ex 0em 2ex 1em ;
1855     font-size:.9em ;
1856     border-radius: 20px 0px 0px 20px ;
1857 }
1858
1859 div.sidetoccocontents {
1860 /*     border-top: 1px solid silver ; */
1861     overflow-y: auto ;
```

```
1862     width: 100% ;
1863     text-align: left ;
1864 }
1865
1866
1867 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
1868     text-indent: 0 ; }
1869
1870 nav.sidetoc p a {color:black ; font-size: .7em ;}
1871
1872 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
1873     border-bottom: 1px solid silver ;    }
1874
1875 nav.sidetoc a:hover {text-decoration: underline ; }
1876
1877
1878
1879 section.textbody { margin: 0ex 1em 0ex 1em ;}
1880
1881
1882 div.multicolsheading { -webkit-column-span: all;
1883     -moz-column-span: all; column-span: all; }
1884 div.multicols { -webkit-columns: 3 380px ;
1885     -moz-columns: 3 380px ; columns: 3 380px ; }
1886 div.multicols p {margin-top: 0ex}
1887
1888
1889
1890 /* Used to support algorithmicx: */
1891 span.floatright { float: right ; }
1892
1893
1894
1895
1896 /* Native LaTeX theorems: */
1897
1898 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
1899 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
1900
1901
1902 /* theorem, amsthm, and ntheorem packages */
1903
1904 span.theoremheader,
1905 span.theoremheaderplain,
1906 span.theoremheaderdefinition,
1907 span.theoremheaderbreak,
1908 span.theoremheadermarginbreak,
1909 span.theoremheaderchangebreak,
1910 span.theoremheaderchange,
1911 span.theoremheadermargin
```

```
1912 {
1913     font-style:normal ; font-weight: bold ; margin-right: 1em ;
1914 }
1915
1916 span.amsthmnameplain,
1917 span.amsthmnamedefinition,
1918 span.amsthmnumberplain,
1919 span.amsthmnumberdefinition
1920 {
1921     font-style:normal ; font-weight: bold ;
1922 }
1923
1924
1925 span.amsthmnameremark,
1926 span.amsthmnumberremark
1927 {font-style:italic ; font-weight: normal ; }
1928
1929
1930 span.amsthmnoteplain,
1931 span.amsthmnotedefinition
1932 {font-style:normal ;}
1933
1934
1935 span.theoremheaderremark,
1936 span.theoremheaderproof,
1937 span.amsthmproofname
1938 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
1939
1940 span.theoremheadersc
1941 {
1942     font-style:normal ;
1943     font-variant: small-caps ;
1944     font-weight: normal ;
1945     margin-right: 1em ;
1946 }
1947
1948 .theoremendmark {float:right}
1949
1950 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
1951 div.theorembodybreak, div.theorembodynonumberbreak,
1952 div.theorembodymarginbreak,
1953 div.theorembodychangebreak,
1954 div.theorembodychange,
1955 div.theorembodymargin
1956 {
1957     font-style:italic;
1958     margin-top: 3ex ; margin-bottom: 3ex ;
1959 }
1960
1961 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
```

```
1962 div.theorembodyplainupright, nonumberplainuprightsc,
1963 div.amsthmbodydefinition, div.amsthmbodyremark,
1964 div.amsthmproof
1965 {
1966     font-style: normal ;
1967     margin-top: 3ex ; margin-bottom: 3ex ;
1968 }
1969
1970 span.amsthmnoteremark {}
1971
1972
1973
1974 /*
1975 For CSS LaTeX and related logos:
1976 Based on:
1977 http://edward.oconnor.cx/2007/08/tex-poshlet
1978 http://nitens.org/taraborelli/texlogo
1979 */
1980
1981 .latexlogofont {
1982     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1983         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1984     font-variant: normal ;
1985 }
1986
1987 .latexlogo {
1988     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1989         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1990     letter-spacing: .03em ;
1991     font-size: 1.1em;
1992 }
1993
1994 .latexlogo sup {
1995     text-transform: uppercase;
1996     letter-spacing: .03em ;
1997     font-size: 0.85em;
1998     vertical-align: 0.15em;
1999     margin-left: -0.36em;
2000     margin-right: -0.15em;
2001 }
2002
2003 .latexlogo sub {
2004     text-transform: uppercase;
2005     vertical-align: -0.5ex;
2006     margin-left: -0.1667em;
2007     margin-right: -0.125em;
2008     font-size: 1em;
2009 }
2010
2011 .xetexlogo {
```

```
2012     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2013         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2014     letter-spacing: .03em ;
2015     font-size: 1.1em;
2016 }
2017
2018 /* A smaller gap between Xe and Tex v.s. LaTeX: */
2019 .xetexlogo sub {
2020     text-transform: uppercase;
2021     vertical-align: -0.5ex;
2022     margin-left: -0.0667em;
2023     margin-right: -0.2em;
2024     font-size: 1em;
2025     letter-spacing: .03em ;
2026 }
2027
2028 /* A large gap between Xe and LaTeX v.s. TeX: */
2029 .xelatexlogo sub {
2030     text-transform: uppercase;
2031     vertical-align: -0.5ex;
2032     margin-left: -0.0667em;
2033     margin-right: -.05em;
2034     font-size: 1em;
2035     letter-spacing: .03em ;
2036 }
2037
2038 .amslogo {
2039     font-family: "TeXGyreChorus","URW Chancery L",
2040         "Apple Chancery","ITC Zapf Chancery","Monotype Corsiva",
2041         "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
2042         "Hoefler Text", Times, "Times New Roman", serif;
2043     font-style: italic;
2044 }
2045
2046 .lyxlogo {
2047     font-family: "URW Classico", Optima, "Linux Biolinum O",
2048         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
2049         Verdana, sans-serif ;
2050 }
2051
2052
2053
2054
2055 /* Only display top and bottom navigation if a small screen: */
2056 /* Hide the sidetoc if a small screen: */
2057 nav.topnavigation { display:none; }
2058 nav.botnavigation { display:none; }
2059
2060 @media screen and (max-width: 45em) {
2061     nav.sidetoc {display:none;} */
```

```
2062     nav.sidetoc {  
2063         float: none ;  
2064         width: 100% ;  
2065         margin: 5ex 0px 5ex 0px ;  
2066         padding: 0 ;  
2067         border-radius: 0 ;  
2068         border-bottom: 1px solid black ;  
2069         border-top: 1px solid black ;  
2070         box-shadow: none ;  
2071     }  
2072 /*  nav.topnavigation { display:block } */  
2073 nav.botnavigation { display:block }  
2074 .marginpar {  
2075     max-width: 100%;  
2076     float: none;  
2077     display:block ;  
2078     margin: 1ex 1em 1ex 1em ;  
2079 }  
2080 }  
2081  
2082 @media print {  
2083     body {  
2084         font-family: "Linux Libertine O",  
2085         "DejaVu Serif", "Bitstream Vera Serif",  
2086         "Liberation Serif", "Nimbus Roman No 9 L",  
2087         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;  
2088     }  
2089     nav.sidetoc { display:none; }  
2090     nav.topnavigation { display: none; }  
2091     nav.botnavigation { display: none; }  
2092 }  
2093  
2094 @media handheld {  
2095     nav.sidetoc { display:none; }  
2096     nav.topnavigation { display:block }  
2097     nav.botnavigation { display:block }  
2098 }  
2099  
2100 @media projection {  
2101     nav.sidetoc { display:none; }  
2102     nav.topnavigation { display:block }  
2103     nav.botnavigation { display:block }  
2104 }  
2105 \end{filecontents*}  
2106 % \end{Verbatim} % for syntax highlighting  
2107 \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.

```
2108 \begin{warpprint}
2109 \begin{filecontents*}{lwarp_sagebrush.css}
2110 @import url("lwarp.css") ;
2111
2112
2113 A:link {color:#105030 ; text-decoration: none ; }
2114 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
2115 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
2116 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
2117
2118
2119
2120 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2121 {
2122     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2123             "Linux Libertine 0", "Liberation Serif",
2124             "Nimbus Roman No 9 L", "FreeSerif",
2125             "Hoefler Text", Times, "Times New Roman", serif;
2126     font-variant: small-caps ;
2127     font-weight: normal ;
2128     color: #304070 ;
2129     text-shadow: 2px 2px 3px #808080;
2130 }
2131
2132 h1 { /* title of the entire website, used on each page */
2133     font-variant: small-caps ;
2134     color: #304070 ;
2135     text-shadow: 2px 2px 3px #808080;
2136     background-color: #F7F7F0 ;
2137     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
2138 }
2139
2140 h1 {
2141     border-bottom: 1px solid #304070;
2142     border-top: 2px solid #304070;
2143 }
2144
2145 h2 {
2146     border-bottom: 1px solid #304070;
2147     border-top: 2px solid #304070;
2148     background-color: #F7F7F0 ;
2149     background-image: linear-gradient(to bottom, #F7F7F0, #DADOC0);
```

```
2150 }
2151
2152
2153
2154 div.abstract {
2155     background: #f5f5eb ;
2156     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2157
2158     border: 1px solid silver;
2159     border-radius: 1em ;
2160 }
2161
2162 div.abstract dl {line-height:1.5;}
2163 div.abstract dt {color:#304070;}
2164
2165 div.abstracttitle{
2166     font-family: "URW Classico", Optima, "Linux Biolinum O",
2167         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2168         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2169     font-weight:bold;
2170     font-variant: small-caps ;
2171     font-size:1.5em;
2172     border-bottom: 1px solid silver ;
2173     color: #304070 ;
2174     text-align: center ;
2175     text-shadow: 1px 1px 2px #808080;
2176 }
2177
2178 span.abstractrunintitle{
2179     font-family: "URW Classico", Optima, "Linux Biolinum O",
2180         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2181         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2182     font-weight:bold;
2183 }
2184
2185
2186 div.epigraph {
2187     background: #f5f5eb ;
2188     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2189
2190     border: 1px solid silver ;
2191     border-radius: 1ex ;
2192     box-shadow: 3px 3px 3px #808080 ;
2193 }
2194
2195
2196 .example {
2197     background-color: #f5f5eb ;
2198     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2199
```

```
2200 }
2201
2202 div.exampletitle{
2203     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2204         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2205         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2206     font-weight:bold;
2207     font-variant: small-caps ;
2208     border-bottom: 1px solid silver ;
2209     color: #304070 ;
2210     text-align: center ;
2211     text-shadow: 1px 1px 2px #808080;
2212 }
2213
2214
2215 .sidebar {
2216     background-color: #f5f5eb ;
2217     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2218
2219 }
2220
2221 div.sidebartitle{
2222     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2223         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2224         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2225     font-weight:bold;
2226     font-variant: small-caps ;
2227     border-bottom: 1px solid silver ;
2228     color: #304070 ;
2229     text-align: center ;
2230     text-shadow: 1px 1px 2px #808080;
2231 }
2232
2233
2234 .fancyvrblabel {
2235     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2236         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2237         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2238     font-weight:bold;
2239     font-variant: small-caps ;
2240     font-size: 1.5em ;
2241     color: #304070 ;
2242     text-align: center ;
2243     text-shadow: 1px 1px 2px #808080;
2244 }
2245
2246
2247
2248 div.minipage {
2249     background-color: #eeeeee7 ;
```

```
2250     border: 1px solid silver ;
2251     border-radius: 1ex ;
2252 }
2253
2254 section.textbody > div.minipage {
2255     box-shadow: 3px 3px 3px #808080 ;
2256 }
2257
2258 div.fboxBlock div.minipage { box-shadow: none ; }
2259
2260 .framed .minipage , .framedleftbar .minipage {
2261     border: none ;
2262     background: none ;
2263     padding: 0ex ;
2264     margin: 0ex ;
2265 }
2266
2267 figure.figure .minipage, figcaption .minipage { border: none; }
2268
2269 div.marginblock div.minipage { border: none; }
2270
2271 figure , div.marginblock {
2272     background-color: #eeeeee ;
2273     border: 1px solid silver ;
2274     border-radius: 1ex ;
2275     box-shadow: 3px 3px 3px #808080 ;
2276 }
2277
2278 figure figure {
2279     border: 1px solid silver ;
2280     margin: 0em ;
2281     box-shadow: none ;
2282 }
2283
2284 /*
2285 figcaption {
2286     border-top: 1px solid silver ;
2287     border-bottom: 1px solid silver ;
2288     background-color: #e8e8e8 ;
2289 }
2290 */
2291
2292
2293 div.table {
2294     box-shadow: 3px 3px 3px #808080 ;
2295 }
2296
2297 /*
2298 .tnotes {
2299     background: #e8e8e8;
```

```
2300     border: 1px solid silver;
2301 }
2302 */
2303
2304
2305 nav.topnavigation{
2306     background-color: #b0b8b0 ;
2307     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
2308 }
2309
2310 nav.botnavigation{
2311     background-color: #b0b8b0 ;
2312     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
2313 }
2314
2315
2316
2317 header{
2318     background-color: #F7F7F0 ;
2319     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
2320 }
2321
2322 footer{
2323     background-color: #F7F7F0 ;
2324     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
2325 }
2326
2327
2328
2329 nav.sidetoc {
2330     background-color: #F7F7F0 ;
2331     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
2332     box-shadow: 3px 3px 3px #808080 ;
2333     border-radius: 0px 0px 0px 20px ;
2334 }
2335
2336 div.sidetoctitle {color: #304070 ; }
2337
2338 nav.sidetoc a:hover {
2339     color:#006000 ;
2340     text-decoration: none ;
2341     text-shadow:0px 0px 2px #a0a0a0;
2342 }
2343
2344
2345 @media screen and (max-width: 45em) {
2346     nav.sidetoc { border-radius: 0 ; }
2347 }
2348
2349
```

```
2350 \end{filecontents*}
2351 % \end{Verbatim}%
2352 \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.

```
2353 \begin{warpprint}
2354 \begin{filecontents*}{lwarp_formal.css}
2355 @import url("lwarp.css") ;
2356
2357
2358
2359 A:link {color:#802020 ; text-decoration:none; }
2360 A:visited {color:#802020 ; text-shadow:none ;}
2361 A:hover {color:#400000 ; text-shadow:none ;}
2362 A:active {color:#C00000 ; text-shadow:none ;}
2363
2364
2365 body {
2366     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2367             "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2368             "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2369             "Times New Roman", serif;
2370     background: #ffffcf5;
2371 }
2372
2373 span.textrm {
2374     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2375             "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2376             "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2377             "Times New Roman", serif;
2378 }
2379
2380 span.textsf {
2381     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2382             Geneva, Verdana, sans-serif ;
2383 }
2384
2385
2386
2387 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2388 {
```

```
2389     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2390         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2391         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2392         "Times New Roman", serif;
2393     color: #800000 ;
2394     text-shadow: none ;
2395 }
2396
2397 h1, h2 {
2398     background-color: #ffffcf5 ;
2399     background-image: none ;
2400     border-bottom: 1px solid #808080;
2401     border-top: 2px solid #808080;
2402 }
2403
2404 div.abstracttitle {
2405     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2406         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2407         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2408         "Times New Roman", serif;
2409     color: black ;
2410     text-shadow: none ;
2411 }
2412
2413 span.abstractrunintitle {
2414     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2415         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2416         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2417         "Times New Roman", serif;
2418     color: black ;
2419     text-shadow: none ;
2420 }
2421
2422 div.abstract { font-size: 100% }
2423
2424 .sidebar {
2425     background: #ffffcf5;
2426     background-image: none ;
2427     margin: 2em 5% 2em 5%;
2428     padding: 0.5em 1em;
2429     border: none ;
2430     border-top : 1px solid silver;
2431     border-bottom : 1px solid silver;
2432     font-size: 90% ;
2433 }
2434
2435 div.sidebartitle{
2436     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2437         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2438         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
```

```
2439     "Times New Roman", serif;
2440     color: #800000 ;
2441     text-shadow: none ;
2442     border: none ;
2443 }
2444
2445 .example {
2446     background: #ffffcf5;
2447     background-image: none ;
2448     margin: 2em 5% 2em 5%;
2449     padding: 0.5em 1em;
2450     border: none ;
2451     border-top : 1px solid silver;
2452     border-bottom : 1px solid silver;
2453 }
2454
2455 div.exampletitle{
2456     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2457         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2458         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2459         "Times New Roman", serif;
2460     color: #800000 ;
2461     text-shadow: none ;
2462     border: none ;
2463 }
2464
2465 div.fancyvrblabel{
2466     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2467         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2468         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2469         "Times New Roman", serif;
2470     color: #800000 ;
2471     text-shadow: none ;
2472     border: none ;
2473 }
2474
2475
2476
2477 .verse {
2478     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2479         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2480         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2481         "Times New Roman", serif;
2482 }
2483
2484
2485 figure {
2486     margin: 3ex 5% 3ex 5% ;
2487     padding: 1ex 1em 1ex 1em ;
2488     background-color: #ffffcf5 ;
```

```
2489     overflow-x: auto ;
2490     border: none ;
2491 /*      border-top: 1px solid silver; */
2492 /*      border-bottom: 1px solid silver; */
2493 }
2494
2495
2496 figcaption , .lstlisting {
2497     border: none ;
2498 /*      border-top: 1px solid silver ; */
2499 /*      border-bottom: 1px solid silver ; */
2500     background-color: #fffcf5 ;
2501 }
2502
2503 .tnotes {
2504     background: #fffcf5 ;
2505 }
2506
2507 .theorem {
2508     background: none ;
2509 }
2510
2511 .minipage {
2512     background-color: #fffcf5 ;
2513     border: none ;
2514 }
2515
2516 div.floatrow figure { border: none ; }
2517
2518 figure figure { border: none ; }
2519
2520
2521 nav.toc, nav.lof, nav.lot, nav.lol {
2522     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2523         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2524         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2525         "Times New Roman", serif;
2526 }
2527
2528 nav.sidetoc {
2529     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2530         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2531         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2532         "Times New Roman", serif;
2533     background-image: linear-gradient(to bottom, #fffcf5, #C0C0C0);
2534     border-radius: 0px 0px 0px 20px ;
2535 }
2536
2537 div.sidetoctitle{
2538     color: #800000 ;
```

```
2539 }
2540
2541 header{
2542     background-color: #e0e0e0 ;
2543     background-image: linear-gradient(to top, #ffffcf5, #b0b0b0);
2544     text-align:center ;
2545 }
2546
2547 footer{
2548     background-color: #e0e0e0 ;
2549     background-image: linear-gradient(to bottom, #ffffcf5, #b0b0b0);
2550     padding: 2ex 1em 2ex 1em ;
2551     clear:right ;
2552     text-align:left ;
2553 }
2554
2555 nav.botnavigation {
2556     background: #dedcd5 ;
2557     border-top: 1px solid black ;
2558 }
2559 \end{filecontents*}
2560 % \end{Verbatim}%
2561 \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.

```
2562 \begin{warpprint}
2563 \begin{filecontents*}{sample_project.css}
2564 /* ( --- Start of project.css --- ) */
2565 /* A sample project-specific CSS file for lwarp --- ) */
2566
2567 /* Load default lwarp settings: */
2568 @import url("lwarp.css") ;
2569 /* or lwarp_formal.css, lwarp_sagebrush.css */
2570
2571 /* Project-specific CSS setting follow here. */
2572 /* . . . */
2573
2574 /* ( --- End of project.css --- ) */
2575 \end{filecontents*}
2576 % \end{Verbatim}%
2577 \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.

```
2578 \begin{warpprint}
2579 \begin{filecontents*}{lwarp.xdy}
2580 (require "tex/inputenc/latin.xdy")
2581 (merge-rule "\PS *" "Postscript")
2582 (require "texindy.xdy")
2583 (require "page-ranges.xdy")
2584 (require "book-order.xdy")
2585 (markup-locref :open "\hyperindexref{" :close "}")
2586 \end{filecontents*}
2587 % \end{Verbatim} for syntax highlighting
2588 \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.

```
2589 \begin{warpprint}
2590 \begin{filecontents*}{lwarp_mathjax.txt}
2591 <!-- https://groups.google.com/forum/#!topic/
2592           mathjax-users/jUtewUcE2bY -->
2593 <script type="text/x-mathjax-config">
2594 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
2595     var seteqsectionDefault = {name: "", num: 0};
2596     var seteqsections = {}, seteqsection = seteqsectionDefault;
2597     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
2598     var AMS = MathJax.Extension["TeX/AMSmash"];
2599     TEX.Definitions.Add({
2600         macros: {
2601             seteqsection: "mySection",
2602             seteqnumber: "mySetEqNumber"
2603         }
2604     });
2605
2606     PARSE.Augment({
2607         mySection: function (name) {
2608             seteqsection.num = AMS.number;
```

```
2609     var n = this.GetArgument(name);
2610     if (n === "") {
2611       seteqsection = seteqsectionDefault;
2612     } else {
2613       if (!seteqsections["_"+n])
2614         seteqsections["_"+n] = {name:n, num:0};
2615       seteqsection = seteqsections["_"+n];
2616     }
2617     AMS.number = seteqsection.num;
2618   },
2619   mySetEqNumber: function (name) {
2620     var n = this.GetArgument(name);
2621     if (!n || !n.match(/^\ *[0-9]+\ */))
2622       n = ""; else n = parseInt(n)-1;
2623     <!-- $ syntax highlighting -->
2624     if (n === "" || n < 1)
2625       TEX.Error
2626       ("Argument to "+name+" should be a positive integer");
2627     AMS.number = n;
2628   }
2629 });
2630 MathJax.Hub.Config({
2631   TeX: {
2632     equationNumbers: {
2633       formatTag: function (n)
2634         {return "("+(seteqsection.name+"."+n).replace(/\./,"")+")"}, 
2635       formatID: function (n) {
2636         n = (seteqsection.name+'.'+n).replace
2637           (/[:]'/>&)/g,"").replace(/\./,"");
2638         return 'mjax-eqn-' + n;
2639       }
2640     }
2641   }
2642 });
2643 });
2644 </script>
2645
2646 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
2647 <script type="text/x-mathjax-config">
2648   MathJax.Ajax.config.path["Contrib"] =
2649     "https://cdn.mathjax.org/mathjax/contrib";
2650 </script>
2651
2652 <!-- https://github.com/burnpanck/MathJax-siunitx -->
2653
2654 <script type="text/x-mathjax-config">
2655   MathJax.Hub.Config({
2656     extensions: ["tex2jax.js","[siunitx]/siunitx.js"],
2657     jax: ["input/TeX","output/HTML-CSS"],
2658     tex2jax: {inlineMath: [[[$,$],[["\\(","\\")]]]}},
```

```

2659   TeX: {extensions: ["AMSmath.js", "AMSSymbols.js", "sinuitx.js"]}
2660 });
2661 MathJax.Ajax.config.path['siunitx'] = 'http://rawgit.com/burnpanck/MathJax-siunitx/master/';
2662 </script>
2663
2664 <script type="text/x-mathjax-config">
2665 MathJax.Hub.Config({
2666   TeX: {
2667     equationNumbers: {
2668       autoNumber: "AMS"
2669     }
2670   }
2671 });
2672 </script>
2673
2674 <!-- Alternative CDN provider: -->
2675 <script type="text/javascript" async
2676 src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?config=TeX-AMS_HTML-full">
2677 </script>
2678
2679 <!-- No longer supported after April 30, 2017: -->
2680 <!--
2681 <script
2682   src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
2683 </script>
2684 -->
2685
2686 \end{filecontents*}
2687 % \end{Verbatim}%
2688 \end{warpprint}

```

### 31.10 lwarfpmk option

The following is only generated if the `lwarfpmk` option was given to `lwarf`.

```

2689 \begin{LWR@createlwarfpmk}

Prog lwarfpmk Creates a local copy of lwarfpmk:

2690 \begin{filecontents*}{lwarfpmk.lua}
2691 #!/usr/bin/env texlua
2692
2693 -- Copyright 2016-2017 Brian Dunn
2694
2695 -- Print the usage of the lwarfpmk command:
2696
2697 printversion = "v0.42"

```

```
2698
2699 function printhelp ()
2700 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.");
2701 end
2702
2703 function printusage ()
2704 print ( [[
2705
2706 lwarpmk print [project]: Compile the print version if necessary.
2707 lwarpmk print1 [project]: Forced single compile of the print version.
2708 lwarpmk printindex [project]: Process the index for the print version.
2709 lwarpmk printglossary [project]: Process the glossary for the print version.
2710 lwarpmk html [project]: Compile the HTML version if necessary.
2711 lwarpmk html1 [project]: Forced single compile of the HTML version.
2712 lwarpmk htmlindex [project]: Process the index for the html version.
2713 lwarpmk htmlglossary [project]: Process the glossary for the html version.
2714 lwarpmk again [project]: Touch the source code to trigger recompiles.
2715 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
2716 lwarpmk pdftohtml [project]:
2717     For use with latexmk or a Makefile:
2718     Convert project_html.pdf to project_html.html and
2719     individual HTML files.
2720 lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, *_html_inc.*., .gl*
2721 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
2722 lwarpmk -h: Print this help message.
2723 lwarpmk --help: Print this help message.
2724
2725 ]] )
2726 printconf ()
2727 end
2728
2729 -- Print the format of the configuration file lwarpmk.conf:
2730
2731 function printconf ()
2732 print ( [[
2733 An example lwarpmk.conf or <project>.lwarpmkconf project file:
2734 --
2735 opsystem = "Unix"    (or "Windows")
2736 latexname = "pdflatex" (or "lualatex", or "xelatex")
2737 sourcename = "projectname" (the source-code filename w/o .tex)
2738 homehtmlfilename = "index" (or perhaps the project name)
2739 htmlfilename = "" (or "projectname" - filename prefix)
2740 latexmk = "false" (or "true" to use latexmk to build PDFs)
2741 languge = "english" (use a language supported by xindy)
2742 xdyfile = "lwarp.xdy" (or a custom file based on lwarp.xdy)
2743 --
2744 Filenames must contain only letters, numbers, underscore, or dash.
2745 Values must be in "quotes".
2746
2747 ]] ) ;
```

```
2748 end
2749
2750
2751 -- Split one large sourcefile into a number of files,
2752 -- starting with destfile.
2753 -- The file is split at each occurrence of <!--|Start file|newfilename|*
2754
2755 function splitfile (destfile,sourcefile)
2756 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
2757 local sfile = io.open(sourcefile)
2758 io.output(destfile)
2759 for line in sfile:lines() do
2760 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
2761 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
2762 io.output(newfilename) ;
2763 else -- not a splitpoint
2764 io.write (line .. "\n") ;
2765 end
2766 end -- do
2767 io.close(sfile)
2768 end -- function
2769
2770 -- Incorrect value, so print an error and exit.
2771
2772 function cvalueerror ( line, linenum , cvalue )
2773     print ( linenum .. " : " .. line ) ;
2774     print ("lwarpmk: incorrect variable value \" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
2775     printconf () ;
2776     os.exit(1) ;
2777 end
2778
2779 -- Load settings from the project's "lwarpmk.conf" file:
2780
2781 function loadconf ()
2782 -- Default configuration filename:
2783 local conffile = "lwarpmk.conf"
2784 -- Optional configuration filename:
2785 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
2786 -- Default language:
2787 language = "english"
2788 -- Default xdyfile:
2789 xdyfile = "lwarp.xdy"
2790 -- Verify the file exists:
2791 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
2792 print("lwarpmk: " .. conffile .. " does not exist.")
2793 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
2794 printhelp () ;
2795 os.exit(1) -- exit the entire lwarpmk script
2796 else -- file exists
2797 -- Read the file:
```

```
2798 print ("lwarpmk: Reading " .. conffile .. ".")
2799 local cfile = io.open(conffile)
2800 -- Scan each line:
2801 local linenum = 0
2802 for line in cfile:lines() do -- scan lines
2803 linenum = linenum + 1
2804 i,j,cvarname,cvalue = string.find (line, "([%w-]*%)%s*=%s*\"([%w%-_.]*%)\\\"") ;
2805 -- Error if incorrect enclosing characters:
2806 if ( i == nil ) then
2807 print ( linenum .. " : " .. line ) ;
2808 print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
2809 printconf () ;
2810 os.exit(1) ;
2811 end
2812 if ( cvarname == "opsystem" ) then
2813     -- Verify choice of opsystem:
2814     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
2815         opsystem = cvalue
2816     else
2817         cvalueerror ( line, linenum , cvalue )
2818     end
2819 elseif ( cvarname == "latexname" ) then
2820     -- Verify choice of LaTeX compiler:
2821     if (
2822         (cvalue == "pdflatex") or
2823         (cvalue == "xelatex") or
2824         (cvalue == "lualatex")
2825     ) then
2826         latexname = cvalue
2827     else
2828         cvalueerror ( line, linenum , cvalue )
2829     end
2830 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
2831 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
2832 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
2833 elseif ( cvarname == "latexmk" ) then latexmk = cvalue
2834 elseif ( cvarname == "language" ) then language = cvalue
2835 elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue
2836 else
2837 print ( linenum .. " : " .. line ) ;
2838 print ("lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. conffile .. ".\n" ) ;
2839 printconf () ;
2840 os.exit(1) ;
2841 end
2842 end -- do scan lines
2843 io.close(cfile)
2844 end -- file exists
2845 -- Select some operating-system commands:
2846 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
2847 rmname = "rm"
```

```
2848 mvname = "mv"
2849 touchnamepre = "touch"
2850 touchnamepost = ""
2851 dirslash = "/"
2852 opquote= "\`"
2853 elseif opsystem=="Windows" then -- For Windows
2854 rmname = "DEL"
2855 mvname = "MOVE"
2856 touchnamepre = "COPY /b"
2857 touchnamepost = "+,,"
2858 dirslash = "\\"
2859 opquote= "\""
2860 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
2861 end --- for Windows
2862
2863 -- set xindycmd according to pdflatex vs xelatex/lualatex:
2864 if ( latexname == "pdflatex" ) then
2865 xindycmd = "texindy -C utf8"
2866 glossarycmd = "xindy -C utf8"
2867 else
2868 xindycmd = "xindy -M texindy -C utf8"
2869 glossarycmd = "xindy -C utf8"
2870 end
2871
2872 end -- loadconf
2873
2874
2875 function refreshdate ()
2876 os.execute(touchnamepre .. " " .. sourcename .. ".tex" .. touchnamepost)
2877 end
2878
2879
2880 -- Scan the LaTeX log file for the phrase "Rerun to get",
2881 -- indicating that the file should be compiled again.
2882 -- Return true if found.
2883
2884 function reruntoget (filesource)
2885 local fsource = io.open(filesource)
2886 for line in fsource:lines() do
2887 if ( string.find(line,"Rerun to get") ~= nil ) then
2888     io.close(fsource)
2889     return true
2890 end
2891 end
2892 io.close(fsource)
2893 return false
2894 end
2895
2896
2897 -- Compile one time, return true if should compile again.
```

```
2898 -- fsuffix is "" for print, "_html" for HTML output.
2899
2900 function onetime (fsuffix)
2901 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
2902 err = os.execute(
2903 --     "echo " ..
2904     latexname .. " " .. sourcename..fsuffix )
2905 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2906 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
2907 end
2908
2909
2910 -- Compile up to five times.
2911 -- fsuffix is "" for print, "_html" for HTML output
2912
2913 function manytimes (fsuffix)
2914 if onetime(fsuffix) == true then
2915 if onetime(fsuffix) == true then
2916 if onetime(fsuffix) == true then
2917 if onetime(fsuffix) == true then
2918 if onetime(fsuffix) == true then
2919 end end end end end
2920 end
2921
2922 -- Exit if the given file does not exist.
2923
2924 function verifyfileexists (filename)
2925 if (lfs.attributes ( filename , "modification" ) == nil ) then
2926 print ( "lwarpmk: " .. filename .. " not found." ) ;
2927 os.exit ( 1 ) ;
2928 end
2929 end
2930
2931
2932 -- Convert <project>_html.pdf into HTML files:
2933
2934 function pdftohtml ()
2935     -- Convert to text:
2936     print ("lwarpmk: Converting " .. sourcename
2937             .."_html.pdf to " .. sourcename .. "_html.html")
2938     os.execute("pdftotext -enc UTF-8 -nopgbrk -layout "
2939             .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
2940     -- Split the result into individual HTML files:
2941     splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
2942 end
2943
2944
2945 -- Remove auxiliary files:
2946
2947 function removeaux ()
```

```
2948     os.execute ( rmtree .. " " ..
2949         sourcename .. ".aux" .. sourcename .. "_html.aux" ..
2950         sourcename .. ".toc" .. sourcename .. "_html.toc" ..
2951         sourcename .. ".lof" .. sourcename .. "_html.lof" ..
2952         sourcename .. ".lot" .. sourcename .. "_html.lot" ..
2953         sourcename .. ".idx" .. sourcename .. "_html.idx" ..
2954         sourcename .. ".ind" .. sourcename .. "_html.ind" ..
2955         sourcename .. ".log" .. sourcename .. "_html.log" ..
2956         sourcename .. ".gl*" .. sourcename .. "_html.gl*" ..
2957         "*_html_inc.*"
2958     )
2959 end
2960
2961
2962
2963 -- Create lateximages based on lateximages.txt:
2964 function createlateximages ()
2965 print ("lwarpmk: Creating lateximages.")
2966 local limagesfile = io.open("lateximages.txt")
2967 -- Create the lateximages directory, ignore error if already exists
2968 err = os.execute("mkdir lateximages")
2969 -- Scan lateximages.txt
2970 for line in limagesfile:lines() do
2971 -- lwimgpage is the page number in the PDF which has the image
2972 -- lwimgnum is the sequential lateximage number to assign for the image
2973 i,j,lwimgpage,lwimgnum = string.find (line,"|(.*)|(.*)|")
2974 -- For each entry:
2975 if ( (i~=nil) ) then
2976 -- Separate out the image into its own single-page pdf:
2977 err = os.execute(
2978 "pdfseparate -f " .. lwimgpage .. " -l " ..
2979 lwimgpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
2980 -- Crop the image:
2981 err = os.execute(
2982 "pdffcrop --hires lateximagetemp-" .. lwimgpage .. ".pdf lateximage-" .. lwimgnum .. ".pdf")
2983 if ( err ~= 0 ) then print ("lwarpmk: File error.") ; os.exit(1) ; end
2984 -- Convert the image to svg:
2985 err = os.execute(
2986 "pdftocairo -svg lateximage-" .. lwimgnum .. ".pdf lateximage-" .. lwimgnum .. ".svg")
2987 if ( err ~= 0 ) then print ("lwarpmk: File error.") ; os.exit(1) ; end
2988 -- Move the result into lateximages/:
2989 err = os.execute(
2990 mvname .. " lateximage-" .. lwimgnum .. ".svg lateximages" .. dirslash)
2991 if ( err ~= 0 ) then print ("lwarpmk: File error.") ; os.exit(1) ; end
2992 -- Remove the temporary files:
2993 err = os.execute(
2994 rmname .. " lateximage-" .. lwimgnum .. ".pdf lateximagetemp-" .. lwimgpage .. ".pdf")
2995 if ( err ~= 0 ) then print ("lwarpmk: File error.") ; os.exit(1) ; end
2996 end
2997 end -- do
```

```
2998 io.close(limagesfile)
2999 end -- function
3000
3001
3002 -- Use latexmk to compile source and index:
3003 -- fsuffix is "" for print, or "_html" for HTML
3004 function compilelatexmk ( fsuffix )
3005     -- The recorder option is required to detect changes in <project>.tex
3006     -- while we are loading <project>_html.tex.
3007     err=os.execute ( "latexmk -pdf -dvi -ps -recorder "
3008         .. "-e "
3009         .. opquote
3010         .. "$makeindex = q/" -- $
3011         .. xindycmd
3012         .. " -M " .. xdyfile
3013         .. " -L " .. language .. " /"
3014         .. opquote
3015         .. " -pdflatex="" .. latexname .. "%0 %S\" "
3016         .. sourcename..fsuffix ..".tex" ) ;
3017     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
3018 end
3019
3020
3021
3022 -- lwarpmk --version :
3023
3024 if (arg[1] == "--version") then
3025 print ( "lwarpmk: " .. printversion )
3026
3027 else -- not --version
3028
3029 -- print intro:
3030
3031 print ("lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarf package.")
3032
3033 -- lwarpmk print:
3034
3035 if arg[1] == "print" then
3036 loadconf ()
3037 if ( latexmk == "true" ) then
3038     compilelatexmk ("")
3039     print ("lwarpmk: Done.")
3040 else -- not latexmk
3041     verifyfileexists (sourcename .. ".tex") ;
3042     -- See if up to date:
3043     if (
3044         ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
3045         (
3046             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3047             lfs.attributes ( sourcename .. ".pdf" , "modification" )
```

```
3048      )
3049  ) then
3050      -- Recompile if not yet up to date:
3051      manytimes("")
3052      print ("lwarpmk: Done.") ;
3053  else
3054      print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
3055  end
3056 end -- not latexmk
3057
3058 elseif arg[1] == "print1" then
3059   loadconf ()
3060   verifyfileexists (sourcename .. ".tex") ;
3061   onetime("")
3062   print ("lwarpmk: Done.") ;
3063
3064 -- lwarf printindex:
3065 -- Compile the index then touch the source
3066 -- to trigger a recompile of the document:
3067
3068 elseif arg[1] == "printindex" then
3069 loadconf ()
3070 print ("lwarpmk: Processing the index.")
3071 os.execute(
3072   xindycmd
3073   .. " -M " .. xdyfile
3074   .. " -L " .. language
3075   .. " " .. sourcename .. ".idx")
3076 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3077 refreshdate ()
3078 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3079 print ("lwarpmk: Done.")
3080
3081 -- lwarf printglossary:
3082 -- Compile the glossary then touch the source
3083 -- to trigger a recompile of the document:
3084
3085 elseif arg[1] == "printglossary" then
3086 loadconf ()
3087 print ("lwarpmk: Processing the glossary.")
3088
3089 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
3090   " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
3091   .. sourcename .. ".glo")
3092 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3093 refreshdate ()
3094 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3095 print ("lwarpmk: Done.")
3096
3097 -- lwarpmk html:
```

```
3098
3099 elseif arg[1] == "html" then
3100 loadconf ()
3101 if ( latexmk == "true" ) then
3102     compilelatexmk ("_html")
3103     pdftohtml ()
3104     print ("lwarfpmk: Done.")
3105 else -- not latexmk
3106     verifyfileexists ( sourcename .. ".tex" ) ;
3107     -- See if exists and is up to date:
3108     if (
3109         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
3110         (
3111             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3112             lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
3113         )
3114     ) then
3115         -- Recompile if not yet up to date:
3116         manytimes("_html")
3117         pdftohtml ()
3118         print ("lwarfpmk: Done.")
3119     else
3120         print ("lwarfpmk: " .. homehtmlfilename .. ".html is up to date.")
3121     end
3122 end -- not latexmk
3123
3124 elseif arg[1] == "html1" then
3125     loadconf ()
3126     verifyfileexists ( sourcename .. ".tex" ) ;
3127     onetime("_html")
3128     pdftohtml ()
3129     print ("lwarfpmk: Done.")
3130
3131 elseif arg[1] == "pdftohtml" then
3132     loadconf ()
3133     pdftohtml ()
3134
3135 -- lwarfpmk htmlindex:
3136 -- Compile the index then touch the source
3137 -- to trigger a recompile of the document:
3138
3139 elseif arg[1] == "htmlindex" then
3140 loadconf ()
3141 print ("lwarfpmk: Processing the index.")
3142 os.execute(
3143     xindycmd
3144     .. " -M " .. xdyfile
3145     .. " -L " .. language
3146     .. " " .. sourcename .. "_html.idx"
3147 )
```

```
3148 print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
3149 refreshdate ()
3150 print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
3151 print ("lwarpmk: Done.")
3152
3153 -- lwarpmk htmlglossary:
3154 -- Compile the glossary then touch the source
3155 -- to trigger a recompile of the document:
3156
3157 elseif arg[1] == "htmlglossary" then
3158 loadconf ()
3159 print ("lwarpmk: Processing the glossary.")
3160
3161 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " ..sourcename ..
3162     "_html -t " .. sourcename .. "_html.glg -o " ..sourcename ..
3163     "_html.gls " ..sourcename .. "_html.glo")
3164
3165 print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
3166 refreshdate ()
3167 print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
3168 print ("lwarpmk: Done.")
3169
3170 -- lwarpmk limages:
3171 -- Scan the lateximages.txt file to create lateximages,
3172 -- then touch the source to trigger a recompile.
3173
3174 elseif arg[1] == "limages" then
3175 loadconf ()
3176 print ("lwarpmk: Processing images.")
3177 createlateximages ()
3178 print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
3179 refreshdate ()
3180 print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
3181 print ("lwarpmk: Done.")
3182
3183 -- lwarpmk again:
3184 -- Touch the source to trigger a recompile.
3185
3186 elseif arg[1] == "again" then
3187 loadconf ()
3188 print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
3189 refreshdate ()
3190 print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
3191 print ("lwarpmk: Done.")
3192
3193 -- lwarpmk clean:
3194 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*., .gl*
3195
3196 elseif arg[1] == "clean" then
3197 loadconf ()
```

```

3198 removeaux ()
3199 print ("lwarpmk: Done.")
3200
3201 -- lwarpmk cleanall
3202 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.* , .gl*
3203 --     and also project.pdf, *.html
3204
3205 elseif arg[1] == "cleanall" then
3206 loadconf ()
3207 removeaux ()
3208 os.execute ( rmname .. " " ..
3209     sourcename .. ".pdf" .. sourcename .. "_html.pdf" ..
3210     "*.html"
3211     )
3212 print ("lwarpmk: Done.")
3213
3214 -- lwarpmk with no argument :
3215
3216 elseif (arg[1] == nil) then
3217 printhelp ()
3218
3219 -- lwarpmk -h or lwarpmk --help :
3220
3221 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
3222 printusage ()
3223
3224 else
3225 print ("lwarpmk: Unknown command \"..arg[1].."."\n")
3226 printhelp ()
3227 end
3228
3229 end -- not --version
3230 \end{filecontents*}
3231 % \end{Verbatim}%
3232 \end{LWR@createlwarpmk}

```

## 32 Stacks

**for HTML output:** 3233 \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:

```
3234 \newcommand*{\LWR@depthnone}{-5}
```

all sectioning depths are deeper than \LWR@depthfinished:

```
3235 \newcommand*{\LWR@depthfinished}{-4}
3236 \newcommand*{\LWR@depthpart}{-1}
3237 \newcommand*{\LWR@depthchapter}{0}
3238 \newcommand*{\LWR@depthsection}{1}
3239 \newcommand*{\LWR@depthsubsection}{2}
3240 \newcommand*{\LWR@depthsubsubsection}{3}
3241 \newcommand*{\LWR@depthparagraph}{4}
3242 \newcommand*{\LWR@depth subparagraph}{5}
```

used by \itemize, \enumerate, \description:

```
3243 \newcommand*{\LWR@depthlist}{6}
```

used by \item:

```
3244 \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.

```
3245 \newcommand*{\LWR@closeone}{}% top of the stack
3246 \newcommand*{\LWR@closetwo}{}%
3247 \newcommand*{\LWR@closethree}{}%
3248 \newcommand*{\LWR@closefour}{}%
3249 \newcommand*{\LWR@closefive}{}%
3250 \newcommand*{\LWR@closesix}{}%
```

---

```

3251 \newcommand*{\LWR@closeseven}{}
3252 \newcommand*{\LWR@closeeight}{}
3253 \newcommand*{\LWR@closenine}{}
3254 \newcommand*{\LWR@closeten}{}
3255 \newcommand*{\LWR@closeeleven}{}
3256 \newcommand*{\LWR@closetwelve}{}

```

### 32.3 Closing depths

A stack to record the depth of each level:

-  Note that nested L<sup>A</sup>T<sub>E</sub>X structures may push depths which are non-sequential.

*Ex:*

---

```

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

```

---

```

3257 \newcommand*{\LWR@closedepthonen}{\LWR@depthnone}% top of the stack
3258 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
3259 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
3260 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
3261 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
3262 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
3263 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
3264 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
3265 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
3266 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}
3267 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
3268 \newcommand*{\LWR@closedepthtwelve}{\LWR@depthnone}

```

### 32.4 Pushing and popping the stack

\pushclose {<action>} {<depth>}

Pushes one return action and its L<sup>A</sup>T<sub>E</sub>X depth onto the stacks.

```
3269 \NewDocumentCommand{\pushclose}{m m}
```

```

3270 {
3271 \let\LWR@closetwelve\LWR@closeeleven
3272 \let\LWR@closeeleven\LWR@closeten
3273 \let\LWR@closeten\LWR@closenine
3274 \let\LWR@closenine\LWR@closeeight
3275 \let\LWR@closeeight\LWR@closeseven
3276 \let\LWR@closeseven\LWR@closesix
3277 \let\LWR@closesix\LWR@closefive
3278 \let\LWR@closefive\LWR@closefour
3279 \let\LWR@closefour\LWR@closethree
3280 \let\LWR@closethree\LWR@closetwo
3281 \let\LWR@closetwo\LWR@closeone
3282 \let\LWR@closeone#1
3283 \let\LWR@closedepthtwelve\LWR@closedeptheleven
3284 \let\LWR@closedepthelevel\LWR@closedepthten
3285 \let\LWR@closedepthten\LWR@closedepthnine
3286 \let\LWR@closedepthnine\LWR@closedeptheight
3287 \let\LWR@closedeptheight\LWR@closedepthseven
3288 \let\LWR@closedepthseven\LWR@closedepthsix
3289 \let\LWR@closedepthsix\LWR@closedepthfive
3290 \let\LWR@closedepthfive\LWR@closedepthfour
3291 \let\LWR@closedepthfour\LWR@closedepththree
3292 \let\LWR@closedepththree\LWR@closedepthtwo
3293 \let\LWR@closedepthtwo\LWR@closedepthonne
3294 \let\LWR@closedepthonne#2
3295 }

```

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

```

3296 \newcommand*{\popclose}{}
3297 {
3298 \let\LWR@closeone\LWR@closetwo
3299 \let\LWR@closetwo\LWR@closethree
3300 \let\LWR@closethree\LWR@closefour
3301 \let\LWR@closefour\LWR@closefive
3302 \let\LWR@closefive\LWR@closesix
3303 \let\LWR@closesix\LWR@closeseven
3304 \let\LWR@closeseven\LWR@closeeight
3305 \let\LWR@closeeight\LWR@closenine
3306 \let\LWR@closenine\LWR@closeten
3307 \let\LWR@closeten\LWR@closeeleven
3308 \let\LWR@closeeleven\LWR@closetwelve
3309 \let\LWR@closedepthonne\LWR@closedepthtwo
3310 \let\LWR@closedepthtwo\LWR@closedepththree
3311 \let\LWR@closedepththree\LWR@closedepthfour
3312 \let\LWR@closedepthfour\LWR@closedepthfive
3313 \let\LWR@closedepthfive\LWR@closedepthsix
3314 \let\LWR@closedepthsix\LWR@closedepthseven
3315 \let\LWR@closedepthseven\LWR@closedeptheight

```

---

```

3316 \let\LWR@closedeptheight\LWR@closedepthnine
3317 \let\LWR@closedepthnine\LWR@closedephten
3318 \let\LWR@closedephten\LWR@closedeptheleven
3319 \let\LWR@closedeptheleven\LWR@closedepthtwelve
3320 }

3321 \end{warpHTML}

```

## 33 Data arrays

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

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

To assign an empty value:

```
\LWR@setexpparray{name}{index}{}
```

**for HTML output:** 3322 `\begin{warpHTML}`

```

\lwr@setexpparray {\langle name\rangle} {\langle index\rangle} {\langle contents\rangle}

3323 \NewDocumentCommand{\LWR@setexpparray}{m m m}{%
3324 \ifthenelse{\isempty{#3}}{%
3325 {\csdef{#1#2}{} }{%
3326 {\expandafter\edef\csname #1#2\endcsname{\expandonce{#3}}}{%
3327 }

```

  

```

\lwr@getexpparray {\langle name\rangle} {\langle index\rangle}

3328 \newcommand*{\LWR@getexpparray}[2]{\csuse{#1#2} }

3329 \end{warpHTML}

```

## 34 HTML entities

**for HTML output:** 3330 `\begin{warpHTML}`

HTML entities and HTML Unicode entities:

```
3331 \let\lwr@origampersand\&
```

```
\HTMLentity  {\<entitytag>}

3332 \newcommand*{\HTMLentity}[1]{%
3333 % \LWR@traceinfo{HTMLentity \detokenize{\#1}}%
3334 \begingroup%
3335 \LWR@FBcancel%
3336 \LWR@origampersand#1;%
3337 \endgroup%
3338 % \LWR@traceinfo{HTMLentity done}%
3339 }

\HTMLunicode  {\<hex_unicode>}

3340 \newcommand*{\HTMLunicode}[1]{\HTMLentity{\#x#1}}


\&

3341 \renewcommand*{\&}{\HTMLentity{amp}}


\textless
\textgreater
3342 \let\LWR@origtextless\textless
3343 \renewcommand*{\textless}{\HTMLentity{lt}}
3344
3345 \let\LWR@origtextgreater\textgreater
3346 \renewcommand*{\textgreater}{\HTMLentity{gt}}


3347 \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:** 3348 `\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.

3349 `\providecommand*{\BaseJobname}{\jobname}`

**\HTMLFilename** The prefix for all generated HTML files other than the home page, defaulting to empty. See section 6.7.

```
3350 \providecommand*{\HTMLFilename}{}%
```

**\HomeHTMLFilename** The filename of the home page, defaulting to the \BaseJobname. See section 6.7.

```
3351 \providecommand*{\HomeHTMLFilename}{\BaseJobname}
```

**\SetHTMLFileName** {*number*}

Sets the file number for the next file to be generated. 0 is the home page. Use just before the next sectioning command, and set it to one less than the desired number of the next section. May be used to generate numbered groups of nodes such as 100+ for one chapter, 200+ for another chapter, etc.

```
3352 \newcommand*{\SetHTMLFileName}[1]{%
3353 \setcounter{LWR@htmlfilename}{#1}%
3354 }
```

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

Defaults to use section names in the filenames.

```
3355 \newbool{FileSectionNames}
3356 \booltrue{FileSectionNames}
```

```
3357 \end{warpall}
```

**for HTML output:** 3358 \begin{warpHTML}

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

```
3359 \newcounter{LWR@htmlfilename}
3360 \setcounter{LWR@htmlfilename}{0}
```

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

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

```
3361 \newcommand*{\LWR@htmlsectionfilename}[1]{%
3362 \LWR@traceinfo{\LWR@htmlsectionfilename A}%
}
```

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

```

3363 \LWR@traceinfo{about to assign temp}%
3364 \edef\tempone{\#1}%
3365 \LWR@traceinfo{about to compare with ??}%
3366 \ifthenelse{\equal{\LWR@tempone}{??}}{%
3367 {\LWR@traceinfo{found ??}}{%
3368 {\LWR@traceinfo{not found ??}}{%
3369 \LWR@traceinfo{about to compare with zero or empty}%
3370 \ifthenelse{%
3371     \equal{\LWR@tempone}{0}}{%
3372     \OR \equal{\LWR@tempone}{}}{%
3373     \OR \equal{\LWR@tempone}{??}}{%
3374 }{%
3375 {%
3376     \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
3377     \HomeHTMLFilename.html%
3378 }%

```

For a  $\text{\TeX}$  section named “Index” or “index” without a prefix, create a filename with a leading underscore to avoid colliding with the `HTML filename index.html`:

```

3379 {%
3380     \LWR@traceinfo{LWR@htmlsectionfilename C \LWR@tempone}%
3381     \ifthenelse{%
3382         \equal{\HTMLFilename}{}}{\AND}{%
3383         \equal{\LWR@tempone}{Index}}{\OR}{%
3384         \equal{\LWR@tempone}{index}}{%
3385     }{%
3386     {%
3387         \LWR@traceinfo{prefixing the index name with an underscore.}%
3388         \#1.html%
3389     }%

```

Otherwise, create a filename with the chosen prefix:

```

3390     {\HTMLFilename#1.html}%
3391 }%
3392 \LWR@traceinfo{LWR@htmlsectionfilename Z}%
3393 }

```

`\LWR@htmlrefsectionfilename {<label>}`

Prints the filename for the given label

```

3394 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
3395 \LWR@traceinfo{LWR@htmlrefsectionfilename A: !#1!}%
3396 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%

```

---

```

3397 \LWR@traceinfo{LWR@htmlrefsectionfilename B}%
3398 }

3399 \end{warpHTML}

```

## 36 Homepage link

**for HTML output:** 3400 \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.

```

3401 \newcommand*\{\LinkHome\}{%
3402 \LWR@subhyperrefclass{%
3403 \HomeHTMLFilename.html}%
3404 {\Home}{linkhome}%
3405 }

```

\LWR@topnavigation Creates a link to the homepage at the top of the page for use when the window is too narrow for the sidetoc.

```

3406 \newcommand*\{\LWR@topnavigation\}%
3407 \LWR@htmlelementclassline{nav}{topnavigation}\{\LinkHome\}
3408 }

```

\LWR@botnavigation Creates a link to the homepage at the bottom of the page for use when the window is too narrow for the sidetoc.

```

3409 \newcommand*\{\LWR@botnavigation\}%
3410 \LWR@htmlelementclassline{nav}{botnavigation}\{\LinkHome\}
3411 }

```

```
3412 \end{warpHTML}
```

## 37 \PrintStack diagnostic tool



Diagnostics tool: Prints the  $\text{\TeX}$  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:** 3413 \begin{warpHTML}

\PrintStack Prints the closedepth stack.

```

3414 \newcommand*{\PrintStack}{%
3415 \LWR@startpars
3416 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
3417 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
3418 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
3419 \LWR@closedepthten{} \LWR@closedeptheleven{} \LWR@closedepthtwelve{}
3420 }

3421 \end{warpHTML}

```

## 38 Closing stack levels

**for HTML output:** 3422 \begin{warpHTML}

Close one nested level:

```

3423 \newcommand*{\LWR@closeoneprevious}{%
3424
3425 \LWR@closeone{}
3426
3427 \popclose{}
3428 }

```

\LWR@closeoneprevious {<depth>} Close everything up to the given depth:

```
3429 \newcommand*{\LWR@closeoneprevious}[1]{
```

Close any pending paragraph:

```
3430 \LWR@stoppars
```

Close anything nested deeper than the desired depth:

```

3431 \whiledo{\not{(\LWR@closedepthone<#1\})}}{\LWR@closeoneprevious}
3432 }

```

```
3433 \end{warpHTML}
```

## 39 PDF pages and styles

**for HTML output:** 3434 \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.

```
3435 \newcommand{\LWR@forcenewpage}{%
3436 \LWR@stoppars\LWR@orignewpage\LWR@startpars%
3437 }
```

`\pagestyle`, etc. are nullified for HTML output.

```
\pagestyle {<style>}
```

```
3438 \renewcommand*{\pagestyle}[1]{}
```

```
\thispagestyle {<style>}
```

```
3439 \renewcommand*{\thispagestyle}[1]{}
```

```
\markboth {<left>} {<right>}
```

```
3440 \renewcommand*{\markboth}[2]{}
```

```
\markright {<right>}
```

```
3441 \renewcommand*{\markright}[1]{}
```

```
\raggedbottom
```

```
3442 \renewcommand*{\raggedbottom}{}{}
```

```
\flushbottom
```

```
3443 \renewcommand*{\flushbottom}{}{}
```

```
\sloppy
```

```
3444 \renewcommand*{\sloppy}{}{}
```

```
\fussy
```

```
3445 \renewcommand*{\fussy}{}{}
```

```
\pagenumbering {⟨commands⟩}

3446 \renewcommand*{\pagenumbering}[1]{}

3447 \end{warpHTML}
```

## 40 HTML tags, spans, divs, elements

**for HTML output:** 3448 \begin{warpHTML}

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

```
3449 \newcommand*{\LWR@tagtitle}{h1}
3450 \newcommand*{\LWR@tagtitleend}{/h1}
3451 \newcommand*{\LWR@tagpart}{h2}
3452 \newcommand*{\LWR@tagpartend}{/h2}
3453 \newcommand*{\LWR@tagchapter}{h3}
3454 \newcommand*{\LWR@tagchapterend}{/h3}
3455 \newcommand*{\LWR@tagsection}{h4}
3456 \newcommand*{\LWR@tagsectionend}{/h4}
3457 \newcommand*{\LWR@tagsubsection}{h5}
3458 \newcommand*{\LWR@tagsubsectionend}{/h5}
3459 \newcommand*{\LWR@tagsubsubsection}{h6}
3460 \newcommand*{\LWR@tagsubsubsectionend}{/h6}
3461 \newcommand*{\LWR@tagparagraph}{span class="paragraph"{}}
3462 \newcommand*{\LWR@tagparagraphend}{/span}
3463 \newcommand*{\LWR@tag subparagraph}{span class="subparagraph"{}}
3464 \newcommand*{\LWR@tag subparagraphend}{/span}
3465
3466 \newcommand*{\LWR@tagregularparagraph}{p}
```

### 40.2 Babel-French

Adjust babel-french for HTML spaces. So far, this only works for pdflatex and xelatex.

(Based on original code by DANIEL FLIPO.)

```
3467 \providecommand*{\LWR@FBcancel}{}%
3468
3469 \AtBeginDocument{%
3470 \@ifundefined{frenchbsetup}{%
3471 }{}}
```

```

3472 {%
3473     \frenchbsetup{FrenchFootnotes=false}%

3474 %
3475     \LetLtxMacro{\LWR@FBcancel}{\NoAutoSpacing}%
3476     \renewcommand*{\FBcolonspace}{%
3477         \begingroup%
3478         \LWR@FBcancel%
3479         \LWR@origampersand{}nbsp;%
3480         \endgroup%
3481     }%
3482     \renewcommand*{\FBthinspace}{%
3483         \begingroup%
3484         \LWR@FBcancel%
3485         \LWR@origampersand{\#x202f;}~, \,
3486         \endgroup%
3487     }%
3488     \renewcommand*{\FBguillspace}{%
3489         \begingroup%
3490         \LWR@FBcancel%
3491         \LWR@origampersand{}nbsp;~, for \og xyz \fg{}%
3492         \endgroup%
3493     }%
3494     \DeclareDocumentCommand{\FBmedkern}{.}{%
3495         \begingroup%
3496         \LWR@FBcancel%
3497         \LWR@origampersand{\#x202f;}~, \,
3498         \endgroup%
3499     }%
3500     \DeclareDocumentCommand{\FBthickkern}{.}{%
3501         \begingroup%
3502         \LWR@FBcancel%
3503         \LWR@origampersand{}nbsp;~%
3504         \endgroup%
3505     }%
3506     \renewcommand*{~}{\HTMLentity{nbsp}}% was overwritten by babel-french
3507     \ifFBunicode%
3508     \else%
3509         \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}%
3510         \DeclareTextCommandDefault{\FBtextellipsis}{\textellipsis\xspace}%
3511     \fi%
3512 }%
3513 }

```

### 40.3 HTML tags

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

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

```
3514 \newcommand*{\LWR@htmltagc}[1]{%
3515 {%
3516 % \LWR@traceinfo{\LWR@htmltagc \detokenize{#1}}%
3517 \begingroup%
3518 \LWR@FBcancel%
3519 \ifmmode\else\protect\LWR@origttfamily\fi%
3520 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
3521 \endgroup%
3522 }%
3523 }
```

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.

```
3524 \newcommand*{\LWR@nestspanitem}{%
3525 \if@newlist\else{\LWR@htmltagc{br /}}\fi%
3526 \LWR@origitem%
3527 }
3528
3529 \newenvironment*{\LWR@nestspan}
3530 {%
3531 \LWR@traceinfo{\LWR@nestspan}%
3532 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}>0}{%
3533 {%
3534 \LWR@traceinfo{\LWR@nestspan: inside a lateximage}%
3535 }%
3536 {%
3537 \LWR@traceinfo{\LWR@nestspan: NOT inside a lateximage}%
3538 \addtocounter{\LWR@spandepth}{1}%
3539 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{}{%
3540 \RenewDocumentEnvironment{BlockClass}{o m}{}{%
3541 \renewcommand{\BlockClassSingle}[2]{##2}%
3542 \renewcommand{\LWR@forcenewpage}{}%
3543 \renewcommand{\LWR@itemizestart}{%
3544 \let\item\LWR@nestspanitem%
3545 }%
3546 \renewcommand{\LWR@itemizeend}{}%
3547 \renewcommand{\LWR@enumeratestart}{%
3548 \let\item\LWR@nestspanitem%
3549 }%
3550 \renewcommand{\LWR@enumerateend}{}%
3551 \renewcommand{\LWR@descriptionstart}{%
3552 \let\item\LWR@nestspanitem%
```

```

3553      }%
3554      \renewcommand{\LWR@descriptionend}{}}%
3555 }% not in a lateximage
3556 }% starting env
3557 {%- ending env
3558 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}>}{0}{}%
3559 {}%
3560 {\addtocounter{LWR@spandepth}{-1}}%
3561 \LWR@traceinfo{LWR@nestspan: done}%
3562 }%
3563
3564 \AfterEndEnvironment{LWR@nestspan}{\global\let\par\LWR@closeparagraph}

```

\LWR@htmlspan {*tag*} {*text*}

 \LWR@spandepth is used to ensure that paragraph tags are not generated inside a span. The exact sequence of when to add and subtract the counter is important to correctly handle the paragraph tags before and after the span.

```

3565 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
3566 \LWR@ensuredoingapar%
3567 \LWR@htmtagc{#1}%
3568 \begin{LWR@nestspan}%
3569 #2%
3570 \LWR@htmtagc{/#1}%
3571 \end{LWR@nestspan}%
3572 }

```

\LWR@htmlspanclass [*style*] {*class*} {*text*}

```

3573 \NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{%
3574 \LWR@traceinfo{LWR@htmlspanclass #2}%
3575 \LWR@ensuredoingapar%
3576 \LWR@subhtmlelementclass{span}[#1]{#2}%
3577 \begin{LWR@nestspan}%
3578 #3%
3579 \LWR@htmtagc{/span}%
3580 \LWR@traceinfo{LWR@htmlspanclass done}%
3581 \end{LWR@nestspan}%
3582 }

```

\LWR@htmtag {*tag*}

Print an HTML tag: <*tag*>

```

3583 \newcommand*{\LWR@htmtagb}[1]{%
3584 \LWR@htmtagc{#1}%

```

```
3585 \endgroup%
3586 }
3587
3588 \newcommand*{\LWR@htmltag}{%
3589 \begingroup\catcode`\_=12
3590 \LWR@htmltagb%
3591 }
```

## 40.4 Block tags and comments

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

```
\LWR@htmlopencomment
\LWR@htmlclosecomment
3592 \newcommand*{\LWR@htmlopencomment}{%
3593 {%
3594 % \LWR@traceinfo{\LWR@htmlopencomment}%
3595 \begingroup%
3596 \LWR@FBcancel%
3597 \ifmmode\else\protect\LWR@origttfamily\fi%
3598 \LWR@origtextless{}!{-}{-}%
3599 \endgroup%
3600 }%
3601 }
3602
3603 \newcommand*{\LWR@htmlclosecomment}{%
3604 {%
3605 % \LWR@traceinfo{\LWR@htmlclosecomment}%
3606 \begingroup%
3607 \LWR@FBcancel%
3608 \ifmmode\else\protect\LWR@origttfamily\fi%
3609 {-}{-}\LWR@origtextgreater{}%
3610 \endgroup%
3611 }%
3612 }

\LWR@htmlcomment {<comment>}
3613 \newcommand{\LWR@htmlcomment}[1]{%
3614 \LWR@htmlopencomment{}%
3615 {%
3616 \LWR@origttfamily% break ligatures
3617 #1%
3618 }%
3619 \LWR@htmlclosecomment{}}
```

```
\LWR@htmlblockcomment {⟨comment⟩}

3620 \newcommand{\LWR@htmlblockcommentb}[1]
3621 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
3622
3623 \newcommand{\LWR@htmlblockcomment}
3624 {%
3625 \begingroup\catcode`\_=12%
3626 \LWR@htmlblockcommentb%
3627 }
```

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

```
3628 \newcommand*\LWR@htmlblocktag[1]{%
3629 \LWR@stoppars%
3630 \LWR@htmntag{#1}%
3631 \LWR@startpars%
3632 }
```

## 40.5 Div class and element class

```
\LWR@subhtmlelementclass {⟨element⟩} [⟨style⟩] {⟨class⟩}
```

Factored and reused in several places.

The trailing spaces allow more places for a line break.

```
3633 \NewDocumentCommand{\LWR@subhtmlelementclass}{m o m}{%
3634 \LWR@traceinfo{\LWR@subhtmlelementclass #1 #3}%
3635 \IfValueTF{#2}{%
3636 {%
3637   \ifthenelse{\equal{#2}{}}{%
3638     {\LWR@htmntag{#1 class="#3"}% empty option
3639     {\LWR@htmntag{#1 class="#3" style="#2"}% non-empty option
3640 }% option
3641 {\LWR@htmntag{#1 class="#3"}% no option
3642 \LWR@traceinfo{\LWR@subhtmlelementclass done}%
3643 }}
```

```
\LWR@htmlelementclass {⟨element⟩} {⟨class⟩} [⟨style⟩]
```

```
3644 \NewDocumentCommand{\LWR@htmlelementclass}{m o m}{%
3645 \LWR@stoppars%
3646 \LWR@subhtmlelementclass{#1}[#2]{#3}%
3647 \LWR@startpars%
3648 }
```

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

3649 \newcommand*{\LWR@htmlelementclassend}[2]{%
3650 \LWR@stoppars%
3651 \LWR@htmltag{/\#1}%
3652 \ifbool{HTMLDebugComments}{%
3653   \LWR@htmlcomment{End of #1 ``#2''}%
3654 }{}%
3655 \LWR@startpars%
3656 }

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

3657 \NewDocumentCommand{\LWR@htmldivclass}{o m}{%
3658 \LWR@htmlelementclass{div}[\#1]{\#2}%
3659 }

\LWR@htmldivclassend  {\langle class\rangle}

3660 \newcommand*{\LWR@htmldivclassend}[1]{%
3661 \LWR@htmlelementclassend{div}{\#1}%
3662 }
```

## 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}

3663 \NewDocumentCommand{\LWR@htmlelementclassline}{m o m +m}{%
3664 \LWR@stoppars%
3665 \LWR@subhtmlelementclass{\#1}{\#2}{\#3}%
3666 #4%
3667 \LWR@htmltag{/\#1}%
3668 \LWR@startpars%
3669 }
```

## 40.7 HTML5 semantic elements

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

```

3670 \newcommand*\LWR@htmlelement}[1]{%
3671 \LWR@htmlblocktag{#1}%
3672 }

\LWR@htmlelementend {<element>}

3673 \newcommand*\LWR@htmlelementend}[1]{%
3674 \LWR@stoppars
3675 \LWR@htmlltag{/#1}
3676 \LWR@startpars
3677 }
3678
3679 \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** [*style*] {*class*} High-level interface for <div> classes.

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

**for HTML output:** 3680 \begin{warpHTML}
3681 \NewDocumentEnvironment{BlockClass}{o m}%
3682 {
3683 \LWR@htmldivclass[#1]{#2}%
3684 }
3685 {
3686 \LWR@htmldivclassend{#2}%
3687 }
3688 \end{warpHTML}

**for PRINT output:** 3689 \begin{warpprint}
3690 \NewDocumentEnvironment{BlockClass}{o m}{}{}%
3691 \end{warpprint}

**\BlockClassSingle** {*class*} {*text*} A single-line <div>, without a paragraph tag for the line of text.

**for HTML output:** 3692 \begin{warpHTML}
3693 \newcommand{\BlockClassSingle}[2]{%
3694 \LWR@htmlelementclassline{div}{#1}{#2}}%

```

3695 }
3696 \end{warpHTML}

for PRINT output: 3697 \begin{warpprint}
3698 \newcommand{\BlockClassSingle}[2]{#2}
3699 \end{warpprint}

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

for HTML output: 3700 \begin{warpHTML}
3701 \NewDocumentCommand{\InlineClass}{o m +m}{%
3702 \LWR@htmlspanclass[#1]{#2}{#3}%
3703 }%
3704 \end{warpHTML}

for PRINT output: 3705 \begin{warpprint}
3706 \NewDocumentCommand{\InlineClass}{o m +m}{#3}%
3707 \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: 3708 \begin{warpHTML}
3709 \NewDocumentEnvironment{LWR@BlockClassWP}{m m m}{%
3710 }%
3711 \LWR@stoppars%
3712 \ifbool{FormatWP}{%
3713 }%
3714 \LWR@maybeinthisfloat%
3715 \LWR@htmlltag{%
3716 div class="#3" %
3717 id="autofloat-\arabic{LWR@thisfloat}"%
3718 \ifblank{#1}{}{ style="#1"}%
3719 }%
3720 }% FormatWP
3721 }% not FormatWP
3722 \LWR@htmlltag{%
3723 div class="#3"%
3724 \ifblank{#2}{}{ style="#2"}%
3725 }%
3726 }% not FormatWP
3727 \LWR@startpars%
3728 }%
3729 {\LWR@htmldivclassend{#3}}
3730 \end{warpHTML}

for PRINT output: 3731 \begin{warpprint}
3732 \NewDocumentEnvironment{LWR@BlockClassWP}{m m m}{ }{ }%
3733 \end{warpprint}

```

## 40.9 Closing HTML tags

**for HTML output:** 3734 \begin{warpHTML}

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

```
3735 \newcommand*{\LWR@printclosepart}
3736   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
3737 \newcommand*{\LWR@printclosechapter}
3738   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
3739 \newcommand*{\LWR@printclosesection}
3740   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
3741 \newcommand*{\LWR@printclosesubsection}
3742   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
3743 \newcommand*{\LWR@printclosesubsubsection}
3744   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
3745 \newcommand*{\LWR@printcloseparagraph}
3746   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
3747 \newcommand*{\LWR@printclosesubparagraph}
3748   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}
```

Lists require closing HTML tags:

```
3749 \newcommand*{\LWR@printcloselistitem}
3750   {\LWR@htmlltag{/li}}
3751 \newcommand*{\LWR@printclosedescitem}
3752   {\LWR@htmlltag{/dd}}
3753 \newcommand*{\LWR@printcloseitemize}
3754   {\LWR@htmlltag{/ul}}
3755 \newcommand*{\LWR@printcloseenumerate}
3756   {\LWR@htmlltag{/ol}}
3757 \newcommand*{\LWR@printclosedescription}
3758   {\LWR@htmlltag{/dl}}
```

3759 \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  $\text{\LaTeX}$  paragraph is being used (\LWR@doingapar). \LWR@lateximagedepth is used to prevent nesting

tags inside a `\teximage`. `LWR@spandepth` is used to prevent nesting paragraph tags inside a paragraph, which became important inside `\fbox` commands and other spans.

**for HTML output:** 3760 `\begin{warpHTML}`

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

```
3761 \newcounter{LWR@spandepth}
3762 \setcounter{LWR@spandepth}{0}
```

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

```
3763 \newbool{LWR@doingstartpars}
3764 \boolfalse{LWR@doingstartpars}
```

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

```
3765 \newbool{LWR@doingapar}
3766 \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:

```
3767 \newcommand*\LWR@ensuredoingapar{%
3768 \ifbool{LWR@doingstartpars}{%
3769 {\global\booltrue{LWR@doingapar}}{%
3770 {}{%
3771 }}
```

`\LWR@openparagraph`

```
3772 \newcommand*\LWR@openparagraph{%
3773 {}%
```

See if paragraph handling is enabled:

```
3774 \ifbool{LWR@doingstartpars}{%
3775 {}% handling pars}
```

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

```
3776 \ifthenelse{%
3777   \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3778   \cnttest{\value{LWR@spandepth}}{>}{0}%
3779 }% nested par tags?
```

If so: Do nothing if already started a `\teximage` page. Cannot nest a `\teximage`. Also do nothing if already inside a `<span>`. Do not nest paragraph tags inside a `<span>`.

```
3780      {}% no nested par tags
```

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

```
3781      {}% yes nest par tags
3782          \LWR@htmltagc{\LWR@tagregularparagraph}{}%
```

Manually indent item list labels to avoid left margin intrustion:

See if are nested inside an item list:

```
3783          \ifnumcomp{\@listdepth}{>}{0}%
3784              {}%
```

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

```
3785          \LWR@orighspace{1in}%
3786          }%
3787          {}%
```

Now have started a paragraph.

```
3788          \global\booltrue{\LWR@doingapar}{}%
```

At the endof each paragraph, generate closing tag and do regular /par stuff. (Attempting to use the everyhook cr hook for `\LWR@closeparagraph` does not work well.)

```
3789          \let\par\LWR@closeparagraph%
3790          }% end of yes nest par tags
3791 }% end of handling pars
3792 {}% not handling pars
3793 }
```

`\LWR@closeparagraph`

```
3794 \newcommand*{\LWR@closeparagraph}%
3795 {}%
```

See if paragraph handling is enabled:

```
3796 \ifbool{\LWR@doingapar}{}%
```

If currently in paragraph mode:

```
3797 {%
  handling pars
```

See if already started a `\teximage` or a `<span>`:

```
3798     \ifthenelse{%
 3799       \cnttest{\value{LWR@lateximagedepth}}>}{0} \OR%
 3800       \cnttest{\value{LWR@spandepth}}>}{0}%
 3801   }%
```

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

```
3802     {%
 3803       no nested par tags
 3804       \ifthenelse{%
 3805         \cnttest{\value{LWR@spandepth}}>}{0}%
 3806         \AND%
 3807         \cnttest{\value{LWR@lateximagedepth}}=}{0}%
 3808         \ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmlltagc{br /}}%
 3809       }%
 3810     }%
 3811     no nested par tags
```

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

```
3811     {%
 3812       yes nest par tags
```

Print a closing tag and some extra vertical space:

```
3812     \unskip%
 3813     \LWR@htmlltagc{/LWR@tagregularparagraph}%
 3814     \LWR@orignewline%
```

No longer doing a paragraph:

```
3815     \global\boolfalse{LWR@doingapar}%
 3816 % Disable the special \env{minipage} \& \cs{hspace} interaction
 3817 % until a new minipage is found:
 3818 %   \begin{macrocode}
 3819     \global\boolfalse{LWR@minipagethispar}%
 3820   }%
 3821 }%
 3822 }%
 3823 }%
```

Add a `\parbreak` if in a `<span>`, but not in a table outside a row:

```
3822 {%
 3823   not handling pars
 3824   \ifthenelse{\cnttest{\value{LWR@spandepth}}>}{0}{}%
```

---

```

3824     {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}}%
3825     {}%
3826 }% not handling pars

```

Finish with regular paragraph processing

```

3827 \LWR@origpar%
3828 }

```

```
3829 \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:** 3830 \begin{warpHTML}

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

```

3831 \newcommand*{\LWR@startpars}{%
3832   {}%

```

Ignore if inside a span:

```

3833 \ifthenelse{\cnttest{\value{LWR@spandepth}}>0}{%
3834   {}%
3835   {}%

```

See if currently handling HTML paragraphs:

```
3836 \ifbool{LWR@doingstartpars}{%
```

If already in paragraph mode, do nothing.

```
3837   {}%
```

If not currently in paragraph mode:

```
3838   {}%
```

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

```
3839 \PushPreHook{par}{\LWR@openparagraph}%
```

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

```
3840 \let\par\LWR@closeparagraph
3841
3842 }% an intentionally blank line
```

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

```
3843 \global\setbool{\LWR@doingstartpars}{true}%
```

No <par> tag yet to undo:

```
3844 \global\boolfalse{\LWR@doingapar}%
3845 }% nestspan
3846 }
```

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

```
3847 \newcommand*{\LWR@stopars}%
3848 {%
```

Ignore if inside a span:

```
3849 \ifthenelse{\cnttest{\value{\LWR@spandepth}}{>}{0}}%
3850 {}%
3851 {}%
```

See if currently handling HTML paragraphs:

```
3852 \ifbool{\LWR@doingapar}{}
```

if currently in an HTML paragraph:

```
3853 {%
```

Print a closing tag:

```
3854 \unskip%
3855 \LWR@htmlltagc{/LWR@tagregularparagraph}%
3856 \LWR@orignewline%
```

No longer have an open HTML paragraph:

```
3857 \global\boolfalse{LWR@doingapar}%
```

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

```
3858 \global\boolfalse{LWR@minipagethispar}
3859
3860 }% an intentionally blank line
```

If was not in an HTML paragraph:

```
3861 {}%
```

See if currently allowing HTML paragraphs:

```
3862 \ifbool{LWR@doingstartpars}{}%
```

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

```
3863 {\ClearPreHook{par}}%
```

Else: do nothing

```
3864 {}%
```

no longer in paragraph mode

```
3865 \global\setbool{LWR@doingstartpars}{false}%
```

no `<p>` tag to undo:

```
3866 \global\boolfalse{LWR@doingapar}%
3867 }% nestspan
3868 }
```

```
3869 \end{warpHTML}
```

## 43 Page headers and footers

**for HTML & PRINT:** 3870 `\begin{warpall}`

In the following, catcode is manually changes back and forth without groups, since new macros are being defined which must not be contained within the groups.

```
3871 \newcommand{\LWR@firstpagetop}{} % for the home page alone
3872 \newcommand{\LWR@pagetop}{} % for all other pages
3873 \newcommand{\LWR@pagebottom}{}
3874
3875 \newcommand{\LWR@setfirstpagetopb}[1]{%
3876 \renewcommand{\LWR@firstpagetop}{#1}
3877 \catcode`\_=8
3878 }

\HTMLFirstPageTop {<text and logos>}

3879 \newcommand{\HTMLFirstPageTop}{%
3880 \catcode`\_=12
3881 \LWR@setfirstpagetopb
3882 }

3883 \newcommand{\LWR@setpagetopb}[1]{%
3884 \renewcommand{\LWR@pagetop}{#1}
3885 \catcode`\_=8
3886 }

\HTMLPageTop {<text and logos>}

3887 \newcommand{\HTMLPageTop}{%
3888 \catcode`\_=12
3889 \LWR@setpagetopb
3890 }

3891 \newcommand{\LWR@setpagebottomb}[1]{%
3892 \renewcommand{\LWR@pagebottom}{#1}
3893 \catcode`\_=8
3894 }

\HTMLPageBottom {<text and logos>}

3895 \newcommand{\HTMLPageBottom}{%
3896 \catcode`\_=12
3897 \LWR@setpagebottomb
3898 }

3899 \end{warpall}
```

## 44 CSS

**for HTML output:** 3900 \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.

```
3901 \newcommand*{\LWR@currentcss}{lwarp.css}
```

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

```
3902 \newcommand*{\LWR@newcssb}[1]{%
3903 \renewcommand*{\LWR@currentcss}{#1}
3904 \catcode`\_=8
3905 }
3906
3907 \newcommand*{\CSSFilename}{
3908 \catcode`\_=12
3909 \LWR@newcssb
3910 }
3911 \end{warpHTML}
```

**for PRINT output:** 3912 \begin{warpprint}
3913 \newcommand\*{\CSSFilename}[1]{}
3914 \end{warpprint}

## 45 HTML meta description and author

**for HTML & PRINT:** 3915 \begin{warpall}

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

```
3916 \providecommand{\theauthor}{}
3917 \newcommand{\theHTMLAuthor}{\theauthor}
3918
3919 \newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}
3920 \end{warpall}
```

**for HTML & PRINT:** 3921 \begin{warpall}

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



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

---

\HTMLDescription {*New HTML meta description.*} Assigns the HTML file's description meta tag.

```

3922 \newcommand{\LWR@currentHTMLDescription}{}
3923
3924 \newcommand{\HTMLDescription}[1]{%
3925 \renewcommand{\LWR@currentHTMLDescription}{#1}
3926 }
3927
3928 \end{warpall}
```

## 46 Footnotes

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

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

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 L<sup>A</sup>T<sub>E</sub>X box \LWR@footnotes. Using this instead of the original \footins box avoids having footnotes be printed by the output routine, since footnotes should be printed per HTML page instead of per PDF page.

See section 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 output:** 3929 \begin{warpHTML}

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

3930 \newbox\LWR@footnotes

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

\@makefntext {<text>}

3931 \long\def\@makefntext#1{\textsuperscript{\@thefnmark} #1}

\@makefnmark

3932 \def\@makefnmark{\hbox{\textsuperscript{\@thefnmark}}}

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

In a `lateximage` during HTML output, the `lateximage` is placed inside a print-mode `minipage`, but the footnotes are broken out by:

```
\def\@mpfn{footnote}
\def\thempfn{\thefootnote}
\let\@footnotetext\LWR@footnotetext
```

\LWR@footnotetext {<text>}

3933 \long\def\LWR@footnotetext#1{%
3934 \global\setbox\LWR@footnotes=\vbox{%

Add to any current footnotes:

3935 \unvbox\LWR@footnotes%

Remember the footnote number for \ref:

```
3936 \protected\edef\@currentlabel{%
3937 \csname p@footnote\endcsname\@thefnmark%
3938 }% @currentlabel
```

Open a group:

3939 \color@begingroup%

Use HTML superscripts even inside a `lateximage`:

```
3940     \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

Use paragraph tags if in a tabular data cell or a `lateximage`:

```
3941     \ifthenelse{%
3942         \boolean{LWR@doingstartpars} \AND%
3943         \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3944     }{%
3945     {}%
3946     {\LWR@htmllagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
3947     \@makefntext{#1}%
```

Closing paragraph tag:

```
3948     \ifthenelse{%
3949         \boolean{LWR@doingstartpars} \AND%
3950         \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3951     }{%
3952     {\par}%
3953     {}%
3954     {\LWR@htmllagc{/}\LWR@tagregularparagraph}%
3955     \LWR@orignewline%
3956 }
```

Close the group:

```
3957     \color@endgroup%
3958 }% vbox
3959 }%
```

```
\@footnotetext {<text>}

3960 \let\@footnotetext\LWR@footnotetext
```

## 46.5 Minipage footnote implementation

```
\@mpfootnotetext {<text>}

3961 \long\def\@mpfootnotetext#1{%
3962 \global\setbox\@mpfootins\vbox{%
```

```

3963   \unvbox\@mpfootins
3964   \reset@font\footnotesize
3965   \hsize\columnwidth
3966   \parboxrestore
3967   \protected@edef\@currentlabel
3968   {\csname p@mpfootnote\endcsname\@thefnmark}%
3969   \color@begingroup

```

Use paragraph tags if in a tabular data cell or a `lateximage`:

```

3970   \ifthenelse{%
3971     \boolean{LWR@doingstartpars} \AND%
3972     \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3973   }%
3974   {}%
3975   {\LWR@htmlltagc{\LWR@tagregularparagraph}}%

3976   \makefntext{%
3977     \ignorespaces#1%
3978   }%

```

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

```

3979   \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{%
3980   {}%
3981   {}%
3982   \LWR@htmlltagc{/}\LWR@tagregularparagraph}%
3983   \LWR@orignewline%
3984   }%
3985   \color@endgroup%
3986 }% vbox
3987 }

```

## 46.6 Printing pending footnotes

`\LWR@printpendingfootnotes` Enclose the footnotes in a class, print, then clear.

```

3988 \newcommand*{\LWR@printpendingfootnotes}{%
3989 \ifvoid\LWR@footnotes\else
3990   \LWR@forcenewpage
3991   \begin{BlockClass}{footnotes}
3992   \LWR@origmedskip
3993   \unvbox\LWR@footnotes
3994   \setbox\LWR@footnotes=\vbox{}
3995   \end{BlockClass}
3996 \fi

```

---

3997 }

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

```
3998 \newcommand*\LWR@epubprintpendingfootnotes{%
3999 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}{%
4000 {\LWR@printpendingfootnotes}%
4001 {}%
4002 }%
4003 \end{warpHTML}
```

## 47 Marginpars

**for HTML output:** 4004 \begin{warpHTML}

```
\marginpar [⟨left⟩] {⟨right⟩}

4005 \renewcommand{\marginpar}[2] []{%
4006 \ifbool{FormatWP}{%
4007 {}%
4008 \begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{}{\marginblock}%
4009 #2
4010 \end{LWR@BlockClassWP}%
4011 }{%
4012 {\LWR@htmlspanclass{\marginpar}{#2}}%
4013 }%
4014 \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:** 4015 \begin{warpall}

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

```
4016 \newcounter{FileDepth}
4017 \setcounter{FileDepth}{-5}
```

Bool **CombineHigherDepths** Combine higher-level sections together into one file?

```
4018 \newbool{CombineHigherDepths}
4019 \booltrue{CombineHigherDepths}

4020 \end{warpall}
```

**for HTML output:** 4021 \begin{warpHTML}

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

```
4022 \newcommand*{\LWR@thisfilename}{}%
```

\LWR@thisnewfilename The filename being sanitized.

```
4023 \newcommand*{\LWR@thisnewfilename}{}%
```

\LWR@filenamenoblocks {*filename*}

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

 Be sure that this does not result in filename collisions! Use the optional TOC caption entry parameter for formatting. Remember to \protect \TeX commands which appear in section names and TOC captions.

```
4024 \newcommand*{\LWR@filenamenoblocks}[1]{%
4025 \begingroup
```

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

```
4026 \LWR@nullfonts
4027 \renewcommand*{\LWR@htmlltagc}[1]{}%
```

Replaces common symbols and short words with hyphens:

```
4028 \edef\LWR@thisnewfilename{\#1}
4029 \fullexpandarg
```

Convert spaces into hyphens:

```
4030 \StrSubstitute{\LWR@thisfilename}{ }{-}[\LWR@thisfilename]
```

Convert punctuation into hyphens:

```
4031 \StrSubstitute{\LWR@thisfilename}{,}{-}[\LWR@thisfilename]
4032 \StrSubstitute{\LWR@thisfilename}{'}{-}[\LWR@thisfilename]
4033 \StrSubstitute{\LWR@thisfilename}{%
4034     {\LWR@origampersand}{-}}[\LWR@thisfilename]
4035 \StrSubstitute{\LWR@thisfilename}{+}{-}[\LWR@thisfilename]
4036 \StrSubstitute{\LWR@thisfilename}{,}{-}[\LWR@thisfilename]
4037 \StrSubstitute{\LWR@thisfilename}{/}{-}[\LWR@thisfilename]
4038 \StrSubstitute{\LWR@thisfilename}{:}{-}[\LWR@thisfilename]
4039 \StrSubstitute{\LWR@thisfilename}{;}{-}[\LWR@thisfilename]
4040 \StrSubstitute{\LWR@thisfilename}{=}{-}[\LWR@thisfilename]
4041 \StrSubstitute{\LWR@thisfilename}{?}{-}[\LWR@thisfilename]
4042 \StrSubstitute{\LWR@thisfilename}{@}{-}[\LWR@thisfilename]
4043 \StrSubstitute{\LWR@thisfilename}{"}{-}[\LWR@thisfilename]
4044 \StrSubstitute{\LWR@thisfilename}{%
4045     {\textless}{-}}[\LWR@thisfilename]
4046 \StrSubstitute{\LWR@thisfilename}{%
4047     {\textgreater}{-}}[\LWR@thisfilename]
4048 \StrSubstitute{\LWR@thisfilename}{\#}{-}[\LWR@thisfilename]
4049 \StrSubstitute{\LWR@thisfilename}{\%}{-}[\LWR@thisfilename]
4050 \StrSubstitute{\LWR@thisfilename}{\{}{-}[\LWR@thisfilename]
4051 \StrSubstitute{\LWR@thisfilename}{\}}{-}[\LWR@thisfilename]
4052 \StrSubstitute{\LWR@thisfilename}{\|}{-}[\LWR@thisfilename]
4053 \StrSubstitute{\LWR@thisfilename}{%
4054     {\textbackslash}{-}}[\LWR@thisfilename]
4055 \StrSubstitute{\LWR@thisfilename}{^}{-}[\LWR@thisfilename]
4056 \StrSubstitute{\LWR@thisfilename}{\~}{-}[\LWR@thisfilename]
4057 %      "\~{}" for babel
4058 \StrSubstitute{\LWR@thisfilename}{\{}{-}[\LWR@thisfilename]
4059 \StrSubstitute{\LWR@thisfilename}{\}}{-}[\LWR@thisfilename]
4060 \StrSubstitute{\LWR@thisfilename}{\`}{-}[\LWR@thisfilename]
```

Convert short words:

```
4061 \StrSubstitute{\LWR@thisfilename}{-s-}{-}[\LWR@thisfilename]
4062 \StrSubstitute{\LWR@thisfilename}{-S-}{-}[\LWR@thisfilename]
4063 \StrSubstitute{\LWR@thisfilename}{-a-}{-}[\LWR@thisfilename]
4064 \StrSubstitute{\LWR@thisfilename}{-A-}{-}[\LWR@thisfilename]
4065 \StrSubstitute{\LWR@thisfilename}{-an-}{-}[\LWR@thisfilename]
4066 \StrSubstitute{\LWR@thisfilename}{-AN-}{-}[\LWR@thisfilename]
4067 \StrSubstitute{\LWR@thisfilename}{-to-}{-}[\LWR@thisfilename]
4068 \StrSubstitute{\LWR@thisfilename}{-TO-}{-}[\LWR@thisfilename]
4069 \StrSubstitute{\LWR@thisfilename}{-by-}{-}[\LWR@thisfilename]
4070 \StrSubstitute{\LWR@thisfilename}{-BY-}{-}[\LWR@thisfilename]
```

```

4071 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
4072 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
4073 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
4074 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
4075 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
4076 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
4077 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
4078 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

4079 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
4080 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
4081 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
4082 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
4083 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
4084 %      emdash
4085 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
4086 %      endash
4087 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
4088 \endgroup
4089 }

```

\LWR@newhtmlfile {*section name*}

Finishes the current HTML page with footnotes, footer, navigation, then starts a new HTML page with an HTML comment telling where to split the page and what the new filename and css are, then adds navigation, side TOC, header, and starts the text body.

```
4090 \newcommand*{\LWR@newhtmlfile}[1]{
```

At the bottom of the ending file:

```

4091 \LWR@htmlelementclassend{section}{textbody}
4092
4093 \LWR@printpendingfootnotes
4094

```

No footer between files if EPUB:

```

4095 \ifbool{FormatEPUB}
4096 {}
4097 {
4098     \LWR@htmlelement{footer}
4099
4100     \LWR@pagebottom
4101

```

```
4102     \LWR@htmlelementend{footer}
4103 }
```

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

```
4104 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4105 {}
4106 {\ifnumcomp{\value{LWR@htmlfilename}}{>}{0}{\LWR@botnavigation}{}}
```

End of this HTML file:

```
4107 \LWR@stopars
4108 \LWR@htmlltag{/body}\LWR@orignewline
4109 \LWR@htmlltag{/html}\LWR@orignewline
4110 \LWR@orignewline
4111
4112 \addtocounter{LWR@htmlfilename}{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.

```
4113 \ifbool{FileSectionNames}%
4114 {\LWR@filenamenoblanks{\#1}}
4115 {\renewcommand*{\LWR@thisfilename}{\theLWR@htmlfilename}}
```

Include an HTML comment to instruct lwarfpmk 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.

```
4116 \LWR@htmlblockcomment{%
4117 |Start file|%
4118 \LWR@htmlsectionfilename{\LWR@thisfilename}|%
4119 }
```

At the top of the starting file:

```
4120 \LWR@stopars
4121
4122 \LWR@filestart{ -- #1}%
4123 there is an EMdash in front of the #1
```

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

```
4124 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
```

```
4125 {}
4126 {\LWR@topnavigation}
4127
```

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

```
4128 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4129 {}
4130 {
4131     \LWR@htmlelement{header}
4132
4133     \LWR@pagetop
4134
4135     \LWR@htmlelementend{header}
4136 }
4137
```

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

```
4138 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4139 {}
4140 {\ifcvoid{\thetitle}{}{\LWR@printthetitle}}
4141
```

No sidetoc if formatting for an EPUB or word processor:

```
4142 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4143 {}
4144 {\LWR@sidetoc}
4145
```

Start of the <textbody>:

```
4146 \LWR@htmlelementclass{section}{textbody}
4147
```

Keep paragraph tags disabled for now:

```
4148 \LWR@stoppars
4149
```

Track the page numbers:

```
4150 \setcounter{LWR@latestautopage}{\value{page}}
4151 }

4152 \end{warpHTML}
```

## 49 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing L<sup>A</sup>T<sub>E</sub>X code and packages. Formatting is handled by css, so the emulated code has much less work to do than the print versions.

- Unicode** Section names and the resulting filenames with accented characters are partially supported, depending on the ability of pdflatex to generate characters and pdftotext to read them. If extra symbols appear in the text, it may be that pdflatex is actually producing a symbol over or under a character, resulting in pdftotext picking up the accent symbol separately.

X<sub>PL</sub>L<sup>A</sup>T<sub>E</sub>X and LuaL<sup>A</sup>T<sub>E</sub>X directly support accented section and file names.

**for HTML output:** 4153 \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.

```
4154 \newbool{LWR@forcinghtmlpage}
4155 \boolfalse{LWR@forcinghtmlpage}
4156
4157 \newcommand*{\ForceHTMLPage}{%
4158 \global\booltrue{LWR@forcinghtmlpage}%
4159 }
```

- \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 tocbind. Also see \ForceHTMLPage.

For print output, no effect.

```
4160 \newbool{LWR@forcinghtmltoc}
4161 \boolfalse{LWR@forcinghtmltoc}
4162
4163 \newcommand*{\ForceHTMLTOC}{%
4164 \global\booltrue{LWR@forcinghtmltoc}%
4165 }
```

```

4166 \end{warpHTML}

for PRINT output: 4167 \begin{warpprint}
4168 \newcommand*{\ForceHTMLPage}(){}
4169 \newcommand*{\ForceHTMLTOC}){}
4170 \end{warpprint}

for HTML output: 4171 \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.

```

4172 \newbool{LWR@mainmatter}
4173 \DeclareDocumentCommand{\mainmatter}{}{%
4174 \booltrue{LWR@mainmatter}%
4175 }

```

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

```

4176 \DeclareDocumentCommand{\frontmatter}{}{%
4177 \boolfalse{LWR@mainmatter}%
4178 }

```

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

```

4179 \DeclareDocumentCommand{\backmatter}{}{%
4180 \boolfalse{LWR@mainmatter}%
4181 }

```

## 49.3 Sectioning support macros

**\LWR@sectionnumber** {*<section type>*}

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

```

4182 \newcommand*{\LWR@sectionnumber}[1]{%
4183 \InlineClass{sectionnumber}{#1}%
4184 }

```

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

---

\LWR@createautosec {*<section type>*}

Create an autosection tag.

```
4185 \newcommand*{\LWR@createautosec}[1]{%
4186 \LWR@htmlltag{#1 id="autosec-\thepage"}{}%
4187 }
```

\LWR@pushoneclose {{*depth*} }{{*printclose*}} Stacks the new sectioning level's closing tag, to be used when this section is closed some time later.

 \LWR@stoppars must be executed first.

```
4188 \NewDocumentCommand{\LWR@pushoneclose}{m m}{\pushclose{#2}{#1}}
```

\LWR@startnewdepth {{*depth*} }{{*printclose*}}

Closes currently stacked tags of a lesser level, then opens the new nesting level by saving this new sectioning level's closing tag for later use.

 \LWR@stoppars must be executed first.

```
4189 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
4190 \LWR@closeprevious{#1}%
```

Push a new section depth:

```
4191 \LWR@pushoneclose{#1}{#2}%
4192 }
```

Ctr LWR@prevFileDepth Remembers the previous LWR@FileDepth.

Initialized to a deep level so that any section will trigger a new HTML page after the home page.

```
4193 \newcounter{LWR@prevFileDepth}
4194 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

\@secCntFormat {{*sectiontype*}}

```
4195 \def\@secCntFormat#1{\csname the#1\endcsname\protect\quad}
```

\simplechapterdelim Used by tocbibind and anonchap.

```
4196 \newcommand*\simplechapterdelim{}
```

\@chapcntformat {\<sectiontype>}

\let to \@seccntformat by default, but may be redefined by \simplechapter and \restorechapter from tocbibind or anonchap.

```
4197 \let\@chapcntformat\@seccntformat
```

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

The common actions for the high-level sectioning commands.

```
4198 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
4199 \LWR@traceinfo{LWR@section |#2| |#3|}%
4200 \% \ifthenelse{%
4201 \% % \equal{#2}{\-NoValue-}\AND%
4202 \% \isempty{#3}%
4203 \% }%
4204 \% empty section name
4205 \% \LWR@traceinfo{LWR@section: empty section}%
4206 \% empty section name
4207 \% not an empty section name
4208 \LWR@traceinfo{LWR@section: not an empty section}%
4209 \LWR@stopars%
```

Cancel special minipage horizontal space interaction:

```
4210 \global\boolfalse{LWR@minipagethispar}%
```

Start a new HTML file unless starred, and if is a shallow sectioning depth.

Exception: Also start a new HTML file for \part\*, for appendix.

Generate a new L<sup>A</sup>T<sub>E</sub>X page so that toc and index page number points to the section:

```
4211 \LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
4212 \IfBooleanTF{#1}{\LWR@traceinfo{LWR@section: starred}}{}%
4213 \ifbool{LWR@forcinghtmlpage}{\LWR@traceinfo{LWR@section: forcinghtmlpage}}{}%
4214 \ifthenelse{%
4215   \(%
4216     \(\NOT\equal{#1}{\BooleanTrue}\)\OR%
4217     \(\cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}\)\OR%
4218     \(\boolean{LWR@forcinghtmlpage}\)%
4219   \)%
4220   \AND%
```

```

4221      \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}%
4222      \AND%
4223      \(%
4224          \NOT\boolean{CombineHigherDepths}\OR%
4225          \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}%
4226      \)%
4227      \AND%
4228      \NOT\equal{#3}{}% phantomsection
4229 }%

```

If so: start a new HTML file:

```

4230 {%
4231     \LWR@traceinfo{LWR@section: new HTML file}%

```

See if there was an optional TOC name entry:

```
4232     \IfNoValueTF{#2}{%
```

If no optional entry

```
4233     {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```

4234     {\LWR@newhtmlfile{#2}}%
4235 }% new file

```

Else: No new HTML file:

```
4236 {%
4237     \LWR@traceinfo{LWR@section: not a new HTML file}%
4238     \LWR@orignewpage%
4239 }% not new file

```

Remember this section's name for \nameref:

```

4241 \IfValueT{#3}{%
4242 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
4243 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
4244 }%

```

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

```

4245
4246 \ifbool{HTMLDebugComments}{%
4247   \LWR@htmlcomment{Opening #4 ``#3''{}}%
4248 }{}%
4249

```

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

```

4250 \ifthenelse{%
4251   \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
4252 }%
4253 {\LWR@startpars}%
4254 }%

```

Create the opening tag with an autosec:

```
4255 \LWR@createautosec{\csuse{LWR@tag#4}}%
```

Check if starred:

```

4256 \IfBooleanTF{#1}%
4257 {%

```

Starred, but also forcing a TOC entry, so add unnumbered TOC name or regular name:

```

4258 \ifbool{LWR@forcinghtmltoc}{%
4259 {\addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}}%
4260 {}}%
4261 }% starred

```

Not starred, so step counter and add to TOC:

```
4262 {%
not starred
}
```

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

```

4263 \ifthenelse{%
4264   \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
4265 }%
4266 {%
if secnumdepth
}

```

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

```

4267 \LWR@traceinfo{LWR@section: about to test main matter}%
4268 \ifbool{LWR@mainmatter}{%
4269 {%
}

```

```

4270          \LWR@traceinfo{LWR@section: yes mainmatter}%
4271          \refstepcounter{#4}%

```

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

```

4272          \LWR@traceinfo{LWR@section: about to addcontentsline}%
4273          \addcontentsline{toc}{#4}%
4274          {%
4275              \protect\numberline{\csuse{the#4}}%
4276              {\ignorespaces\IfValueTF{#2}{#2}{#3}\protect\relax}%
4277          }%
4278          \LWR@traceinfo{LWR@section: finished addcontentsline}%
4279      }% end of if main matter

```

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

```

4280      {%
4281          \LWR@traceinfo{LWR@section: no main matter}%
4282          \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4283      }% end of not main matter
4284  }% end of secnumdepth

```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```

4285      {%
4286          \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4287      }%

```

For part, print the section type:

```

4288      \ifbool{LWR@mainmatter}%
4289      {%
4290          \ifthenelse{%
4291              \(\cnttest{\csuse{LWR@depth#4}}{<=}\}%
4292                  {\value{secnumdepth}}\)\ AND%
4293              \(\cnttest{\csuse{LWR@depth#4}}{<=}\{\LWR@depthpart\}\)%
4294          }%
4295          {\csuse{#4name}~{}}
4296      }%

```

Print the section number:

```

4297          \LWR@traceinfo{LWR@section: about to print section number}%
4298          \ifthenelse{%
4299              \cnttest{\csuse{LWR@depth#4}}{<=}\{\value{secnumdepth}\}%
4300          }%
4301          {%
4302              \ifstreq{\#4}{chapter}%
4303                  {\protect\LWR@sectionnumber{\chapcntformat{#4}}}%

```

---

```

4304      {\protect\LWR@sectionnumber{\@secntformat{#4}}}\%
4305      }%
4306      {}%
4307      \LWR@traceinfo{\LWR@section: finished print section number}%
4308      }{}%
4309 }% end of not starred

```

Print the section name:

```
4310 #3%
```

Close the heading tag, such as /H2:

```
4311 \LWR@htmlltag{\csuse{\LWR@tag#4end}}%
```

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

```
4312 \label{autopage-\thepage}%
```

Start paragraph handing unless is an inline paragraph or subparagraph:

```

4313 \ifthenelse{%
4314     \cnttest{\csuse{\LWR@depth#4}}{<}{\LWR@depthparagraph}%
4315 }%
4316 {\LWR@startpars}%
4317 }%

```

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.

```

4318 \ifthenelse{%
4319     \NOT\equal{#1}{\BooleanTrue}\OR%
4320     \cnttest{\csuse{\LWR@depth#4}}{=}{\LWR@depthpart}%
4321 }%
4322 {}% not starred
4323     \setcounter{LWR@prevFileDepth}{\csuse{\LWR@depth#4}}%
4324 }% not starred
4325 }%

```

Reset to defaults if not a phantomsection:

```
4326 \ifstrempty{#3}%
4327 {}%
4328 {%
4329 \global\boolearnfalse{LWR@forcinghtmlpage}%
4330 \global\boolearnfalse{LWR@forcinghtmltoc}%
4331 }%
4332 %
4333 % }% not an empty section name
4334 \LWR@traceinfo{LWR@section: done}%
4335 }
```

## 49.4 \section and friends

```
\part * [<TOC name>] {<name>}
4336 \DeclareDocumentCommand{\part}{s o m}{%
4337 \LWR@epubprintpendingfootnotes%
4338 \LWR@stoppars%
4339
4340 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
4341
4342 \LWR@section{#1}{#2}{#3}{part}%
4343 }

\chapter * [<TOC name>] {<name>}
4344 \let\@printcites\relax% for quotchap package
4345
4346 \@ifundefined{chapter}%
4347 {}%
4348 {%
4349 \DeclareDocumentCommand{\chapter}{s o m}{%
4350 \LWR@traceinfo{chapter #3}%
4351 \LWR@epubprintpendingfootnotes%
4352 \LWR@stoppars%
4353
4354 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
4355
4356 \LWR@section{#1}{#2}{#3}{chapter}%
4357 \@printcites% for quotchap package
4358 }
4359 }

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

```
4360 \DeclareDocumentCommand{\section}{s o m}{%
4361   \LWR@epubprintpendingfootnotes%
4362   \LWR@stoppars%
4363
4364   \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
4365
4366   \LWR@section{#1}{#2}{#3}{section}%
4367 }

\subsection * [<TOC name>] {<name>}

4368 \DeclareDocumentCommand{\subsection}{s o m}{%
4369   \LWR@epubprintpendingfootnotes%
4370   \LWR@stoppars%
4371
4372   \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
4373
4374   \LWR@section{#1}{#2}{#3}{subsection}%
4375 }

\subsubsection * [<TOC name>] {<name>}

4376 \DeclareDocumentCommand{\subsubsection}{s o m}{%
4377   \LWR@epubprintpendingfootnotes%
4378   \LWR@stoppars%
4379
4380   \LWR@startnewdepth{\LWR@depthsubsubsection}{%
4381     \LWR@printclosesubsubsection}%
4382
4383   \LWR@section{#1}{#2}{#3}{subsubsection}%
4384 }

\paragraph * [<TOC name>] {<name>}

4385 \DeclareDocumentCommand{\paragraph}{s o m}{%
4386   \LWR@epubprintpendingfootnotes%
4387   \LWR@stoppars%
4388
4389   \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
4390
4391   \LWR@section{#1}{#2}{#3}{paragraph}%
4392 }

\ subparagraph * [<TOC name>] {<name>}

4393 \DeclareDocumentCommand{\ subparagraph}{s o m}{%
4394   \LWR@epubprintpendingfootnotes%
```

```
4395 \LWR@stopars%
4396
4397 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
4398
4399 \LWR@section{#1}{#2}{#3}{subparagraph}%
4400 }

4401 \end{warpHTML}
```

## 50 Starting a new file

**for HTML & PRINT:** 4402 \begin{warpall}

\HTMLLanguage Default language for the HTML lang tag.

```
4403 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
4404
4405 \newcommand*{\HTMLLanguage}[1]{%
4406 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
4407 }
```

```
4408 \end{warpall}
```

**for HTML output:** 4409 \begin{warpHTML}

\LWR@filestart {\<title\\_suffix>}

Creates the opening HTML tags.

```
4410 \newcommand*{\LWR@filestart}[1]{%
4411 \LWR@traceinfo{\LWR@filestart}}
```

Locally temporarily disable direct-formatting commands:

```
4412 \begingroup
4413 \LWR@traceinfo{\LWR@filestart: A}
4414 \LWR@nullfonts
4415 \LWR@traceinfo{\LWR@filestart: B}
```

Create the page's HTML header:

```
4416 \LWR@htmlltag{!DOCTYPE html}\LWR@orignewline
4417 \LWR@traceinfo{\LWR@filestart: C}
```

The language is user-adjustable:

```
4418 \LWR@htmltag{html lang="\LWR@currentHTMLLanguage"{} }\LWR@orignewline
```

Start of the meta data:

```
4419 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
4420 \LWR@htmltag{meta charset="UTF-8" /}\LWR@orignewline
```

Author:

```
4421 \ifcsempty{theHTMLAuthor}{}{  
4422 \LWR@htmltag{meta name="author" content="\theHTMLAuthor" /}\LWR@orignewline  
4423 }
```

lwarf is the generator:

```
4424 \LWR@htmltag{meta name="generator" content="LaTeX lwarf package" /}%  
4425     \LWR@orignewline
```

If there is a description, add it now:

```
4426 \ifdefempty{\LWR@currentHTMLDescription}{}{  
4427 \LWR@htmltag{  
4428 meta name="description" content="\LWR@currentHTMLDescription" /}%  
4429     \LWR@orignewline  
4430 }%
```

Mobile-friendly viewport:

```
4431 \LWR@htmltag{meta name="viewport" %  
4432 content="width=device-width, initial-scale=1.0" /}%  
4433     \LWR@orignewline
```

IE patch:

```
4434 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline  
4435 \LWR@htmltag{  
4436 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{} }%  
4437 \LWR@htmltag{/script}\LWR@orignewline  
4438 \LWR@htmltag{![endif]{-}{-}}\LWR@orignewline
```

The page's title:

```
4439 \ifcsvvoid{thetitle}{}{%
```

---

```
4440 \LWR@htmntag{title}\thetitle#1\LWR@htmntag{/title}\LWR@orignewline%
4441 }%
```

The page's stylesheet:

```
4442 \LWR@htmntag{%
4443 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
4444 \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.

```
4445 \ifbool{mathjax}{%
4446 {%
4447   \boolfalse{LWR@verbtags}
4448   \verbatiminput{l warp_mathjax.txt}%
4449   \booltrue{LWR@verbtags}
4450   \LWR@stoppars
4451 }% end of mathjax
4452 {}%
```

End of the header:

```
4453 \LWR@htmntag{/head}\LWR@orignewline
```

Start of the body:

```
4454 \LWR@htmntag{body}\LWR@orignewline
4455 \endgroup
4456 \LWR@traceinfo{LWR@filestart: done}
4457 }

4458 \end{warpHTML}
```

## 51 Starting HTML output

**for HTML output:** 4459 \begin{warpHTML}

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

```
4460 \catcode`\$=\active
4461 \newcommand*\{\LWR@LwarpStart}
4462 {%
4463 \LWR@traceinfo{LWR@l warpStart}
```

If formatting for a word processor, force filedepth to single-file only, force HTML debug comments off.

```
4464 \ifbool{FormatWP}{%
4465     \setcounter{FileDepth}{-5}%
4466     \boolfalse{HTMLDebugComments}%
4467 }{}}
```

Expand and detokenize \HomeHTMLFilename and \HTMLFilename:

```
4468 \edef\LWR@strresult{\HomeHTMLFilename}
4469 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
4470 \edef\LWR@strresult{\HTMLFilename}
4471 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn and empty page style:

```
4472 \LWR@origonecolumn%
4473 \LWR@origpagestyle{empty}
```

Reduce chance of line overflow in verbatim environments:

```
4474 \LWR@origscriptsize%
```

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

```
4475 \LWR@origraggedright%
4476 \LetLtxMacro{\\"}{\LWR@endofline}%
```

Spread the lines for pdftotext to read them well:

```
4477 \linespread{1.3}%
```

For pdftotext to reliably identify paragraph splits:

```
4478 \setlength{\parindent}{0pt}
4479 \setlength{\parskip}{2ex}
```

For the lateximages record file:

```
4480 \immediate\openout\LWR@lateximagesfile=lateximages.txt
```

Removes space after the caption in the HTML:

```
4481 \setlength{\belowcaptionskip}{-3ex}
```

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

---

```

4482 \renewcommand{\ps@plain}{}

\centering Not used in the HTML environment:
\raggedleft
\raggedright 4483 \renewcommand*{\centering}{}
4484 \renewcommand*{\raggedleft}{}
4485 \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:

```

4486 \LetLtxMacro{\LWR@origtabular}{\tabular}
4487 \LetLtxMacro{\LWR@origendtabular}{\endtabular}
4488 \LetLtxMacro{\tabular}{\LWR@tabular}
4489 \LetLtxMacro{\endtabular}{\endLWR@tabular}

```

Float captions:

```
4490 \let\LWR@origcaption\caption
```

Labels: `\ltx@label` is used in `amsmath` environments and is also patched by `cleveref`.

#### [Label in HTML](#)

```

4491 \let\LWR@origltx@label\ltx@label
4492 \let\ltx@label\LWR@htmlmathlabel

```

Logos:

```

4493 \let\TeX\LWR@TeX
4494 \let\LaTeX\LWR@LaTeX
4495 \let\LuaTeX\LWR@LuaTeX
4496 \let\LuaLaTeX\LWR@LuaLaTeX
4497 \let\XeTeX\LWR@XeTeX
4498 \let\XeLaTeX\LWR@XeLaTeX
4499 \let\ConTeXt\LWR@ConTeXt

```

Not yet started any paragraph handling:

```

4500 \global\boolfalse{LWR@doingapar}
4501 \global\boolfalse{LWR@doingstartpars}

```

Start a new HTML file and a header:

```

4502 \LWR@traceinfo{LWR@lwarpStart: Starting new file.}
4503 \LWR@filestart{}
4504 \LWR@traceinfo{LWR@lwarpStart: Generating first header.}

```

---

```

4505 \LWR@htmlltag{header}\LWR@orignewline
4506 \LWR@startpars
4507 \LWR@firstpagetop
4508 \LWR@stoppars
4509 \LWR@htmlltag{/header}\LWR@orignewline
4510 \LWR@traceinfo{\LWR@lwarpStart: Generating textbody.}
4511 \LWR@htmlltag{section class="textbody"{}}

```

Document and page settings:

```

4512 \mainmatter
4513 \LWR@origpagenumbering{arabic}

```

Patch the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native L<sup>A</sup>T<sub>E</sub>X environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```
4514 \LWR@patchlists
```

Ensure that math mode is active to call lwarp's patches:

```
4515 \catcode`$=\active
```

Allow HTML paragraphs to begin:

```

4516 \LWR@startpars
4517 \LWR@traceinfo{\LWR@lwarpStart: done}
4518 }
4519 \catcode`$=3% math shift until lwarp starts

4520 \end{warpHTML}

```

## 52 Ending HTML output

**for HTML output:** 4521 \begin{warpHTML}

\LWR@requesttoc {<boolean>} {<suffix>} Requests that a toc, lof, or lot be generated.

```

4522 \newcommand*{\LWR@requesttoc}[2]{%
4523 \ifbool{#1}{%
4524 {%
4525     \expandafter\newwrite\csuse{tf@#2}%
4526     \immediate\openout \csuse{tf@#2} \jobname.#2\relax
4527 }{}%
4528 }

```

\LWR@LwarpEnd Final stop of all HTML output:

```
4529 \newcommand*\LWR@LwarpEnd{  
4530 {  
4531 \LWR@stopars  
4532 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
4533 \LWR@htmlelementclassend{section}{textbody}
```

Print any pending footnotes:

```
4534 \LWR@printpendingfootnotes
```

Create the footer:

```
4535 \LWR@htmlelement{footer}  
4536  
4537 \LWR@pagebottom  
4538  
4539 \LWR@htmlelementend{footer}
```

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

Presumably has a table-of-contents.

```
4540 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}{  
4541 {}  
4542 {  
4543     \ifnumcomp{\value{LWR@htmlfilename}}{>}{0}{\LWR@botnavigation}{}  
4544 }  
  
4545 \LWR@stopars% final stop of all paragraphs
```

Finish the HTML file:

```
4546 \LWR@htmlltag{/body}\LWR@orignewline  
4547 \LWR@htmlltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
4548 \LWR@orignewpage
```

For `\teximage` commands:

```
4549 \immediate\closeout\LWR@teximagesfile
4550 }

4551 \end{warpHTML}
```

## 53 Title page

**package support** `\warp` supports the native L<sup>A</sup>T<sub>E</sub>X 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.

**affiliation** `\warp` provides for the `\author` macro an additional `\affiliation` macro to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using `titlingpage`'s `\theauthor` in the main text.

**reusing titlepage information** The `titling` package maintains the definitions of `\thetitle`, `\theauthor`, etc., after the title has been typeset. These commands are to be used to refer to the document's title and author, etc., in the main text. These definitions have the `\thanks` and `\affiliation` removed, and for `\author` the `\and` is replaced to generate a simple inline list of authors separated by commas. Note: `\theauthor` does not work well with `authblk` unless the traditional L<sup>A</sup>T<sub>E</sub>X syntax is used.

**⚠ \theauthor, authblk**

**custom titlepages** `\printtitle`, `\printauthor`, etc., are provided for use inside a custom `titlepage` or `titlingpage` environment, and these retain the `\thanks` and `\affiliation`.

**\printthanks** `\printthanks` has been added to force the printing of thanks inside a `titlingpage` environment when `\maketitle` is not used.

**⚠**

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** {<*author*>} While using `\maketitle` and print mode, the author is treated as a single-column `tabular` and the `\and` feature finishes the current `tabular` then starts a new one for the next author. Each author thus is placed into its own `tabular`, and an affiliation may be placed on its own line such as

---

```
\author{Name \\ Affiliation \and Second Name \\ Second Affiliation}
```

For HTML, the entire author block is placed inside a `<div>` of class `author`, and each individual author is inside a `<div>` of class `oneauthor`.

`\@title` `\@title`, `\@author`, etc. store the values as originally assigned, including any `\thanks`, `\and`, or `\affiliation`. These are low-level macros intended to be used by other macros only inside a `titlepage` or `titlingpage`, and are used by `\maketitle`. The author is printed inside a single-column `tabular`, which becomes multiple single-column `tabulars` if multiples authors are included. For HTML these `tabulars` become side-by-side `<div>`s of class `oneauthor`, all of which are combined into one `<div>` of class `author`.

`\printtitle` `\printtitle`, etc. are user-level macros intended to be used in custom `titlepage` or `titlingpage` environments in cases where `\maketitle` is not desired. These `\printdate` commands preserve the `\thanks`, etc., and should not be used in the main text.

`\thetitle` `\thetitle`, `\theauthor`, and `\thedate` are available if `titling` has been loaded, and are sanitized user-level versions from which have been removed the `\thanks` and `\affiliation`, and `\and` is changed for inline text usage. The author is printed inline without `\affiliation` or `\thanks`, with `\and` placing commas between multiple authors. Thus, these commands are to be used in the main text whenever the user wishes to refer to the document's title and such. One practical use for this is to place the authors at the bottom of each HTML page, such as:

`\HTMLPageBottom {<text>}`

---

```
\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 L<sup>A</sup>T<sub>E</sub>X 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 {⟨text⟩}

Adds the affiliation to the author for use in \maketitle.

Inside titlepage, this macro prints its argument. Outside, it is null.

**for HTML & PRINT:**

```
4552 \begin{warpall}
4553 \newrobustcmd{\affiliation}[1]{}
4554 \end{warpall}
```

**for PRINT output:**

```
4555 \begin{warpprint}
```

```
4556 \AtBeginEnvironment{titlepage}{
4557 \renewrobustcmd{\affiliation}[1]{\textsc{\small#1}}
4558 }
4559
4560 \AtBeginDocument{
4561 \@ifpackageloaded{titling}{
4562 \AtBeginEnvironment{titlingpage}{
4563 \renewrobustcmd{\affiliation}[1]{\textsc{\small#1}}
4564 }
4565 }{}% titling loaded
4566 }% AtBeginDocument

4567 \end{warpprint}
```

**for HTML output:**

```
4568 \begin{warpHTML}
```

Env titlepage Sets up a <div> of class titlepage.

```
4569 \renewenvironment*{titlepage}
4570 {
4571 \renewrobustcmd{\affiliation}[1]{\InlineClass{affiliation}{##1}}
4572 \LWR@printpendingfootnotes
4573 \LWR@forcenewpage
4574 \BlockClass{titlepage}
4575 }
4576 {
4577 \endBlockClass
4578 \LWR@printpendingfootnotes
4579 }
```

```
4580 \end{warpHTML}
```

### 53.3 Printing the thanks

**for HTML & PRINT:** 4581 \begin{warpall}

\printthanks Forces the \thanks to be printed.

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

4582 \newcommand\*\{\printthanks\}{\@thanks}

4583 \end{warpall}

### 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:** 4584 \begin{warpHTML}

\printtitle

```
4585 \newcommand*\{\printtitle\}
4586 {
4587 \LWR@stoppars
4588 \LWR@htmlltag{\LWR@tagtitle}%
4589 \@title%
4590 \LWR@htmlltag{\LWR@tagtitleend}
4591 \LWR@startpars
4592 }
```

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

```
4593 \newcommand*\{\LWR@printthetitle\}
4594 {
4595 \LWR@stoppars
4596 \LWR@htmlltag{\LWR@tagtitle}%
4597 \thetitle%
4598 \LWR@htmlltag{\LWR@tagtitleend}
4599 \LWR@startpars
4600 }
```

\printauthor HTML version.

4601 \newcommand\*\{\printauthor\}{

The entire author block is contained in a <div> named `author`:

```
4602 \begin{BlockClass}{author}
```

\and finishes one author and starts the next:

```
4603 \renewcommand{\and}{%
4604 \end{BlockClass}%
4605 \begin{BlockClass}{oneauthor}%
4606 }
```

Individual authors are contained in a <div> named `oneauthor`:

```
4607 \begin{BlockClass}{oneauthor}%
4608 \@author%
4609 \end{BlockClass}%
4610 \end{BlockClass}%
4611 }
```

`\printdate`

```
4612 \newcommand*{\printdate}{%
4613 \begin{BlockClass}{titledate}%
4614 \@date%
4615 \end{BlockClass}%
4616 }
```

```
4617 \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:** 4618 \begin{warpprint}

`\printtitle`

```
4619 \newcommand*{\printtitle}{\Huge\@title}
```

`\printauthor` Print mode.

```
4620 \newcommand*{\printauthor}{%
4621   {\large\begin{tabular}[t]{c}\@author\end{tabular}}}
```

---

```
\printdate
4622 \newcommand*{\printdate}{{\small\textrit{\@date}}}
4623 \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:** 4624 \begin{warpHTML}

```
4625 \if@titlepage
4626 \newcommand{\LWR@setfootnoteseries}{%
4627   \renewcommand{\thefootnote{\@arabic\c@footnote}}%
4628 }
4629 \else
4630 \newcommand{\LWR@setfootnoteseries}{%
4631   \renewcommand{\thefootnote{\@fnsymbol\c@footnote}}%
4632 }
4633 \fi
```

\LWR@maketitlesetup Patches \thanks macros.

```
4634 \newcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
4635 \LWR@setfootnoteseries%
4636 \def\@makefnmark{\textsuperscript{\thefootnote}}
\thefootnote \Rightarrow \nameuse{arabic}{footnote}, or
\thefootnote \Rightarrow \nameuse{fnsymbol}{footnote}
```

Redefine the footnote text:

```
4637 \long\def\@makefntext##1{%
```

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

```
4638 \textsuperscript{\@thefnmark} \LWR@orighspace{1in}

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

Print the text:

```
4639 ##1%
4640 }%
4641 }
```

```
\@fnsymbol {\langle counter\rangle}
```

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used \| which was not being seen by pdftotext.

```
4642 \def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
4643 \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLunicode{2016}}\or
4644 **\or \HTMLentity{dagger}\HTMLentity{dagger}\or
4645 \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.

```
4646 \renewcommand*{\maketitle}{%
```

An HTML titlepage <div> is used for all classes.

```
4647 \begin{titlepage}
```

Set up special patches:

```
4648 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4649 \@maketitle
```

Immediately generate any \thanks footnotes:

```
4650 \@thanks
```

Close the HTML titlepage div and cleanup:

```

4651 \end{titlepage}
4652 \setcounter{footnote}{0}%
4653 \global\let\thanks\relax
4654 \global\let\maketitle\relax
4655 \global\let\@maketitle\relax
4656 \global\let\@thanks\@empty
4657 \global\let\@author\@empty
4658 \global\let\@date\@empty
4659 \global\let\@title\@empty
4660 \global\let\title\relax
4661 \global\let\author\relax
4662 \global\let\date\relax
4663 \global\let\and\relax
4664 }
```

\@maketitle HTML mode. Typesets the title, etc.:

```

4665 \DeclareDocumentCommand{\@maketitle}{}{%
4666   \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}%
4667   \@title
4668   \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
4669   \begin{BlockClass}{author}
4670     \renewcommand{\and}{%
4671       \end{BlockClass}%
4672       \begin{BlockClass}{oneauthor}%
4673     }%
4674     \begin{BlockClass}{oneauthor}%
4675       \author
4676       \end{BlockClass}%
4677     \end{BlockClass}%
4678     \begin{BlockClass}{titledate}%
4679       \date
4680     \end{BlockClass}%
4681 }
```

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

```
4682 \newcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
4683 \@thanks
```

Set up special patches:

```
4684 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4685 \@maketitle
```

Immediately generate any \thanks footnotes:

```
4686 \@thanks
4687 }
```

```
4688 \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 lwarf.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. 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:** 4689 \begin{warpall}

\AddSubtitlePublished Adds \published and \subtitle, and related.

```
4690 \AfterPackage{titling}%
4691 \newcommand*\@AddSubtitlePublished{%
4692 %
4693 \newcommand{\@published}{}
4694 %
4695 \newcommand{\published}[1]{\gdef\@published{\##1}}
4696 %
4697 \renewcommand*\@maketitlehooka{\printpublished}
4698 %
4699 \newcommand*\@printpublished{%
4700 \warpprintonly{\begin{center}\unskip}%
4701 \begin{BlockClass}{published}%
4702 \warpprintonly{\large\itshape}%
4703 \@published%
4704 \end{BlockClass}%
4705 \warpprintonly{\end{center}}%
4706 }
```

```

4707
4708 \newcommand{\@subtitle}{}
4709
4710 \newcommand{\subtitle}[1]{\gdef\@subtitle{##1}}
4711
4712 \renewcommand*{\maketitlehookb}{\printsubtitle}
4713
4714 \newcommand*{\printsubtitle}{%
4715 \warpprintonly{\begin{center}\unskip}%
4716 \begin{BlockClass}{subtitle}%
4717 \warpprintonly{\Large\itshape}%
4718 \@subtitle%
4719 \end{BlockClass}%
4720 \warpprintonly{\end{center}}%
4721 }
4722
4723 }% \AddSubtitlePublished
4724 }% AfterPackage

4725 \end{warpall}

```

## 54 Abstract

The following code replaces the L<sup>A</sup>T<sub>E</sub>X default, and will itself be replaced later if the abstract package is loaded.

**for HTML output:** 4726 \begin{warpHTML}

\abstractname User-redefinable title for the abstract.

Also over-written by the babel package.

```
4727 \providecommand*{\abstractname}{Abstract}
```

Env abstract

```

4728 \DeclareDocumentEnvironment{abstract}{}{%
4729 {%
4730 \LWR@forcenewpage
4731 \BlockClass{abstract}%
4732 \BlockClassSingle{\abstracttitle}{\abstractname}%
4733 }%
4734 {%
4735 \endBlockClass
4736 }%

```

---

4737 \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:** 4738 \begin{warpHTML}  
 4739 \newcommand{\attribution}[1]{%  
 4740 \InlineClass{attribution}{--\,#1}}% emdash  
 4741 \end{warpHTML}

**for PRINT output:** 4742 \begin{warpprint}  
 4743 \newcommand{\attribution}[1]{\textsc{---\,#1}}  
 4744 \end{warpprint}

\citetitle for use inside quote, quotation, verse:

**for HTML output:** 4745 \begin{warpHTML}  
 4746 \newcommand{\citetitle}[1]{%  
 4747 \InlineClass{citetitle}{--\,#1}}% emdash  
 4748 \end{warpHTML}

**for PRINT output:** 4749 \begin{warpprint}  
 4750 \newcommand{\citetitle}[1]{\textsl{---\,#1}}  
 4751 \end{warpprint}

### 55.2 Quotes, quotations

**for HTML output:** 4752 \begin{warpHTML}

Env quote

```
4753 \renewenvironment*{quote}
4754 {
4755 \LWR@forcenewpage
4756 \LWR@htmlblocktag{blockquote}
4757 }
4758 {\LWR@htmlblocktag{/blockquote}}
```

```

4759
4760 \renewenvironment*{quotation}
4761 {
4762 \LWR@forcenewpage
4763 \LWR@htmlblocktag{blockquotation}
4764 }
4765 {\LWR@htmlblocktag{/blockquotation}}
4766 \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. `lwarf` provides `\attribution`, which works for both print and `HTML` output. To combine the two so that `\attrib` is used for print and `\attribution` is used for `HTML`:

---

```

\begin{warpHTML}
\let\attrib\attribution
\end{warpHTML}

```

---

- Len `\leftskip`  
 Len `\leftmargini`  
 Len `\TMLvleftskip`  
 Len `\TMLleftmargini`
- 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:** 4767 `\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`.

```
4768 \newlength{\HTMLvleftskip}
4769 \setlength{\HTMLvleftskip}{1em}
```

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

```
4770 \newlength{\HTMLleftmargini}
4771 \setlength{\HTMLleftmargini}{4.5em}

4772 \end{warpall}
```

## 56 Verbatim and tabbing

**for HTML & PRINT:** 4773 \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.

```
4774 \newlength{\VerbatimHTMLWidth}
4775 \setlength{\VerbatimHTMLWidth}{4in}
4776 \end{warpall}
```

**for HTML output:** 4777 \begin{warpHTML}

Bool LWR@verbtags Used to temporarily turn off verbatim tags while doing \verbatiminput in the HTML head.

```
4778 \newbool{LWR@verbtags}
4779 \booltrue{LWR@verbtags}
```

\LWR@atbeginverbatim [<style>] {<class>}

Encloses a verbatim environment with the given css class.

```
4780 \newcommand*{\LWR@atbeginverbatim}[2] []
4781 {%
```

Avoid excessive space between lines:

```
4782 \setlength{\parskip}{0ex} %
```

Stop generating HTML paragraph tags:

```
4783 \LWR@stopars%
```

Create a new pre of the given class. The tags may temporarily be turned off for internal use, such as loading the MathJax script.

```
4784 \ifbool{LWR@verbtags}{%
4785     \LWR@htmltag{pre class="#2"
4786     \ifthenelse{\equal{#1}{}}{}{style="#1"}}
4787 }% pre
4788 }{}}
```

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:

```
4789 \begingroup%
4790 \LWR@origttfamily%
4791 \LWR@FBcancel%
```

Do not produce HTML tags for `\hspace` inside a `verse` par. Restore plain L<sup>A</sup>T<sub>E</sub>X `\hspace` functionality:

```
4792 \LetLtxMacro{\hspace}{\LWR@orighspace}%
4793 }
```

`\LWR@afterendverbatim` Finishes enclosing a verbatim environment.

```
4794 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the pre:

```
4795 \endgroup%
4796 \unskip%
```

At the end of the environment, close the pre:

```
4797 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}}
4798
4799 }{}}
```

Resume regular paragraph handling:

```
4800 \LWR@startpars%
4801 }
```

`\verbatiminput {<filename>}`

Patch `\verbatiminput` to add HTML tags:

---

```

4802 \let\LWRV@origverbatim@input\verbatim@input
4803
4804 \renewcommand{\verbatim@input}[2]{%
4805 \ifbool{\LWR@verbtags}{\LWR@forcenewpage}{}{%
4806 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}{}{%
4807 \LWRV@origverbatim@input[#1]{#2}}{%
4808 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}%
4809 }{%

```

Env  **verbatim**

```

4810 \AfterEndPreamble{%
4811 \LWR@traceinfo{Patching verbatim.}%
4812 \AtBeginEnvironment{verbatim}{%
4813 \LWR@forcenewpage
4814 \LWR@atbeginverbatim{verbatim}\unskip\LWR@origvspace*{-\baselineskip}{}{%
4815 }{%
4816 \AfterEndEnvironment{verbatim}{%
4817 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}%
4818 }{%
4819 }{%

```

Env  **tabbing**

```

4820 \AfterEndPreamble{%
4821 \LWR@traceinfo{Patching tabbing.}%
4822 \AtBeginEnvironment{tabbing}{%
4823 \LWR@forcenewpage
4824 \LWR@atbeginverbatim{tabbing}\unskip\LWR@origvspace*{-\baselineskip}{}{%
4825 }{%
4826 \AfterEndEnvironment{tabbing}{%
4827 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}%
4828 }{%
4829 }{%

```

```
4830 \end{warpHTML}
```

## 57 Theorems

```
\newtheorem {⟨text⟩} [⟨counter⟩] -or- [⟨oldname⟩] {⟨text⟩}
```

A few minor changes are made to supply HTML tags.

- The entire theorem is placed into a <div> of class theoremcontents.

- The label for each theorem is placed inside a `<span>` of class `theoremlabel`.
- The contents are placed inside a `<div>` of class `theoremcontents`.

**for HTML output:** 4831 `\begin{warpHTML}`

```

\@begintheorem {⟨name⟩} {⟨number⟩}

4832 \renewcommand{\@begintheorem}[2]{%
4833 \LWR@forcenewpage
4834 \BlockClass{theoremcontents}
4835 \InlineClass{theoremlabel}{#1\ #2\ }%
4836 }

\@opargbegintheorem {⟨name⟩} {⟨number⟩} {⟨oparg⟩}

4837 \renewcommand{\@opargbegintheorem}[3]{%
4838 \LWR@forcenewpage
4839 \BlockClass{theoremcontents}
4840 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }%
4841 }

\@endtheorem

4842 \renewcommand*{\@endtheorem}{%
4843 \endBlockClass% theoremcontents
4844 }

4845 \end{warpHTML}

```

## 58 Lists

The environments `itemize`, `enumerate`, and `description` are patched when `lwarp` is started. These patches support the standard  $\text{\TeX}$  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  $\text{\TeX}$  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:** 4846 \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.

```

4847 \newcommand*\LWR@itemizeitem{%
4848 \LWR@stoppars%
4849 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem{}}%
4850 \LWR@htmlltag{li}%
4851 \LWR@startpars%
4852 \LWR@origitem%
4853 }

```

Env itemize [*options*]

```

4854 \newcommand*\LWR@itemizestart{%
4855 \LWR@stoppars%
4856 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4857 \LWR@htmlltag{ul style="list-style-type:none"}{%
4858 \LWR@startpars%
4859 \let\item\LWR@itemizeitem%
4860 }
4861
4862 \newcommand*\LWR@itemizeend{%
4863 \LWR@stoppars%
4864 \LWR@closeprevious{\LWR@depthlistitem}%
4865 \LWR@closeoneprevious{}%
4866 \LWR@startpars%

```

```
4867 }
```

## 58.2 Enumerate

An HTML unordered list is used with customized L<sup>A</sup>T<sub>E</sub>X-generated labels.

```
Env  enumerate  [<options>]

4868 \newcommand*{\LWR@enumeratestart}{%
4869 \LWR@stoppars%
4870 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4871 \LWR@htmlltag{ul style="list-style-type:none"}{%
4872 \LWR@startpars%
4873 \let\item{\LWR@itemizeitem}%
4874 }%
4875
4876 \newcommand*{\LWR@enumerateend}{%
4877 \LWR@stoppars%
4878 \LWR@closeprevious{\LWR@depthlistitem}%
4879 \LWR@closeoneprevious{}}%
4880 \LWR@startpars%
4881 }
```

## 58.3 Description

\LWR@descitem [<label>] Handles an \item inside a description.

```
4882 \newcommand*{\LWR@descitem}[1] []%
4883 {%
4884 \LWR@stoppars%
4885 \LWR@setlatestname{#1}%
4886 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem{}}%
```

Temporarily disable \hspace, which article.cls, etc. use per \item for descriptions only. This causes l warp to mistakenly place an empty span between HTML list tags.

```
4887 \LetLtxMacro{\hspace}{\LWR@nohspace}%
```

Process the original \item code:

```
4888 \LWR@origitem[]%
```

Restore \hspace for use in the item text:

```
4889 \LetLtxMacro{\hspace}{\LWR@hspace}%
```

Be sure the label doesn't print to the left of the rest of the file:

```
4890 \LWR@orighspace{in}
4891 \LWR@htmlltag{dt}#1\LWR@htmlltag{/dt}%
4892 \LWR@orignewline%
4893 \LWR@htmlltag{dd}%
4894 \LWR@startpars%
4895 }
```

**Env description [*options*]**

```
4896 \newcommand*{\LWR@descriptionstart}{%
4897 \LWR@stoppars%
4898 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}%
4899 \LWR@htmlltag{dl}%
4900 \LWR@startpars%
4901 \let\item{\LWR@descitem}%
4902 }
4903
4904 \newcommand*{\LWR@descriptionend}{%
4905 \LWR@stoppars%
4906 \LWR@closeprevious{\LWR@depthlistitem}%
4907 \LWR@closeoneprevious{}%
4908 \LWR@startpars%
4909 }
```

**\LWR@patchlists** Patches list environments.

\LWR@patchlists remembers \item as defined by whatever packages have been loaded, then patches the itemize, enumerate, and description environments and \item. This works with the native L<sup>A</sup>T<sub>E</sub>X environments, as well as those provided by enumitem, enumerate, and paralist.

```
4910 \newcommand*{\LWR@patchlists}{%
4911 \LetLtxMacro{\LWR@origitem}{\item}
4912 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
4913 \AtEndEnvironment{itemize}{\LWR@itemizeend}
4914 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
4915 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
4916 \AtBeginEnvironment{description}{\LWR@descriptionstart}
4917 \AtEndEnvironment{description}{\LWR@descriptionend}
4918 }

4919 \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 &

- 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}
{
\begin{tabular}{cc}
left & right \\
\end{tabular}
}
{
\begin{TabularMacro}\ResumeTabular
left & right \\
\end{tabular}
}
\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 lwarf, so \warpprintonly and \warpHTMLonly must be used to make versions for print and HTML.

- See section 178.2 for \multicolumnrow.

lwarf 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.

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}[0pt]{Text} & ...
... & \mcolrowcell & \mcolrowcell & ...
... & \mcolrowcell & \mcolrowcell & ...
```

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

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

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

### Column specifiers:

 \* column specification

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

@ and !

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

\multirow

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

 \newcolumntype

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

**Rules:****vertical rules**

- 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:**

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

**longtable headings**

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

⚠ S columns

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

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

## 59.2 Token lookahead

Used by \LWR@futurenonospacelet to look at the next token.

**for HTML output:** 4920 \begin{warpHTML}

```
\LWR@mynexttoken
```

```
4921 \newcommand{\LWR@mynexttoken}{\relax}
```

\futurelet copies the next token then executes a function to analyze

\LWR@futurenonospacelet does the same, but ignores intervening white space

Based on the booktabs style:

```
\LWR@futurenonospacelet
```

```
4922 \def{\LWR@futurenonospacelet#1{\def{\LWR@cs{#1}}%  
4923 \afterassignment{\LWR@fnalone\let\nexttoken= }  
4924 \def{\LWR@fnalone}{\expandafter{\futurelet{\LWR@cs{\LWR@fnltwo}}%  
4925 \def{\LWR@fnltwo}{%  
4926 \expandafter{\ifx{\LWR@cs{\@sp token}\let\next=\LWR@fnlthree}%  
4927 \else\let\next=\nexttoken\fi\next}  
4928 \def{\LWR@fnlthree}{\afterassignment{\LWR@fnalone\let\next= }}
```

\LWR@getmynexttoken Looks ahead and copies the next token into \LWR@mynexttoken.

```
4929 \newcommand*{\LWR@getmynexttoken}{%  
4930 \LWR@traceinfo{\LWR@getmynexttoken}{%  
4931 % nothing must follow this next line  
4932 \LWR@futurenonospacelet{\LWR@mynexttoken\LWR@tabledatacolumnntag}  
4933 }
```

## 59.3 Booleans

Bool LWR@startedrow True if should print a row tag before this column.

```
4934 \newbool{\LWR@startedrow}  
4935 \boolfalse{\LWR@startedrow}
```

Bool LWR@doinghline True if the next row will have an hline or midrule above it. Also used for \midrule.

```
4936 \newbool{\LWR@doinghline}  
4937 \boolfalse{\LWR@doinghline}
```

Bool LWR@doingtbrule True if the next row will have a top/bottom rule above it.

```
4938 \newbool{LWR@doingtbrule}
4939 \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.

```
4940 \newbool{LWR@doingcmidrule}
4941 \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.

```
4942 \newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty \multirow cell, and thus there is no data tag to close.

```
4943 \newbool{LWR@skippingmrowcell}
```

Bool LWR@skippingmcolrowcell True if are doing an empty \multicolumn cell, and thus there is no data tag to close, and do not print @ and ! columns.

```
4944 \newbool{LWR@skippingmcolrowcell}
```

Bool LWR@skipatbang True if just finished a \multicolumn so should not create the trailing @ or ! columns table data cells.

```
4945 \newbool{LWR@skipatbang}
```

Bool LWR@emptyatbang True if finishing a row and should print empty @ or ! column table data cells.

```
4946 \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.

```
4947 \newbool{LWR@intabularmetadata}
4948 \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](http://tex.stackexchange.com/questions/11860/#11860)

```
\LWR@insertatbangcols
```

```
4949 \newcommand*{\LWR@insertatbangcols}{%
4950 \ifbool{\LWR@skipatbang}{%
4951 {}{%
4952 {}{%
4953     \LWR@printatbang{at}{\theLWR@tablecolspos}{%
4954     \LWR@printatbang{bang}{\theLWR@tablecolspos}{%
4955 }{%
4956 }}
```

\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.

```
4957 \newcommand*{\LWR@closetabledatacell}{%
4958 \global\booltrue{\LWR@intabularmetadata}{%
4959 \ifbool{\LWR@existingtabular}{%
4960 {}{ not exiting tabular
4961     \ifboolexpr{bool{\LWR@skippingmrowcell} or bool{\LWR@skippingmcolrowcell}}{%
4962     {}{}}
```

If not skipping a `\multicolumnrow` cell, insert the @ and ! columns after this non-existent column.

```
4963     \ifbool{\LWR@skippingmcolrowcell}{%
4964     {}{%
4965     {\LWR@insertatbangcols}{%
4966 }{%
4967     {}{ not skippingmrowcell}}
```

Insert any < then any @ and ! column contents, unless muted for the `\bottomrule` or a `\multicolumn`:

```
4968     \unskip%
4969     \ifboolexpr{%
4970         bool{\LWR@tabularmutemods} or
4971         bool{\LWR@skipatbang} or
4972         bool{\LWR@emptyatbang}
4973     }{%
4974     {}{%
4975     {\LWR@getexpparray{\LWR@colaferspec}{\theLWR@tablecolspos}}{}}
```

Close paragraphs:

```
4976     \ifbool{\LWR@tableparcell}{\LWR@stoppars}{%
4977     \global\boolfalse{\LWR@tableparcell}{}}
```

Close the table data cell. Skip the @ and ! cells if are closing a multicolumn cell.

```

4978      \leavevmode\unskip\LWR@htmtag{/td}\LWR@orignewline%
4979      \LWR@insertatbangcols%
4980  }% not skipping mrowcell
4981 }% not exiting tabular
4982 \global\boolfalse{LWR@skippingmrowcell}%
4983 \global\boolfalse{LWR@skippingmcolrowcell}%
4984 \global\boolfalse{LWR@skipatbang}%
4985 }
```

LWR@tabulardepth tracks whether & is being used inside a tabular.

```

4986 \newcounter{LWR@tabulardepth}
4987 \setcounter{LWR@tabulardepth}{0}
4988
```

When not used inside a tabular, & performs its original function as recorded here ( with catcode 4 ).

```

4989 \let\LWR@origampmacro&
4990 \end{warpHTML}
```

#### 59.4.1 Localizing & catcodes

**for HTML & PRINT:** 4991 \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.

```

4992 \newcommand{\StartDefiningTabulars}{%
4993 \LWR@traceinfo{StartDefiningTabulars}%
4994 \warpHTMLonly{\catcode`\&=\active}%
4995 }
```

\EndDefiningTabulars Place after defining something with & in it.

```
4996 \newcommand{\EndDefiningTabulars}{%
```

```

4997 \LWR@traceinfo{EndDefiningTabulars}
4998 \warpHTMLonly{\catcode`\&=4}%
4999 }

5000 \end{warpall}

```

#### 59.4.2 Handling &

**for HTML output:** 5001 \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.

```

5002 \newcommand*\LWR@tabularampersand{%
5003 \LWR@traceinfo{\LWR@tabularampersand}%
5004 \ifthenelse{\cnttest{\value{\LWR@tabulardepth}}{>}{0}}{%
5005 {%

```

If not skipping a multirow cell, close the current data cell.

```

5006 \unskip%
5007 \LWR@closetabledatacell%

```

Move to the next column.

```
5008 \addtocounter{\LWR@tablecolspos}{1}%
```

Look at the next token to decide multi or single column data tag.

```

5009 \LWR@getmynexttoken%
5010 }%

```

If not inside a tabular, performs the original action:

```

5011 {\LWR@origampmacro}%
5012 }

```

& 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.

```
5013 \newcommand*{\LWR@tabularfinishrow}{%
```

If not exiting the tabular, or doing a rule, or have already started a row, finish this row:

```
5014 \ifboolexpr{%
5015   not bool {LWR@exittingtabular} or%
5016   bool{LWR@doingtbrule} or%
5017   bool{LWR@doingcmidrule} or%
5018   bool{LWR@doinghline} or%
5019   bool{LWR@startedrow}%
5020 }{%
```

To locally temporarily turn off LWR@exittingtabular so that table data tags will still be generated:

```
5021 \begingroup%
```

If generating a final row for the \bottomrule borders, turn off the @, !, <, and > column output:

```
5022 \ifbool{LWR@exittingtabular}{%
5023   \booltrue{LWR@tabularmutemods}%
5024 }{}}
```

Reenable the table data tags until finished with the final row:

```
5025 \boolfalse{LWR@exittingtabular}%
```

Generate table data tags and ampersands until the right edge:

```
5026 \whiledo{\value{LWR@tablecolspos}<\value{LWR@tabletotalcols}}{%
5027 \LWR@tabledatasinglecolumntag%
```

The following is essentially \LWR@tabularampersand with LWR@emptyatbang added to empty the following cells:

```
5028   \LWR@closetabledatacell%
5029   \addtocounter{LWR@tablecolspos}{1}%
5030   \global\booltrue{LWR@emptyatbang}%
```

Starts the next cell:

```
5031     \LWR@getmynexttoken%
5032 }%
```

Required to close the final cell:

```
5033 \LWR@closetabledatacell%
```

Reenable the original LWR@exittingtabular to close the entire table:

```
5034 \endgroup%
5035 \global\boolfalse{LWR@emptyatbang}%
5036 }{}% ifboolexpr
5037 }
```

## 59.5 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendofline

Throws away options \\[dim] or \\\*

```
\LWR@tabularendofline
```

```
5038 \NewDocumentCommand{\LWR@tabularendofline}{s o}
5039 {%
5040 \ifthenelse{\value{LWR@tablecolspos}<\value{LWR@tabletotalcols}}{%
5041     \LWR@tabularfinishrow%
5042 }%
5043 {%
5044     \LWR@closetabledatacell%
5045 }%
5046 % \begin{macrocode}
5047 \LWR@htmlltag{/tr}\LWR@orignewline
5048 \global\booltrue{LWR@intabularmetadata}
```

Not yet started a table row:

```
5049 \global\boolfalse{LWR@startedrow}
```

Additional setup:

```
5050 \global\boolfalse{LWR@doinglehline}%
5051 \global\boolfalse{LWR@doingtbrule}%
5052 \global\boolfalse{LWR@doingcmidrule}%
5053 \LWR@clearmidrules%
```

Start at first column:

```
5054 \setcounter{LWR@tablecolspos}{1}
```

Look at the next token to decide between single column data tag or a special case:

```
5055 \LWR@getmynexttoken%
5056 }
```

## 59.6 Variables

```
5057 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
5058 \newcommand*{\LWR@pposition}{}%
5059 \newcommand*{\LWR@pleft}{}%
5060 \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 L<sup>A</sup>T<sub>E</sub>X table column, without @, !, >, <, or the vertical bar.

```
5061 \newcommand*{\LWR@tablecolspec}{}%
```

`\LWR@strresult` Holds the result of Str functions.

```
5062 \newcommand*{\LWR@strresult}{}%
5063 \newcommand*{\LWR@strresulttwo}{}%
```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

```
5064 \newcommand*{\LWR@origcolspec}{}%
```

`Ctr LWR@tablecolswidth` Holds the width of the table columns specification.

This is the number of tokens, including one for each @ etc. column, and also one each for the parameters of p, @, etc. columns, and three for each D column.

(This is not the total # of L<sup>A</sup>T<sub>E</sub>X columns in the table.)

```
5065 \newcounter{LWR@tablecolswidth}
```

`Ctr LWR@tablecolspos` Where are currently looking into the table column specification. Index starts at 1.

```
5066 \newcounter{LWR@tablecolspos}
```

Ctr `LWR@tabletotalcols` Holds the final number of table columns, not counting @ and ! columns. This is equal to the number of cells in each row.

```
5067 \newcounter{LWR@tabletotalcols}
```

Ctr `LWR@tabletotalcolsnest` Holds the next column while parsing. Is one more than `LWR@tabletotalcols`.

```
5068 \newcounter{LWR@tabletotalcolsnest}
```

`LWR@colatspec` A data array of specifications for @ columns. The leftmost's index is `leftedge`, the others are counter values. See section 33.

`LWR@colbangspec` A data array of specifications for ! columns. The leftmost's index is `leftedge`, the others are counter values. See section 33.

`LWR@colbeforespec` A data array of specifications for > columns.

`LWR@col afterspec` A data array of specifications for < columns.

`LWR@colbarspec` A data array of specifications for vertical rules.

## 59.7 Parsing @, >, <, !, bar columns

Holds the parsed argument for @, >, <, or ! columns:

```
5069 \newcommand*{\LWR@colparameter}{}%
```

`\LWR@parseatcolumn` Handles @{text} columns.

```
5070 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
5071 \LWR@traceinfo{at column}%
5072 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into `\LWR@strresult`, expanding once:

```
5073 \LWR@traceinfo{about to read the next token:}%
5074 \expandarg%
5075 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]
5076 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
5077 \LWR@traceinfo{have now read the next token}%
```

```

5078 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}
5079 {%
5080   \LWR@traceinfo{at the left edge}%
5081   \LWR@setexpparray{LWR@colatspec}{leftedge}{\LWR@colparameter}%
5082   \LWR@traceinfo{at the left edge: \%}
5083   \LWR@getexpparray{LWR@colatspec}{leftedge}%
5084 }%
5085 {%
5086   \LWR@traceinfo{not at the left edge}%
5087   \LWR@setexpparray{LWR@colatspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
5088   \LWR@traceinfo{at \theLWR@tabletotalcols: \%}
5089   \LWR@getexpparray{LWR@colatspec}{\theLWR@tabletotalcols}%
5090 }%
5091 \let\LWR@colparameter\relax%
5092 \booltrue{LWR@validtablecol}%
5093 }

```

\LWR@parsebangcolumn Handles !{text} columns.

```
5094 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```

5095 \LWR@traceinfo{bang column}%
5096 \addtocounter{LWR@tablecolspos}{1}%

```

Read the next token into \LWR@strresult, expanding once:

```

5097 \LWR@traceinfo{about to read the next token}%
5098 \expandarg%
5099 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]
5100 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@strresult:

```

5101 \LWR@traceinfo{have now read the next token}%
5102 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}
5103 {%
5104   \LWR@traceinfo{at the left edge}%
5105   \LWR@setexpparray{LWR@colbangspec}{leftedge}{\LWR@colparameter}%
5106 }%
5107 {%
5108   \LWR@traceinfo{not at the left edge}%
5109   \LWR@setexpparray{LWR@colbangspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
5110   \LWR@traceinfo{bang \theLWR@tabletotalcols: \LWR@colparameter!}%
5111 }%
5112 \let\LWR@colparameter\relax%
5113 \booltrue{LWR@validtablecol}%

```

5114 }

\LWR@parsebeforecolumn Handles >{text} columns.

5115 \newcommand\*\{\LWR@parsebeforecolumn\}{%

Move to the next token after the '>':

5116 \addtocounter{LWR@tablecolspos}{1}%

Read the next token, expanding once into \LWR@strresult:

5117 \expandarg%

5118 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]%

5119 \fullexpandarg%

Store the result into a data array, expanding once out of \LWR@strresult:

5120 \LWR@setexpparray{\LWR@colbeforespec}{\theLWR@tabletotalcolsnext}[\LWR@colparameter]%

5121 \let\LWR@colparameter\relax%

5122 \booltrue{\LWR@validtablecol}%

5123 }

\LWR@parseaftercolumn Handles <{text} columns.

5124 \newcommand\*\{\LWR@parseaftercolumn\}{%

Move to the next token after the '<':

5125 \addtocounter{LWR@tablecolspos}{1}%

Read the next token, expanding once into \LWR@strresult:

5126 % \StrChar{\#1}{\theLWR@tablecolspos}[\LWR@colparameter]

5127 \expandarg%

5128 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]%

5129 \fullexpandarg%

Store the result into a data array, expanding once out of \LWR@strresult:

5130 \LWR@setexpparray{\LWR@colafterspec}{\theLWR@tabletotalcols}[\LWR@colparameter]%

5131 \let\LWR@colparameter\relax%

5132 \booltrue{\LWR@validtablecol}%

5133 }

\LWR@parsebarcolumn Handles vertical rules.

```
5134 \newcommand*{\LWR@parsebarcolumn}{%
5135 \LWR@traceinfo{bar column}%
```

Remember the bar at this position:

```
5136 \ifthenelse{\cnttest{\value{\LWR@tabletotalcols}}=0}{%
5137 {%
5138     \LWR@setexpparray{\LWR@colbarspec}{leftedge}{tvertbar1}%
5139 }%
5140 {%
5141     \LWR@setexpparray{\LWR@colbarspec}{\the\LWR@tabletotalcols}{tvertbarr}%
5142 }%
5143 \booltrue{\LWR@validtablecol}%
5144 }
```

## 59.8 Parsing 'l', 'c', or 'r' columns

\LWR@parsenormalcolumn {\langle thiscolumn\rangle}

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```
5145 \newcommand*{\LWR@parsenormalcolumn}[1]{%
5146 \appto{\LWR@tablecolspec}{#1}%
5147 \addtocounter{\LWR@tabletotalcols}{1}%
5148 \addtocounter{\LWR@tabletotalcolsnext}{1}%
5149 \LWR@traceinfo{normal column \the\LWR@tabletotalcols: #1}%
5150 \LWR@setexpparray{\LWR@colatspec}{\the\LWR@tabletotalcolsnext}{}%
5151 \LWR@setexpparray{\LWR@colbangspec}{\the\LWR@tabletotalcolsnext}{}%
5152 \LWR@setexpparray{\LWR@colbeforespec}{\the\LWR@tabletotalcolsnext}{}%
5153 \LWR@setexpparray{\LWR@col afterspec}{\the\LWR@tabletotalcolsnext}{}%
5154 \LWR@setexpparray{\LWR@colbarspec}{\the\LWR@tabletotalcolsnext}{}%
5155 \booltrue{\LWR@validtablecol}%
5156 }
```

## 59.9 Parsing 'p', 'm', or 'b' columns

\LWR@parsepcolumn {\langle thiscolumn\rangle} The width will be ignored.

```
5157 \newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type:

```
5158 \LWR@parsenormalcolumn{#1}%
```

Skips the following width token:

```
5159 \addtocounter{LWR@tablecolspos}{1}%
5160 }
```

## 59.10 Parsing ‘D’ columns

From the `dcolumn` package.

`\LWR@parseDcolumn {<thiscolumn>}` The three parameters will be ignored.

```
5161 \newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
5162 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
5163 \addtocounter{LWR@tablecolspos}{3}%
5164 }
```

## 59.11 Parsing the column specifications



HTML CSS cannot exactly match the  $\text{\TeX}$  concept of a baseline for a table row. Table 7 shows the  $\text{\TeX}$  results for various vertical-alignment choices, with the baseline of the first column drawn across all the columns for comparison. See the `p` column specification in table 8 for details.

Table 8 describes how each kind of column is converted to HTML.

`Bool LWR@validtablecol` True if found a valid table column type.

```
5165 \newbool{\LWR@validtablecol}
```

`\LWR@parsetablecols {<colspecs>}`

Scans the column specification left to right.

Table 7: Tabular baseline

l	p	m	b	r
			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

Table 8: Tabular HTML column conversions

**l, r, c:** Converted to table cells without paragraph tags.

Uses css `vertical-align:middle` so that top or bottom-aligned cells may go above or below this cell.

**p:** Converted to table cells with paragraph tags. Ref: Table 7, L<sup>A</sup>T<sub>E</sub>X places the top line of a parbox aligned with the rest of the text line, so css `vertical-align:bottom` is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the css may be desired, changing `td.tdp`, `td.tdP`, `td.tdprule`, and `td.tdPrule` to `vertical-align: middle`. Another possibility is to change L,R,C, and P to `vertical-align: top` and not worry about the alignment of B and M cells or trying to approximate L<sup>A</sup>T<sub>E</sub>X baselines.

**m:** With paragraph tags, css `vertical-align:middle`.

**b:** With paragraph tags, css `vertical-align:top` so that the bottom of the text is closest to the middle of the text line.

**P, M, B:** Horizontally-centered versions.

**S:** Converted to 'r'. From the `siunitx` package.

**D:** Converted to 'c'. From the `dcolumn` package.

**@, !, >, <:** One each, in that order.

**|:** Vertical rule.

**Unknown:** Converted to 'l'.

**\newcolumn:** Currently treated as unknown.

Builds \LWR@tablecols with the final specification, one column per entry. The final number of cells in each row is stored in \LWR@tabletotalcols.

```
5166 \newcommand*{\LWR@parsetablecols}[1]{%
5167 \LWR@traceinfo{\LWR@parsetablecols started}%
```

Remember the original supplied column spec:

```
5168 \renewcommand*{\LWR@origcolspec}{#1}%
```

Clear the parsed resulting column spec:

```
5169 \renewcommand*{\LWR@tablecols}{%}
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
5170 \setcounter{\LWR@tabletotalcols}{0}%
5171 \setcounter{\LWR@tabletotalcolsnext}{1}%
5172 \LWR@setexpparray{\LWR@colatsspec}{leftedge}{}%
5173 \LWR@setexpparray{\LWR@colatsspec}{1}{}%
5174 \LWR@setexpparray{\LWR@colatsspec}{2}{}%
5175 \LWR@setexpparray{\LWR@colatsspec}{3}{}%
5176 \LWR@setexpparray{\LWR@colbangspec}{leftedge}{}%
5177 \LWR@setexpparray{\LWR@colbangspec}{1}{}%
5178 \LWR@setexpparray{\LWR@colbangspec}{2}{}%
5179 \LWR@setexpparray{\LWR@colbangspec}{3}{}%
5180 \LWR@setexpparray{\LWR@colbeforespec}{1}{}%
5181 \LWR@setexpparray{\LWR@colbeforespec}{2}{}%
5182 \LWR@setexpparray{\LWR@colbeforespec}{3}{}%
5183 \LWR@setexpparray{\LWR@colafterspec}{1}{}%
5184 \LWR@setexpparray{\LWR@colafterspec}{2}{}%
5185 \LWR@setexpparray{\LWR@colafterspec}{3}{}%
5186 \LWR@setexpparray{\LWR@colbarspec}{leftedge}{}%
5187 \LWR@setexpparray{\LWR@colbarspec}{1}{}%
5188 \LWR@setexpparray{\LWR@colbarspec}{2}{}%
5189 \LWR@setexpparray{\LWR@colbarspec}{3}{}%
```

Starting at the first column specification:

```
5190 \setcounter{\LWR@tablecolspos}{1}%
```

Place the colspecs string length into \LWR@strresult, and remember the number of characters in the column specification:

```
5191 \LWR@traceinfo{about to StrLen}%
5192 \noexpandarg%
5193 \StrLen{\#1}[\LWR@strresult]%
```

```

5194 \fullexpandarg%
5195 \LWR@traceinfo{finished StrLen}%
5196 \setcounter{\LWR@tablecolswidth}{\LWR@strresult}%

```

Scan through the column specifications:

```
5197 \whiledo{\not\value{\LWR@tablecolspos}>\value{\LWR@tablecolswidth}}{%
```

Place the next single-character column type into \LWR@strresult:

```

5198 \noexpandarg%
5199 \StrChar{\#1}{\the\LWR@tablecolspos}[\LWR@strresult]%
5200 \LWR@traceinfo{position \arabic{\LWR@tablecolspos}: \LWR@strresult}%
5201 \fullexpandarg%

```

Not yet found a valid column type

```
5202 \boolfalse{\LWR@validtablecol}%
```

Note that the parameter for a p{spec} column is a token list which will NOT match l,c,r,p.

```

5203 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{%
5204 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{%
5205 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{%
5206 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{%
5207 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{%
5208 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{%
5209 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{%
5210 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{%
5211 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{%
5212 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{%
5213 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{%
5214 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{%
5215 \IfStrEq{\LWR@strresult}{|}{\LWR@parsebarcolumn}{%
5216 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{%
5217 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{%
5218 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{%

```

From the dcolumn package:

```
5219 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{%
```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```
5220 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{%
```

---

Many people define centered versions “P”, “M”, and “B”:

---

```
\newcolumntype{P}[1]{>{\centering\arraybackslash}p{#1}}
```

---

```
5221 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
5222 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
5223 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%
```

If this column was an invalid column type, convert it to an l column:

```
5224 \ifbool{\LWR@validtablecol}{%
5225   \LWR@traceinfo{invalid column type: \LWR@strresult}%
5226   \LWR@parsonormalcolumn{l}%
5227 }%
5228 \addtocounter{\LWR@tablecolspos}{1}%
5229 }%
5230 }%
```

## 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.

```
5231 \newcommand*{\LWR@maybenewtablerow}{%
5232 {%
5233 \ifbool{\LWR@startedrow}{%
5234 }% started the row
5235 }% not started the row
}
```

Remember that now have started the row:

```
5236 \global\booltrue{\LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```
5237 \global\booltrue{\LWR@intabularmetadata}%
5238 \ifbool{\LWR@doinghline}{%
5239 {%
5240   \LWR@htmlltag{tr class="hline"}%
5241   \LWR@orignewline%
5242 }%
5243 {%
5244   \ifbool{\LWR@doingtbrule}{%
5245     \LWR@htmlltag{tr class="tbrule"}%
5246   }%
5247 }
```

```

5247           \LWR@orignewline%
5248       }%
5249       {\LWR@htmltag{tr}\LWR@orignewline}%
5250   }% end of not doing hline
5251 }% end of not started the row
5252 }

```

### 59.13 Printing vertical bar tags

\LWR@printbartag {*index*}

Adds to a tabular data cell an HTML class name for a left/right vertical bar.

```

5253 \newcommand*{\LWR@printbartag}[1]{%
5254 \ifboolexpr{bool{\LWR@tabularmutemods} or bool{\LWR@emptyatbang}}{%
5255 {}% muting or empty
5256 {}% not muting
5257     \edef\LWR@tempone{\LWR@getexpparray{\LWR@colbarspec}{#1}}%
5258     \ifdefempty{\LWR@tempone}{}{\LWR@tempone}%
5259 }% not muting
5260 }

```

### 59.14 Printing at or bang tags

\LWR@printatbang {*at-or-bang*} {*index*}

```
5261 \newcommand*{\LWR@printatbang}[2]{%
```

Fetch the column at or bang spec:

```
5262 \edef\LWR@atbangspec{\LWR@getexpparray{\LWR@col#1spec}{#2}}%
5263 \LWR@traceinfo{atbang: #2 !\LWR@atbangspec!}%
```

Only generate if is not empty;

```

5264 \ifdefempty{\LWR@atbangspec}{%
5265 {}%
5266 {}% not empty
5267     \LWR@htmltag{%
5268         td class="td#1"%
5269         \LWR@subaddcmidruletrim{}{}%
5270         \LWR@printbartag{#2}%
5271     }%

```

```

5272     \LWR@tdstartstyles%
5273     \LWR@addcmidrulewidth%
5274     \LWR@tdendstyles%
5275 }%

```

Create an empty cell if muting for the \bottomrule:

```

5276     \ifboolexpr{bool{\LWR@tabularmutemods} or bool{\LWR@emptyatbang}}{%
5277     {}%
5278     {\LWR@atbangspec}%
5279 }%
5280     \LWR@htmltag{/td}\LWR@orignewline%
5281 }% not empty
5282 }%

```

\LWR@addleftmostbartag

```

5283 \newcommand*{\LWR@addleftmostbartag}{%
5284 \ifnumcomp{\value{\LWR@tablecolspos}}{=}{1}{%
5285     \LWR@printbartag{leftedge}%
5286 }{}%
5287 }

```

\LWR@tabularleftedge

```

5288 \newcommand*{\LWR@tabularleftedge}{%
5289 \ifthenelse{\cnttest{\value{\LWR@tablecolspos}}=1}{%
5290 {}%
5291     \LWR@printatbang{at}{leftedge}%
5292     \LWR@printatbang{bang}{leftedge}%
5293 }% left edge
5294 {}% not left edge
5295 }

```

## 59.15 Data opening tag

\LWR@tabledatasinglecolumntag Print a table data opening tag with style for alignment

```

5296 \newcommand*{\LWR@tabledatasinglecolumntag}{%
5297 {}%
5298 \LWR@maybenewtablerow%

```

Don't start a new paragraph tag if have already started one:

---

```
5299 \ifbool{LWR@intabularmetadata}{%
5300 {%
```

If have found the end of tabular command, do not create the next data cell:

```
5301     \ifbool{LWR@exittingtabular}{}{%
5302     {%
5303         \LWR@tabularleftedge{%
```

Print the @ and ! contents before first column:

```
5303             \LWR@tabularleftedge{%
```

Fetch the current column's alignment character into \LWR@strresult:

```
5304             \StrChar{\LWR@tablecolspe}{\theLWR@tablecolspos}{[\LWR@strresult]}%
```

print the start of a new table data cell:

```
5305             \LWR@htmntag{td class="td{%
```

append this column's spec:

```
5306             \LWR@strresult{%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag. Also add vertical bar tags.

```
5307             \LWR@addcmidruletrim{%
5308             \LWR@addleftmostbartag{%
5309             \LWR@printbartag{\theLWR@tablecolspos}{%
5310             "%}

5311             \LWR@tdstartstyles{%
5312             \LWR@addcmidrulewidth{%
5313             \LWR@addformatwpalignment{%
5314             \LWR@tdendstyles{%
5315             }{%
```

If this is a p, m, b, or X column, allow paragraphs:

```
5316             \ifthenelse{%
5317                 \equal{\LWR@strresult}{p}\OR%
5318                 \equal{\LWR@strresult}{m}\OR%
5319                 \equal{\LWR@strresult}{b}\OR%
5320                 \equal{\LWR@strresult}{P}\OR%
5321                 \equal{\LWR@strresult}{M}\OR%
5322                 \equal{\LWR@strresult}{B}\OR%
```

---

```

5323           \equal{\LWR@strresult}{X}%
5324       }%
5325   {%
5326     allow pars
5327     \LWR@startpars%
5328     \global\booltrue{\LWR@tableparcell}%
5329   }%
5329   {}% no pars

```

Print the > contents unless muted for the \bottomrule:

```

5330   \ifboolexpr{bool{\LWR@tabularmutemods} or bool{\LWR@emptyatbang}}{%
5331   }%
5332   {\LWR@getexpparray{\LWR@colbeforespec}{\the\LWR@tablecolspos}}%
5333   \global\boolfalse{\LWR@intabularmetadata}%
5334 }% not exiting tabular
5335 }{}% in tabular metadata
5336 }%

```

## 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.

```
5337 \newcounter{LWR@midrulecounter}
```

**Len \LWR@heavyrulewidth** The default width of the rule.

```

5338 \newlength{\LWR@heavyrulewidth}
5339 \setlength{\LWR@heavyrulewidth}{.08em}
```

**Len \LWR@lightrulewidth** The default width of the rule.

```

5340 \newlength{\LWR@lightrulewidth}
5341 \setlength{\LWR@lightrulewidth}{.05em}
```

**Len \LWR@cmidrulewidth** The default width of the rule.

```

5342 \newlength{\LWR@cmidrulewidth}
5343 \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.

```
5344 \newlength{\LWR@thiscmidrulewidth}
5345 \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.

```
5346 \newcommand*{\LWR@clearmidrules}%
5347 {%
5348 \setcounter{LWR@midrulecounter}{1}%
5349 \whiledo{%
5350 \cnttest{\value{LWR@midrulecounter}}{<=}{\value{LWR@tablecolswidth}}}%
5351 }%
5352 {%
5353 \LWR@setexpparray{LWR@midrules}{\theLWR@midrulecounter}{0pt}%
5354 \setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
5355 \LWR@setexpparray{LWR@trimlrules}{\theLWR@midrulecounter}{}%
5356 \LWR@setexpparray{LWR@trimrrules}{\theLWR@midrulecounter}{}%
5357 \addtocounter{LWR@midrulecounter}{1}%
5358 }%
5359 }
```

**\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.

```
5360 \newcommand*{\LWR@subcmidrule}[4]{%
5361 \setcounter{LWR@midrulecounter}{#3}%
5362 \whiledo{\cnttest{\value{LWR@midrulecounter}}{<=}{#4}}{%
5363 {%
5364 \LWR@setexpparray{LWR@midrules}{\theLWR@midrulecounter}{#1}%
5365 \addtocounter{LWR@midrulecounter}{1}%
5366 }% whiledo
5367 \IfSubStr{#2}{1}{\LWR@setexpparray{LWR@trimlrules}{#3}{1}}{}%
5368 \IfSubStr{#2}{r}{\LWR@setexpparray{LWR@trimrrules}{#4}{r}}{}%
5369 \booltrue{LWR@doingcmidrule}%
5370 }
```

---

\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.

```
5371 \NewDocumentCommand{\LWR@docmidrule}{O{\LWR@cmidrulewidth} D(){} >{\SplitArgument{1}{-}m}%
5372 {\LWR@subcmidrule{#1}{#2}#3}}
```

Used to compute margins, tabular trims:

```
5373 \newlength{\LWR@templengthone}%
5374 \newlength{\LWR@templengthtwo}%
```

Used to add a style to a table data cell:

```
5375 \newboolean{\LWR@tdhavecellstyle}
```

\LWR@tdstartstyle Begins possibly adding a table data cell style.

```
5376 \newcommand*{\LWR@tdstartstyle}{\global\boolfalse{\LWR@tdhavecellstyle}}
```

\LWR@tdaddstyle Starts adding a table data cell style.

```
5377 \newcommand*{\LWR@tdaddstyle}{%
5378 \ifbool{\LWR@tdhavecellstyle}{%
5379 {; }%
5380 { style=""}%
5381 \booltrue{\LWR@tdhavecellstyle}%
5382 }}
```

\LWR@tdendstyle Finishes possibly adding a table data cell style. Prints the closing quote.

```
5383 \newcommand*{\LWR@tdendstyle}{%
5384 \ifbool{\LWR@tdhavecellstyle}{%
5385 "%%
5386 \global\boolfalse{\LWR@tdhavecellstyle}%
5387 }{}%
5388 }
```

\LWR@subaddcmidruletrim {⟨lefttrim⟩} {⟨righttrim⟩} Adds a \cmidrule with optional trim.

```
5389 \newcommand*{\LWR@subaddcmidruletrim}[2]{%
5390 \setlength{\LWR@templengthone}{\LWR@getexpparray{\LWR@midrules}{\the\LWR@tablecolspos}}%
5391 \ifdimcomp{\LWR@templengthone}{>}{0pt}%
5392 {%
```

Print the class without left and right trim letters appended:

```
5393     \LWR@origtilde tdrule#1#2%
```

Remember the width of the rule:

```
5394     \setlength{\LWR@thiscmidrulewidth}{\LWR@templengthone}%
5395 }%
5396 {%
5397     \setlength{\LWR@thiscmidrulewidth}{0pt}%
5398 }%
5399 }
```

\LWR@addcmidruletrim Adds left or right trim to a \cmidrule.

```
5400 \newcommand*{\LWR@addcmidruletrim}{%
5401 \LWR@subaddcmidruletrim%
5402 {\LWR@getexparray{\LWR@trimlrules}{\theLWR@tablecolspos}}%
5403 {\LWR@getexparray{\LWR@trimrrules}{\theLWR@tablecolspos}}%
5404 }
```

\LWR@addrulewidth {\<thiswidth>} {\<defaultwidth>}

If not default width, add a custom style with width and color depending on thiswidth.

Must be placed between \LWR@tdstartstyles and \LWR@tdendstyles.

```
5405 \newcommand{\LWR@addrulewidth}[2]{%
```

Only add a custom width if thiswidth is different than the defaultwidth:

```
5406 \ifboolexpr{%
5407     test{\ifdimcomp[#1]{=}{0pt}}%
5408     or ( test{\ifdimcomp[#1]{=}{#2}} and not bool{FormatWP} )%
5409 }%
5410 {}% default width
5411 {}% custom width
```

Ensure that the width is wide enough to display in the browser:

```
5412     \LWR@forceminwidth{#1}%
```

Begin adding a style:

```
5413     \LWR@tdaddstyle%
```

The style itself:

```
5414     \uselengthunit{PT}%
5415     border-top:\rndprintlength{\LWR@atleastonept} solid %
```

The darkness of the color depends on the thickness of the rule:

```
5416     \ifdimcomp{#1}{<}{\LWR@lightrulewidth}%
5417     {\#AOAOAO}%
5418     {%
5419         \ifdimcomp{#1}{<}{\LWR@heavyrulewidth}%
5420         {\#808080}%
5421         {black}%
5422     }%
5423 }% custom width
5424 }
```

\LWR@addcmidrulewidth Adds a style for the rule width.

Must be placed between \LWR@tdstartstyles and \LWR@tdendstyles.

```
5425 \newcommand{\LWR@addcmidrulewidth}{%
5426 \LWR@addrulewidth{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
5427 }
```

\LWR@WPcell {*text-align*} {*vertical-align*}

```
5428 \newcommand*{\LWR@WPcell}[2]{%
5429 \LWR@tdaddstyle%
5430 text-align:#1; vertical-align:#2%
5431 }
```

\LWR@addformatwpalignment If FormatWP, adds a style for the alignment.

Must be placed between \LWR@tdstartstyles and \LWR@tdendstyles.

```
5432 \newcommand*{\LWR@addformatwpalignment}{%
5433 \ifbool{FormatWP}{%
5434 \StrChar{\LWR@tablecolspe}{\theLWR@tablecolspos}[\LWR@strresult]%
5435 \ifdefstring{\LWR@strresult}{l}{\LWR@WPcell{left}{middle}}{}%
5436 \ifdefstring{\LWR@strresult}{c}{\LWR@WPcell{center}{middle}}{}%
5437 \ifdefstring{\LWR@strresult}{r}{\LWR@WPcell{right}{middle}}{}%
5438 \ifdefstring{\LWR@strresult}{p}{\LWR@WPcell{left}{bottom}}{}%
5439 \ifdefstring{\LWR@strresult}{m}{\LWR@WPcell{left}{middle}}{}%
5440 \ifdefstring{\LWR@strresult}{b}{\LWR@WPcell{left}{top}}{}%
5441 \ifdefstring{\LWR@strresult}{P}{\LWR@WPcell{center}{bottom}}{}%
5442 \ifdefstring{\LWR@strresult}{M}{\LWR@WPcell{center}{middle}}{}%
```

```

5443 \ifdefstring{\LWR@strresult}{B}{\LWR@WPcell{center}{top}}{}%
5444 }{}%
5445 }

```

## 59.17 Multicolumns

### 59.17.1 Parsing multicolumns

```
5446 \newcounter{LWR@tablemulticolswidth}
```

Indexes into the multicolumn specification:

```
5447 \newcounter{LWR@tablemulticolspos}
```

\LWR@printmccoltype {<colspec>} Print any valid column type found. Does not print @, !, >, or < columns or their associated tokens.

This is printed as part of the table data tag's class.

```

5448 \newcommand*\LWR@printmccoltype[1]{%
5449 \LWR@traceinfo{lwr@printmccoltype -#1-}%

```

Get one token of the column spec:

```
5450 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```

5451 \IfStrEq{\LWR@strresult}{l}{%
5452 \IfStrEq{\LWR@strresult}{c}{%
5453 \IfStrEq{\LWR@strresult}{r}{%
5454 \IfStrEq{\LWR@strresult}{p}{%
5455 \IfStrEq{\LWR@strresult}{m}{%
5456 \IfStrEq{\LWR@strresult}{b}{%
5457 \IfStrEq{\LWR@strresult}{P}{%
5458 \IfStrEq{\LWR@strresult}{M}{%
5459 \IfStrEq{\LWR@strresult}{B}{%
5460 \IfStrEq{\LWR@strresult}{S}{%
5461 \IfStrEq{\LWR@strresult}{X}{%
5462 \LWR@traceinfo{lwr@printmccoltype done}%
5463 }

```

\LWR@multicolpartext Print the data with paragraph tags:

```
5464 \newcommand*\LWR@multicolpartext{%
```

---

```

5465 \LWR@startpars%
5466 \LWR@multicoltext%
5467 \LWR@stoppars%
5468 }

```

\LWR@multicolother {*colspec*} For @, !, >, <, print the next token without paragraph tags:

```

5469 \newcommand*\{LWR@multicolother}[1]{%
5470 \addtocounter{LWR@tablemulticolspos}{1}%
5471 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
5472 \LWR@strresult%

```

A valid column data type was found:

```

5473 \booltrue{LWR@validtablecol}%
5474 }

```

\LWR@multicolskip Nothing to print for this column type.

```

5475 \newcommand*\{LWR@multicolskip}{}%

```

A valid column data type was found:

```

5476 \booltrue{LWR@validtablecol}%
5477 }

```

\LWR@printmccoldata {*colspec*} Print the data for any valid column type found.

```

5478 \newcommand*\{LWR@printmccoldata}[1]{%
5479 \LWR@traceinfo{lwr@printmccoldata -#1}%

```

Not yet found a valid column type:

```

5480 \boolfalse{LWR@validtablecol}%

```

Get one token of the column spec:

```

5481 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%

```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```

5482 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
5483 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
5484 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
5485 \IfStrEq{\LWR@strresult}{D}{%

```

```

5486 \addtocounter{LWR@tablemulticolspos}{3}%
5487 \LWR@multicoltext%
5488 }{%
5489 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolpartext}{}%
5490 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolpartext}{}%
5491 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolpartext}{}%
5492 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolpartext}{}%
5493 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolpartext}{}%
5494 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolpartext}{}%
5495 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolpartext}{}%
5496 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolpartext}{}%
5497 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
5498 \IfStrEq{\LWR@strresult}{@}{\LWR@multicolother{#1}}{}%
5499 \IfStrEq{\LWR@strresult}{!}{\LWR@multicolother{#1}}{}%
5500 \IfStrEq{\LWR@strresult}{>}{\LWR@multicolother{#1}}{}%
5501 \IfStrEq{\LWR@strresult}{<}{\LWR@multicolother{#1}}{}%

```

If an invalid column type:

```
5502 \ifbool{LWR@validtablecol}{}{\LWR@multicoltext}%
```

Tracing:

```

5503 \LWR@traceinfo{lwr@printmccoldata done}%
5504 }
```

\parsemulticolumnalignment {⟨1: colspec⟩} {⟨2: printresults⟩}

Scan the multicolumn specification and execute the printfunction for each entry.

Note that the spec for a p{spec} column, or @, >, <, is a token list which will NOT match l, c, r, or p.

```

5505 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
5506 \setcounter{LWR@tablemulticolspos}{1}%
5507 \StrLen{\#1}[\LWR@strresult]%
5508 \setcounter{LWR@tablemulticolswidth}{\LWR@strresult}%

```

Scan across the tokens in the column spec:

```

5509 \whiledo{%
5510 \not\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolswidth}%
5511 }%
5512 {%
```

Execute the assigned print function for each token in the column spec:

```
5513 #2{\#1}%
```

Move to the next token in the column spec:

```
5514 \addtocounter{LWR@tablemulticolspos}{1}%
5515 }%
5516 }
```

### 59.17.2 Multicolumn factored code

```
5517 \newcommand{\LWR@multicoltext}{}
\LWR@domulticolumn [⟨1: vpos⟩] [⟨2: #rows⟩] {⟨3: numcols⟩} {⟨4: colspec⟩} {⟨5: text⟩}
5518 \NewDocumentCommand{\LWR@domulticolumn}{o o m m +m}{%
5519 \LWR@traceinfo{lwr@domulticolumn -#1- -#2- -#3- -#4-}%
}
```

Remember the text to be inserted, and remember that a valid column type was found:

```
5520 \renewcommand{\LWR@multicoltext}{%
5521 #5%
5522 \booltrue{\LWR@validtablecol}%
5523 }%
```

Row processing:

```
5524 \LWR@maybenewtablerow%
5525 \LWR@tabularleftedge%
```

Begin the opening table data tag:

```
5526 \LWR@htmlltag{td colspan="#3" %
5527 \IfValueT{#2}{ % rows?
5528 rowspan="#2" %

5529 \IfValueT{#1}{% vpos?
5530 \ifstrequal{#1}{b}{style="vertical-align:bottom" }{}%
5531 \ifstrequal{#1}{t}{style="vertical-align:top" }{}%
5532 }% vpos?
5533 }% rows?

5534 class="td%"
```

Print the column type:

```
5535 \LWR@parsemulticolumnalignment{#4}{\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”. Also add vertical bar class.

```
5536 \LWR@addcmidruletrim%
5537 \LWR@addleftmostbartag%
5538 \LWR@printbartag{\theLWR@tablecolspos}%
```

Close the class tag's opening quote:

```
5539 "%
```

```
5540 \LWR@tdstartstyles%
5541 \LWR@addcmidrulewidth%
5542 \LWR@addformatwpalignment%
5543 \LWR@tdendstyles%
5544 }% end of the opening table data tag
5545 \global\boolfalse{LWR@intabularmetadata}%
5546 \LWR@parsemulticolumnalignment{#4}{\LWR@printmccoldata}%
5547 }
```

### 59.17.3 Multicolumn

```
\LWR@htmlemulticolumn {<numcols>} {<alignment>} {<text>}

5548 \NewDocumentCommand{\LWR@htmlemulticolumn}{m m +m}%
5549 {%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
5550 \LWR@tabularhtmlcolumns{\theLWR@tablecolspos}{#1}
```

Create the multicolumn tag:

```
5551 \LWR@domulticolumn{\theLWR@tabhtmlcoltotal}{#2}{#3}%
```

Move to the next L<sup>A</sup>T<sub>E</sub>X column:

```
5552 \addtocounter{LWR@tablecolspos}{#1}%
5553 \addtocounter{LWR@tablecolspos}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
5554 \booltrue{LWR@skipatbang}%
5555 }
```

### 59.17.4 Longtable captions

longtable captions use `\multicolumn`.

Bool `LWR@starredlongtable` Per the caption pacakge, step the counter if `longtable*`.

```
5556 \newbool{LWR@starredlongtable}
5557 \boolfalse{LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
5558 \providecommand*\LTcaptive{table}
```

```
\LWR@longtabledatacaptiontag * [<toc entry>] {<caption>}
```

```
5559 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
5560 {%
```

Remember the latest name for `\nameref`:

```
5561 \IfValueTF{#2}{%
  optional given?
  \ifthenelse{\equal{#2}{}}{%
    optional empty?
    {\LWR@setlatestname{#3}}% empty
    {\LWR@setlatestname{#2}}% given and non-empty
  }%
  optional given
  {\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:

```
5567 \LWR@tabularhtmlcolumns{1}{\theLWR@tabletotalcols}
```

Create the multicolumn tag:

```
5568 \LWR@domulticolumn{\theLWR@tabhtmlcoltotal}{P}{%
  \LWR@domulticolumn
  \IfBooleanTF{#1}{%
    star?
  }}
```

Star version, show a caption but do not make a LOT entry:

```
5570 {%
  yes star
  \LWR@figcaption%
  #3%
  \endLWR@figcaption%
}%
5575 {%
  No star:
```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```

5576      \ifbooleq{\LWR@starredlongtable}{%
5577        {%
5578          \ifthenelse{\equal{\#2}{}}{%
5579            {}%
5580            {}%
5581              \refstepcounter{\LTcattypename}%
5582              \protected@edef{\currentlabel}{%
5583                \csuse{\p@{\LTcattypename}}\csuse{\the\LTcattypename}}%
5584            }%
5585          }%
5586        }%
5587      }%
5588    }%

```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```

5586      \LWR@figcaption%
5587      \csuse{\fnum@\LTcattypename}\CaptionSeparator#3%
5588      \endLWR@figcaption%

```

See if an optional caption was given:

```

5589      \ifthenelse{\equal{\#2}{}}{%
5590        {}%
5591      }%
5592      }%

```

if the optional caption was given, but empty, do not form a TOC entry

If the optional caption was given, but might only be []:

```

5591      {%
5592        \IfNoValueTF{\#2}{%
5593          }%
5594          }%

```

The optional caption is []:

```

5593      {%
5594        \addcontentsline{%
5595          \csuse{\ext@\LTcattypename}}{%
5596            \LTcattypename}%
5597            {}%
5598            \protect\newline{%
5599              \csuse{\p@{\LTcattypename}}\csuse{\the\LTcattypename}}%
5600              \ignorespaces #3\protect\relax}%
5601            }%
5602      }%

```

The optional caption has text enclosed:

```

5603      {%
5604          yes TOC entry
5605          \addcontentsline{%
5606              \csuse{ext@\LTcaptype}}{%
5607                  \LTcaptype}%
5608          {%
5609              \protect\newline{%
5610                  \csuse{p@\LTcaptype}\csuse{the\LTcaptype}}{%
5611                      \ignorespaces #2\protect\relax}%
5612          }%
5613      }%
5614 }%

```

Skip any trailing @ or ! columns for this cell:

```

5615 \booltrue{LWR@skipatbang}%
5616 }%
5617 end of \LWR@domulticolumn
5618 \addtocounter{LWR@tablecolspos}{\theLWR@tabletotalcols}
5619 \addtocounter{LWR@tablecolspos}{-1}
5620
5621 }

```

### 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.

```

5622 \newcounter{LWR@tabhtmlcolindex}%
5623 \newcounter{LWR@tabhtmlcolend}%
5624 \newcounter{LWR@tabhtmlcoltotal}%

```

$\text{LWR@tabularhtmlcolumns} \{ \langle \text{starting } \text{\LaTeX} \text{ column} \rangle \} \{ \langle \text{number } \text{\LaTeX} \text{ columns} \rangle \}$

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.

---

```
5625 \newcommand*{\LWR@tabularhtmlcolumns}[2]{%
```

Count the starting index, compute ending index, and begin with the count being the  $\text{\LaTeX}$  span, to which additional @ and ! columns may be added:

```
5626 \setcounter{LWR@tabhtmlcolindex}{#1}%
5627 \setcounter{LWR@tabhtmlcoltotal}{#2}%
5628 \setcounter{LWR@tabhtmlcolend}{#1}%
5629 \addtocounter{LWR@tabhtmlcolend}{#2}%
```

Walk across the  $\text{\LaTeX}$  columns looking for @ and ! columns:

```
5630 \whiledo{\value{LWR@tabhtmlcolindex}<\value{LWR@tabhtmlcolend}}{%
```

Temporarily define a macro equal to the @ specification for this column:

```
5631 \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}%
5632 {\theLWR@tabhtmlcolindex}}%
```

If the @ specification is not empty, add to the count:

```
5633 \ifdefempty{\LWR@atbangspec}{%
5634 {}%
5635 {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the ! columns:

```
5636 \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}%
5637 {\theLWR@tabhtmlcolindex}}%
5638 \ifdefempty{\LWR@atbangspec}{%
5639 {}%
5640 {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Move to the next  $\text{\LaTeX}$  column:

```
5641 \addtocounter{LWR@tabhtmlcolindex}{1}%
5642 }% whiledo
5643 }

5644 \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 178 for the HTML versions.

**for HTML & PRINT:** 5645 \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/problem-with-abbreviation-of-multirow-and-multicolumn-latex>

After the user may have

```
5646 \AtBeginDocument{
```

\@ifundefined{@xmultirow} determines if multirow was never loaded.

```
5647 \@ifundefined{@xmultirow}
5648 {}% no version of multirow was loaded
5649 {}% \@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.

```
5650 \@ifpackageloaded{multirow}{% v2.0 or newer
5651 \@ifpackagelater{multirow}{2016/09/01}{2016/09/27 for v2.0
5652 {}% v2.0+:
5653 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}{%
5654     {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
5655 {\multicolumn{\#1}{\#2}{\@xmultirow{\#3}{\#4}{\#5}{\#6}{\#7}{\#8}}}}%
5656 }
5657 {}% loaded but older, probably not executed:
5658 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}{%
5659     {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
5660 {\multicolumn{\#1}{\#2}{\@xmultirow{\#4}{\#5}{\#6}{\#7}{\#8}}}}%
5661 }
5662 }% 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 vpos option.

```
5663 {}% v1.6 or older did not \ProvidePackage
5664 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}{%
5665     {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
5666 {\multicolumn{\#1}{\#2}{\@xmultirow{\#4}{\#5}{\#6}{\#7}{\#8}}}}%
5667 }
```

The user-level interface. This is provided if the HTML version was not already given.

```
5668 \providecommand*\multicolumnrow}{\LWR@origmulticolumnrow}
5669 }% \xmultirow defined, so multirow was loaded
5670 }% AtBeginDocument

5671 \end{warpall}
```

## 59.19 Utility macros inside a table

**for HTML output:** 5672 \begin{warpHTML}

```
5673 \newcommand*\LWR@donothing{}{}
```

In case bigdelim is not loaded:

```
5674 \newcommand*\ldelim{}{}
5675 \newcommand*\rdelim{}{}
```

5676 \end{warpHTML}

## 59.20 Special-case tabular markers

**for HTML & PRINT:** 5677 \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.

```
5678 \let\TabularMacro\relax
5679 \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:** 5680 \begin{warpHTML}

```
5681 \newcommand*\{\ResumeTabular\}{%
5682 \global\boolfalse{LWR@exittingtabular}%
5683 \global\boolfalse{LWR@tabularmutemods}%
5684 \LWR@getmynexttoken%
5685 }
```

```
5686 \end{warpHTML}
```

**for PRINT output:** 5687 \begin{warpprint}

```
5688 \newcommand*\{\ResumeTabular\}{}
5689 \end{warpprint}
```

## 59.21 Checking for a new table cell

**for HTML output:** 5690 \begin{warpHTML}

Bool LWR@exittingtabular When \end is found, turns off the next opening data tag.

```
5691 \newbool{LWR@exittingtabular}
```

Bool LWR@tabularmutemods Mutes HTML output for @, !, < and >.

This is used while printing the final row to generate \bottomrules.

```
5692 \newbool{LWR@tabularmutemods}
```

- \LWR@tabledatacolumntag Open a new HTML table cell unless the next token is for a macro which does not create data, such as \hline, \toprule, etc:

```
5693 \newcommand*{\LWR@tabledatacolumntag}%
5694 {%
5695 \LWR@traceinfo{\LWR@tabledatacolumntag}%
```

\show\LWR@mynexttoken to see what tokens to look for

If not any of the below, start a new table cell:

```
5696 \let\mynext\LWR@tabledatasinglecolumntag%
```

If exiting the tabular:

```
5697 \ifdefequal{\LWR@mynexttoken}{\end}%
5698     {\booltrue{\LWR@exitingtabular}}{}%
```

`longtable` can have a caption in a cell

```
5699 \ifdefequal{\LWR@mynexttoken}{\caption}%
5700     {\let\mynext\LWR@donothing}{}%
```

Look for other things which would not start a table cell:

```
5701 \ifdefequal{\LWR@mynexttoken}{\multicolumn}%
5702     {\let\mynext\LWR@donothing}{}%
5703 \ifdefequal{\LWR@mynexttoken}{\multirow}%
5704     {\let\mynext\LWR@donothing}{}%
5705 \ifdefequal{\LWR@mynexttoken}{\multicolumnrow}%
5706     {\let\mynext\LWR@donothing}{}%
```

If an `\mrowcell`, this is a cell to be skipped over:

```
5707 \ifdefequal{\LWR@mynexttoken}{\mrowcell}%
5708     {\let\mynext\LWR@donothing}{}%
```

If an `\mcolrowcell`, this is a cell to be skipped over:

```
5709 \ifdefequal{\LWR@mynexttoken}{\mcolrowcell}%
5710     {\let\mynext\LWR@donothing}{}%
5711 %
5712 \ifdefequal{\LWR@mynexttoken}{\TabularMacro}%
5713     {\let\mynext\LWR@donothing}{}%
5714 %
5715 \ifdefequal{\LWR@mynexttoken}{\hline}%
5716     {\let\mynext\LWR@donothing}{}%
```

```

5717 %
5718 \ifdefequal{\LWR@mynexttoken}{\toprule}%
5719   {\let\mynext\LWR@donothing}{\}%
5720 %
5721 \ifdefequal{\LWR@mynexttoken}{\midrule}%
5722   {\let\mynext\LWR@donothing}{\}%
5723 %
5724 \ifdefequal{\LWR@mynexttoken}{\cmidrule}%
5725   {\let\mynext\LWR@donothing}{\}%
5726 %
5727 \ifdefequal{\LWR@mynexttoken}{\specialrule}%
5728   {\let\mynext\LWR@donothing}{\}%
5729 %
5730 \ifdefequal{\LWR@mynexttoken}{\cline}%
5731   {\let\mynext\LWR@donothing}{\}%
5732 %
5733 \ifdefequal{\LWR@mynexttoken}{\bottomrule}%
5734   {\let\mynext\LWR@donothing}{\}%
5735 %
5736 \ifdefequal{\LWR@mynexttoken}{\warpprintonly}%
5737   {\let\mynext\LWR@donothing}{\}%
5738 %
5739 \ifdefequal{\LWR@mynexttoken}{\warpHTMLonly}%
5740   {\let\mynext\LWR@donothing}{\}%
5741 %
5742 \ifdefequal{\LWR@mynexttoken}{\ldelim}%
5743   {\let\mynext\LWR@donothing}{\}%
5744 %
5745 \ifdefequal{\LWR@mynexttoken}{\rdelim}%
5746   {\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:

```

5747 \mynext%
5748 }

5749 \end{warpHTML}

```

## 59.22 \mrowcell

**for HTML & PRINT:** 5750 \begin{warpall}

**\mrowcell** The user must insert \mrowcell into any \multirow cells which must be skipped.  
⚠ This command has no action during print output.

```
5751 \newcommand*{\mrowcell}{}%
```

```
5752 \end{warpall}
```

## 59.23 \mcolrowcell

**for HTML & PRINT:** 5753 \begin{warpall}

**\mcolrowcell** The user must insert \mcolrowcell into any \multicolumnrow cells which must be skipped. This command has no action during print output.

```
5754 \newcommand*{\mcolrowcell}{}%
```

```
5755 \end{warpall}
```

## 59.24 New \tabular definition

**for HTML output:** 5756 \begin{warpHTML}

These are default definitions in case booktabs is not loaded, and are not expected to be used, but must exist as placeholders.

```
5757 \newcommand*{\LWR@origtoprule}[1]{\hline}
5758 \newcommand*{\LWR@origmidrule}[1]{\hline}
5759 \LetLtxMacro{\LWR@origcmidrule}{\cline}
5760 \newcommand*{\LWR@origbottomrule}[1]{\hline}
5761 \newcommand*{\LWR@origaddlinespace}[1][]{}
5762 \newcommand*{\LWR@origmorecmidrules}(){}
5763 \newcommand*{\LWR@origspecialrule}[3]{\hline}
```

**\LWR@hline** The definition of \hline depends on whether tabls has been loaded. If so, optional space below the line may be specified, but will be ignored.

```
5764 \AtBeginDocument{
5765 @ifpackageloaded{lwarp-tables}
5766 {
5767 \newcommand*{\LWR@hline}[1][]{%
5768     \ifbool{FormatWP}{%
```

---

```

5769     {\LWR@docmidrule{1-\theLWR@tabletotalcols}}%
5770     {\booltrue{\LWR@doinghline}}%
5771     \LWR@getmynexttoken}%
5772 }%
5773 {%
5774 \newcommand*{\LWR@hline}{%
5775     \ifbool{FormatWP}{%
5776         {\LWR@docmidrule{1-\theLWR@tabletotalcols}}%
5777         {\booltrue{\LWR@doinghline}}%
5778         \LWR@getmynexttoken}%
5779 }%
5780 }% 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.

```

5781 \AtBeginDocument{%
5782 \ifundefined{frenchbsetup}{%
5783 {%
5784     \newcommand*{\LWR@nullifyNoAutoSpacing}{}%
5785 }%
5786 {%
5787     \newcommand*{\LWR@nullifyNoAutoSpacing}{%
5788         \NoAutoSpacing%
5789         \renewcommand*{\NoAutoSpacing}{}%
5790         \renewcommand*{\LWR@FBcancel}{}%
5791     }%
5792 }%
5793 }% AtBeginDocument

```

Env LWR@tabular [*vertposition*] {[*colspecs*}]

The new tabular environment will be \let in \LWR@LwarpStart, since siunitx might redefine tabular in the user's document.

```

5794 \StartDefiningTabulars
5795
5796 \newenvironment*{\LWR@tabular}[2] []
5797 {%
5798 \LWR@traceinfo{\LWR@tabular started}%
5799 \addtocounter{\LWR@tabulardepth}{1}%

```

Not yet started a table row:

```
5800 \global\boolefalse{\LWR@startedrow}%

```

Not yet doing any rules:

```
5801 \global\boolfalse{LWR@doinghline}%
5802 \global\boolfalse{LWR@doingtbrule}%
5803 \global\boolfalse{LWR@doigcmidrule}%
```

For babel-french, turn off auto spacing one time, then nullify the autospacing commands since were not compatible with the tabular parsing code.

```
5804 \LWR@nullifyNoAutoSpacing%
```

Have not yet found the end of tabular command. Unmute the @ and ! columns.

```
5805 \boolfalse{LWR@exittingtabular}%
5806 \boolfalse{LWR@tabularmutemods}%
```

Create the table tag:

```
5807 \global\booltrue{LWR@intabularmetadata}%
5808 \LWR@forcenewpage
5809 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
5810 \LWR@parsetablecols{#2}%
```

Table col spec is: \LWR@tablecols which is a string of llccrr, etc.

Do not place the table inside a paragraph:

```
5811 \LWR@stopars%
```

Track column # for setting text-align:

```
5812 \setcounter{LWR@tablecolspos}{1}%
```

Start looking for midrules:

```
5813 \LWR@clearmidrules%
```

\\" becomes a macro to end the table row:

```
5814 \LetLtxMacro{\\"}{\LWR@tabularendofline}%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with \LWR@getmynexttoken to see if the next token might create a new data cell:

The optional parameter for \hline supports the tabls package.

```
5815 \LWR@traceinfo{LWR@tabular: redefining macros}%
5816 \renewcommand*{\hline}{\LWR@chline}%
5817 %
5818 \RenewDocumentCommand{\cline}{m}%
5819 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%

5820 \DeclareDocumentCommand{\toprule}{o d()}
5821   {%
5822     \IfValueTF{##1}%
5823       {\LWR@docmidrule[##1] () {1-\theLWR@tabletotalcols}}%
5824     {%
5825       \ifbool{FormatWP}%
5826         {\LWR@docmidrule[##1] () {1-\theLWR@tabletotalcols}}%
5827         {\booltrue{LWR@doingtbrule}}%
5828     }%
5829   \LWR@getmynexttoken}%
5830 %
5831 \DeclareDocumentCommand{\midrule}{o d()}%
5832   {%
5833     \IfValueTF{##1}%
5834       {\LWR@docmidrule[##1] () {1-\theLWR@tabletotalcols}}%
5835     {%
5836       \ifbool{FormatWP}%
5837         {\LWR@docmidrule[##1] () {1-\theLWR@tabletotalcols}}%
5838         {\booltrue{LWR@doingshline}}%
5839     }%
5840   \LWR@getmynexttoken}%
5841 %
5842 \DeclareDocumentCommand{\cmidrule}{O{\LWR@cmidrulewidth} d() m}%
5843 {\LWR@docmidrule[##1] (##2){##3}\LWR@getmynexttoken}%
5844 %
5845 \DeclareDocumentCommand{\bottomrule}{o d()}
5846   {%
5847     \IfValueTF{##1}%
5848       {\LWR@docmidrule[##1] () {1-\theLWR@tabletotalcols}}%
5849     {%
5850       \ifbool{FormatWP}%
5851         {\LWR@docmidrule[##1] () {1-\theLWR@tabletotalcols}}%
5852         {\booltrue{LWR@doingtbrule}}%
5853     }%
5854   \LWR@getmynexttoken}%
5855 %
5856 \DeclareDocumentCommand{\addlinespace}{o}{}%
5857 \DeclareDocumentCommand{\morecmidrules}{}{}%
5858 \DeclareDocumentCommand{\specialrule}{m m m d()}%
5859 {\LWR@docmidrule[##1] () {1-\theLWR@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.

```

5860 \renewcommand{\multicolumn}{\LWR@htmlemulticolumn}%
5861 \renewcommand*{\mrowcell}{%
5862     \LWR@maybenewtablerow%
5863     \LWR@tabularleftedge%
5864     \global\booltrue{\LWR@skippingmrowcell}%
5865 }%
5866 \renewcommand*{\mcolrowcell}{%
5867     \LWR@maybenewtablerow%
5868     \LWR@tabularleftedge%
5869     \global\booltrue{\LWR@skippingmcolrowcell}%
5870 }%
5871 \LetLtxMacro{\caption}{\LWR@longtabledatacaptiontag}%

```

Reset for new processing:

```

5872 \global\boolearn{\LWR@tableparcell}%
5873 \global\boolearn{\LWR@skippingmrowcell}%
5874 \global\boolearn{\LWR@skippingmcolrowcell}%
5875 \global\boolearn{\LWR@skipatbang}%
5876 \global\boolearn{\LWR@emptyatbang}%

```

Set & for its special meaning inside the tabular:

```

5877 \StartDefiningTabulars%
5878 \protected\gdef&{\LWR@tabularampersand}%

```

Look ahead for a possible table data cell:

```

5879 \LWR@traceinfo{\LWR@tabular: about to \LWR@getmynexttoken}%
5880 \LWR@getmynexttoken%
5881 }%

```

Ending the environment:

```

5882 {%
5883 \LWR@traceinfo{\LWR@tabular ending}%
5884 \ifthenelse{\value{\LWR@tablecolspos}<\value{\LWR@tablecolswidth}}{%
5885     \LWR@tabularfinishrow%
5886 }%
5887 {%
5888     \LWR@closetabledatacell%
5889 }%
5890 \LWR@htmlblocktag{/tr}%
5891 \LWR@htmlblocktag{/table}%
5892 \global\boolearn{\LWR@intabularmetadata}%

```

Unnest one level of tabular:

```
5893 \addtocounter{LWR@tabulardepth}{-1}%
```

Restore & to its usual meaning:

```
5894 \protected\gdef&{\LWR@origampmacro}%
5895 \EndDefiningTabulars%
5896 \LWR@traceinfo{LWR@tabular finished ending}%
5897 }
5898
5899 \EndDefiningTabulars

5900 \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 9 shows the data structures related to cross-referencing.

```
for HTML output: 5901 \begin{warpHTML}
```

### 60.1 Setup

\@currentlabelname To remember the most recently defined section name, description, or caption, for \nameref.

```
5902 \newcommand*{\@currentlabelname}{}%
```

Table 9: Cross-referencing data structures

<b>Original L<sup>A</sup>T<sub>E</sub>X:</b>	(print and HTML)
\refstepcounter: Steps the counter and sets \currentlabel.	
\currentlabel: \p@<ctr>\the<ctr> Updated by \refstepcounter.	
\label: Writes to the .aux file: \newlabel{<label>}{{\currentlabel}{\thepage}}	
\newlabel: When the .aux file is read, sets \r@<label>.	
\r@<label>: Set to: {{\currentlabel}{\thepage}}	
\ref: Returns the first part of \r@<label>.	
\pageref: Returns the second part of \r@<label>.	
<b>Added by l warp:</b>	(HTML only)
\label: Adds HTML tags (section 60.3), plus \splabel data (section 60.2):	
zLWR@name: The section name for this label.	
zLWR@htmlfilenumber: The filenumber or name for this label.	
zLWR@teximagedepth: The teximagedepth for this label.	
zLWR@teximagenumber: The teximagenumber for this label.	
\nameref: Emulated from hyperref for l warp. See section 60.4.	
\ref and \nameref: Adds HTML tags. See section 60.4.	
<b>Added by amsmath:</b>	(print and HTML)
\label: Execution is delayed until the math environment is completed.	
\ltx@label: L <sup>A</sup> T <sub>E</sub> X \label, (HTML: patched by l warp,) later patched by cleveref.	
<b>Added by cleveref:</b>	(print and HTML)
\refstepcounter: Added: sets \cref@currentlabel.	
\cref@currentlabel: (<type>=<ctr> unless an alias is used): [<type>] [<arabic<ctr>>] [<parent ctrs>]{\p@<ctr>\the<ctr>} Also see section 46.4 for use with footnotes.	
\label: Writes to the .aux file: \newlabel{<label>@\cref}{{\cref@currentlabel}{\thepage}}	
\newlabel: (Unchanged.) When the .aux file is read, sets \r@<label>@\cref.	
\r@<label>@\cref: Set to: {{\cref@currentlabel}{\thepage}}	
<b>Utility functions:</b> See \cref@getlabel, \cref@gettype, \cref@getcounter, \cref@getprefix.	
<b>Cross-referencing names:</b> \crefname and \Crefname assign human-readable names for references to this counter type.	
<b>Additionally patched by l warp:</b>	(HTML only)
\cref, etc.: Modified for l warp. See section 72.	
\label inside math: See section 65.5.1.	
<b>Footnotes:</b> See \noteentry in section 46.4.	

---

```
\LWR@stripperiod {<text>} [(.)]
```

Removes a trailing period.

```
5903 \def\LWR@stripperiod#1.\ltx@empty#2\@nil{#1}%
```

```
\LWR@setlatestname {<object name>}
```

Removes \label, strips any final period, and remembers the result.

```
5904 \newcommand*\LWR@setlatestname[1]{%
```

Remove \label and other commands from the name, the strip any final period. See zref-titleref and gettitlestring.

```
5905 \GetTitleStringExpand{#1}%
5906 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
5907 \edef\@currentlabelname{%
5908 \expandafter\LWR@stripperiod\@currentlabelname%
5909 \ltx@empty.\ltx@empty\@nil%
5910 }%
5911 }
```

## 60.2 Zref setup

See:

<http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference>

Create a new property list called special:

```
5912 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
5913 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
5914 \zref@newprop{zLWR@htmlfilename}{%
5915 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilename}%
5916 }%
```

Additional properties for lateximages:

---

```
5917 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
5918 \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:

```
5919 \zref@addprop{special}{zLWR@name}
5920 \zref@addprop{special}{zLWR@htmlfilenumber}
5921 \zref@addprop{special}{zLWR@lateximagedepth}
5922 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
5923 \newcommand*{\LWR@spref}[2]{%
5924 \zref@extractdefault{\#1}{\#2}{??}}
```

`\LWR@nameref {<label>}` Returns the section name for this label:

```
5925 \newcommand*{\LWR@nameref}[1]{%
5926 \LWR@sref{\#1}{zLWR@name}%
5927 }
```

`\LWR@htmlfileref {<label>}` Returns the file number for this label:

```
5928 \newcommand*{\LWR@htmlfileref}[1]{%
5929 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
5930 \LWR@sref{\#1}{zLWR@htmlfilenumber}%
5931 }
```

`\LWR@lateximagedepthref {<label>}` Returns the `lateximagedepth` for this label:

```
5932 \newcommand*{\LWR@lateximagedepthref}[1]{%
5933 \LWR@sref{\#1}{zLWR@lateximagedepth}%
5934 }
```

`\LWR@lateximagenumberref {<label>}` Returns the `lateximagenumber` for this label:

```
5935 \newcommand*{\LWR@lateximagenumberref}[1]{%
5936 \LWR@sref{\#1}{zLWR@lateximagenumber}%
5937 }
```

`\LWR@splabel {<label>}` Sanitize the name and then creates the label:

```
5938 \newcommand*{\LWR@splabel}[1]{%
5939 \LWR@setlatestname{\@currentlabelname}%
5940 \zref@labelbylist{\#1}{special}}
```

### 60.3 Labels

\LWR@subsublabel {*label*} Creates an HTML id tag.

```
5941 \newcommand*{\LWR@subsublabel}[1]{%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```
5942 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}>}{0}%
5943 {}%
5944 {% not lateximage
```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```
5945 \ifbool{LWR@doingstartpars}%
5946 {}% pars allowed
5947 \ifbool{LWR@doingapar}%
5948 {}% par started
5949 \LWR@htmltag{a id="#1"\}\LWR@htmltag{/a}%
5950 {}% par started
5951 {}% par not started
5952 \LWR@stoppars%
5953 \LWR@htmltag{a id="#1"\}\LWR@htmltag{/a}%
5954 \LWR@startpars%
5955 {}% par not started
5956 {}% pars allowed
5957 {}% pars not allowed
5958 \LWR@htmltag{a id="#1"\}\LWR@htmltag{/a}%
5959 {}% pars not allowed
5960 {}% not lateximage
5961 }
```

\LWR@newlabel {*label*} [*type*]

\label during HTML output when not in math mode, removing extra spaces around the label, as done by regular L<sup>A</sup>T<sub>E</sub>X \label.

clevereref later encases this to add its own cross-referencing.

The optional *type* is per the ntheorem package, and is ignored.

```
5962 \NewDocumentCommand{\LWR@newlabel}{m o}{%
5963 \LWR@traceinfo{LWR@newlabel: starting}%
5964 \LWR@traceinfo{LWR@newlabel: !#1!}%
5965 % \@bsphack%
```

Create a traditional L<sup>A</sup>T<sub>E</sub>X label, as modified by cleveref:

```
5966 \LWR@origlabel{#1}%
```

Create a special label which holds the section number, LWR@htmlfilename, LWR@lateximagedepth, and LWR@lateximagenumber:

```
5967 \LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
5968 \LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
5969 \LWR@traceinfo{LWR@newlabel: LWR@htmlfilename is \theLWR@htmlfilename}%
5970 \LWR@splabel{#1}%
5971 \LWR@subsublabel{#1}%
5972 % \@esphack%
5973 \LWR@traceinfo{LWR@newlabel: done}%
5974 }
```

## 60.4 References

\LWR@startref {<label>} (Common code for \ref and \nameref.)

Open an HTML tag reference to a filename, # character, and a label.

```
5975 \newcommand*{\LWR@startref}[1]%
5976 {%
5977 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
5978 \LWR@traceinfo{LWR@startref A: !#1!}%
```

Create the filename part of the link:

```
5979 \LWR@htmlltag{a href="#"%
5980 \LWR@traceinfo{LWR@startref B}%
5981 \LWR@htmllrefsectionfilename{#1}%
5982 \LWR@traceinfo{LWR@startref C}%
5983 \#}
```

Create the destination id:

See if LWR@lateximagedepth is unknown:

```
5984 \LWR@traceinfo{LWR@startref D: !#1!}%
5985 \ifthenelse{\equal{\LWR@lidref}{??}}{%
```

“??” if LWR@lateximagedepth is unknown, so create a link with an unknown destination:

```

5986 {%
5987     \LWR@traceinfo{\LWR@startref D0: ??}%
5988     ??%
5989 }%

```

If `\LWR@lateximagedepth` is known. Use a `lateximage` if the depth is greater than zero, or a regular link otherwise:

```

5990 {%
5991     \LWR@traceinfo{\LWR@startref D1: \LWR@lidref}%
5992     \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}{%
5993         {%
5994             \LWR@traceinfo{\LWR@startref D2: \LWR@lidref}%
5995             lateximage\LWR@lateximagenumberref{#1}%
5996         }%
5997         {%
5998             \LWR@traceinfo{\LWR@startref D3}%
5999             #1%
6000         }%
6001 }%
6002 \LWR@traceinfo{\LWR@startref E}%

```

Closing quote:

```

6003 "}{}%
6004 \LWR@traceinfo{\LWR@startref F}%
6005 }

```

`\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)”.

```

6006 \NewDocumentCommand{\LWR@subnewref}{m m}{%
6007 \LWR@traceinfo{\LWR@subnewref #1 #2}%
6008 \LWR@startref{#1}%
6009 \LWR@origref{#2}%
6010 \LWR@htmltag{/a}%
6011 }

```

`\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`.

```

6012 \NewDocumentCommand{\LWR@newref}{s m}{%
6013 \LWR@traceinfo{\LWR@newref #2}%

```

---

```

6014 \IfBooleanTF{#1}%
6015 {\LWR@origref{#2}}%
6016 {\LWR@subnewref{#2}{#2}}%
6017 }
```

\pagerefPageFor Text for page references.

```
6018 \newcommand*{\pagerefPageFor}{\see }
```

\pageref \* {*label*} Create an internal document reference, or just the unlinked number if starred, per hyperref.

```

6019 \NewDocumentCommand{\LWR@newpageref}{s m}{%
6020 \IfBooleanTF{#1}%
6021 {(\pagerefPageFor\LWR@origref{#2})}%
6022 {(\cpageref{#2})}%
6023 }
```

\nameref {*label*}

```

6024 \newcommand*{\nameref}[1]{%
6025 \LWR@traceinfo{\nameref A}%
6026 \LWR@startref{#1}%
6027 \LWR@traceinfo{\nameref B}%
6028 \LWR@nameref{#1}%
6029 \LWR@traceinfo{\nameref C}%
6030 \LWR@htmltag{/a}%
6031 \LWR@traceinfo{\nameref D}%
6032 }
```

\Nameref {*label*} In print, adds the page number. In HTML, does not.

```
6033 \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.

```
6034 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
6035 % \EmulatesPackage{hyperref}[2015/08/01]%. Disabled. Do not do this.
```

Create a link with a text name:

```
\LWR@subhyperref {⟨URL⟩} {⟨text⟩}
```

```
6036 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
6037 \LWR@htmltag{a href="#" target="\_\_blank"\LWR@orignewline}#2\LWR@htmltag{/a}%
6038 \LWR@ensuredoingapar%
6039 }
```

```
\LWR@subhyperrefclass {⟨URL⟩} {⟨text⟩} {⟨htmlclass⟩}
```

```
6040 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
6041 \LWR@htmltag{a href="#"%
6042 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
6043 \LWR@ensuredoingapar%
6044 }
```

```
\href [⟨options⟩] {⟨URL⟩} {⟨text⟩}
```

Create a link with accompanying text:

```
6045 \NewDocumentCommand{\LWR@hrefb}{O{} m +m}{%
6046 \LWR@subhyperref{#2}{#3}%
6047 \endgroup%
6048 \LWR@ensuredoingapar%
6049 }
6050
6051 \newcommand{\href}{%
6052 \LWR@ensuredoingapar%
6053 \begingroup%
6054 \catcode`\_=12
6055 \LWR@hrefb%
6056 }
```

```
\nolinkurl {⟨URL⟩}
```

Print the name of the link without creating the link:

```
6057 \newcommand*{\LWR@nolinkurlb}[1]{#1\endgroup\LWR@ensuredoingapar}
6058
6059 \newcommand{\nolinkurl}{%
6060 \LWR@ensuredoingapar%
6061 \begingroup\catcode`\_=12
6062 \LWR@nolinkurlb%
6063 }
```

```
\url {<URL>}
```

Create a link whose text name is the address of the link. The `url` package may redefine `\url`, so it is `\let` to `\LWR@urlahere` and also redefined by `lwarp-url`.

```
6064 \newcommand*{\LWR@urlb}[1]{%
6065 \href{#1}{#1}%
6066 \endgroup%
6067 \LWR@ensuredoingapar%
6068 }
6069
6070 \newcommand{\LWR@urla}{%
6071 \LWR@ensuredoingapar%
6072 \begingroup\catcode`\_=12
6073 \LWR@urlb%
6074 }
6075
6076 \let\url\LWR@urla
```

```
\LWR@subinlineimage [<alttag>] {<class>} {<filename>} {<extension>} {<style>}
```

```
6077 \newcommand*{\LWR@subinlineimage}[5]{[]}{%
6078 \ifthenelse{\equal{#1}{}}{%
6079 {\LWR@htmlltag{img src="#3.#4" alt="#3" style="#5" class="#2"{} }}%
6080 {\LWR@htmlltag{img src="#3.#4" alt="#1" style="#5" class="#2"{} }}%
6081 }%
6082 \end{warpHTML}
```

Table 10: Float data structures

---

For each <type> of float (figure, table, etc.) there exists the following:

**counter <type>:** A counter called <type>, such as figure, table.

\<type>name: Name. \figurename prints “Figure”, etc.

\ext@<type>: File extension. \ext@figure prints “lof”, etc.

\fps@<type>: Placement.

\the<type>: Number. \thetable prints the number of the table, etc.

\p@<type>: Parent’s number. Prints the number of the [within] figure, etc.

\fnum@<type>: Prints the figure number for the caption.

\<type>name \the<type>, “Figure 123”.

\<type>: Starts the float environment. \figure or \begin{figure}

\end<type>: Ends the float environment. \endfigure or \end{figure}

\tf@<ext>: The L<sup>A</sup>T<sub>E</sub>X file identifier for the output file.

**LWR@have<type>:** A boolean rememebering 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 10 shows the data structure associated with each <type> of float.

\@makecaption is redefined to print the float number and caption text, separated by \CaptionSeparator, which works with the babel package to adjust the caption separator according to the language. French, for example, uses an en-dash instead of a colon: “Figure 123 – Caption text”.

## 61.1 Float captions

**for HTML output:** 6083 \begin{warpHTML}

\LWR@floatbegin {*type*} [*placement*]

Begins a \newfloat environment.

```
6084 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
6085 \ifbool{FormatWP}{\newline}{}}%
6086 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
6087 \addtocounter{LWR@thisfloat}{1}%
6088 \booltrue{LWR@freezethisfloat}%
6089 \begingroup%
```

Settings while inside the environment:

6090 \LWR@origraggedright%

Open an HTML figure tag:

```
6091 \LWR@htmlltag{figure id="autofloat-\arabic{LWR@thisfloat}" class="#1"}%
6092 \ifbool{FormatWP}{%
6093   \LWR@orignewline%
6094   \LWR@BlockClassWP{}{}{wp#1}%
6095 }{}}%
6096 \renewcommand*{\@capttype}{#1}%
6097 \caption@settype{#1}%
6098 \LWR@startpars%
6099 \ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%
6100
6101 === begin #1 ===
6102
6103 }{}}%
6104 }
```

\@float Support packages which create floats directly.  
 \@dblfloat  
 6105 \let\@float\LWR@floatbegin  
 6106 \let\@dblfloat\LWR@floatbegin

\LWR@floatend Ends a \newfloat environment.

```
6107 \newcommand*\LWR@floatend{%
6108 \ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%
6109 === end ===
6111
6112 }{}%
6113 \LWR@stoppars%
```

Close an HTML figure tag:

```
6114 \ifbool{FormatWP}{\endLWR@BlockClassWP}{%
6115 \LWR@htmlelementend{figure}%
6116 \endgroup%
6117 \boolfalse{\LWR@freezethisfloat}%
6118 \LWR@startpars%
6119 \ifbool{FormatWP}{\newline}{%
6120 }
```

\end@float Support packages which create floats directly.

```
\end@dblfloat
6121 \let\end@float\LWR@floatend
6122 \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.

```
6123 \newcounter{\LWR@thisfloat}
```

Bool \LWR@freezethisfloat Prevents multiple increments of \LWR@thisfloat inside a float.

```
6124 \newbool{\LWR@freezethisfloat}
6125 \boolfalse{\LWR@freezethisfloat}
```

\LWR@maybeincthisfloat

```
6126 \newcommand*\LWR@maybeincthisfloat{%
6127 \ifbool{\LWR@freezethisfloat}{}{\addtocounter{\LWR@thisfloat}{1}}%
6128 }
```

\@capttype Remembers which float type is in use.

```
6129 \newcommand*\@capttype{}
```

### 61.1.1 Caption inside a float environment

\CaptionSeparator How to separate the float number and the caption text.

```
6130 \AtBeginDocument{\providecommand*\CaptionSeparator}{:~}}
```

\@makecaption {\langle name and num \rangle} {\langle text \rangle}

Prints the float type and number, the caption separator, and the caption text.

```
6131 \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 **LETX** PDF file, the **LETX** page number at that point is stored in **LWR@latestautopage**, (and the associated filename is remembered by the special **LETX** 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.

```
6132 \newcounter{LWR@nextautofloat}
6133 \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.

```
6134 \newcommand*\LWRsetnextfloat[2]{%
6135 \setcounter{LWR@nextautopage}{#1}%
6136 \setcounter{LWR@nextautofloat}{#2}%
6137 }
```

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.

```
6138 \newcounter{LWR@latestautopage}
6139 \setcounter{LWR@latestautopage}{1}
```

Env LWR@figcaption Encapsulates a caption inside <figcaption>, and if FormatWP then also a <div> with an italic style.

```

6140 \newenvironment*{LWR@figcaption}
6141 {%
6142 \LWR@htmlblocktag{figcaption}
6143 \ifbool{FormatWP}{%
6144 \begin{BlockClass}[font-style:italic]{italic}
6145 \LWR@origvspace*\backslashbaselineskip}
6146 }{}%
6147 }
6148 {%
6149 \ifbool{FormatWP}{\end{BlockClass}}{}%
6150 \LWR@htmlblocktag{/figcaption}%
6151 }

```

```

6152 \let\LWR@origcaption\begin\caption\begin
6153 \let\LWR@origcaption\end\caption\end
6154 \let\LWR@orig@@par\@@par

```

\LWR@caption@begin Low-level patches to create HTML tags for captions.

```

6155 \newcommand{\LWR@caption@begin}{%
6156 {
6157 \LWR@traceinfo{\LWR@caption@begin}%

```

Keep par and minipage changes local:

```
6158 \begingroup%
```

The caption code was not allowing the closing par tag:

```
6159 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a minipage or \parbox inside the caption:

```

6160 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{}{%
6161 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{{\#5}}%

```

Enclose the original caption code inside an HTML tag:

```

6162 \LWR@figcaption%
6163 \LWR@origcaption\begin%
6164 }

```

\LWR@caption@end Low-level patches to create HTML tags for captions.

```

6165 \newcommand{\LWR@caption@end}{%
6166 {%
6167 \LWR@origcaption@end}%

```

Subcaptions were being over-written by the closing HTML tag:

```
6168 \LWR@origvspace*\{\baselineskip}\%
```

Closing tag:

```

6169 \endLWR@figcaption\%
6170 \endgroup\%
6171 % \leavevmode% avoid bad space factor (0) error
6172 \LWR@traceinfo{\LWR@caption@end: done}\%
6173 }%

```

\caption@begin Low-level patches to create HTML tags for captions.  
\caption@end  
6174 \AtBeginDocument{  
6175 \let\caption@begin\LWR@caption@begin  
6176 \let\caption@end\LWR@caption@end  
6177 }

\captionlistentry Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```

6178 \let\LWR@origcaptionlistentry\captionlistentry
6179
6180 \renewcommand*{\captionlistentry}{%
6181 \LWR@maybeinthisfloat\%
6182 \LWR@ensuredoingapar\%
6183 \LWR@htmltag{a id="autofloat-\arabic{\LWR@thisfloat}"{} }\LWR@htmltag{/a}\%
6184 \LWR@origcaptionlistentry\%
6185 }
6186
6187 \def\LWR@LTcaptionlistentry{%
6188 \LWR@ensuredoingapar\%
6189 \LWR@htmltag{a id="autofloat-\arabic{\LWR@thisfloat}"{} }\LWR@htmltag{/a}\%
6190 \bgroup
6191 \@ifstar{\egroup\LWR@LT@captionlistentry}\% gobble *
6192 {\egroup\LWR@LT@captionlistentry}\%
6193 \def\LWR@LT@captionlistentry#1{%
6194 \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.

---

```

6195 \let\LWR@origaddcontentsline\addcontentsline
6196
6197 \renewcommand*{\addcontentsline}[3]{%
6198   \ifthenelse{\equal{#1}{toc}}{}{%
6199     \ifthenelse{\equal{#1}{thm}}{\csdef{ext@\#2}{thm}}{%
6200       \addtocontents{\@nameuse{ext@\#2}}{%
6201         \protect\LWRsetnextfloat{%
6202           {\arabic{\LWR@latestautopage}}{%
6203             {\arabic{\LWR@thisfloat}}}}{%
6204           }% addtocontents
6205     }% not toc
6206   \LWR@origaddcontentsline{#1}{#2}{#3}%
6207 }

```

**\captionof** Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

6208 \AtBeginDocument{
6209 \let\LWR@origcaptionof\captionof
6210
6211 \renewcommand*{\captionof}{%
6212 \LWR@maybeinthisfloat{%
6213 \LWR@stoppars
6214 \LWR@htmlltag{a id="autofloat-\arabic{\LWR@thisfloat}"{} }\LWR@htmlltag{/a}}%
6215 \LWR@origcaptionof%
6216 }
6217 }

6218 \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 L<sup>A</sup>T<sub>E</sub>X infrastructure is used for TOC, along with some patches to generate HTML output.

**for HTML output:** 6219 \begin{warpHTML}

## 62.1 Reading and printing the TOC

```
\LWR@myshorttoc {⟨toc/lof/lot⟩}
```

Reads in and prints the TOC/LOF/LOT at the current position. While doing so, makes the @ character into a normal letter to allow formatting commands in the section names.

Unlike in regular  $\text{\LaTeX}$ , the file is not reset after being read, since the TOC may be referred to again in each HTML page, and is used for the sidetoc.

```
6220 \newcommand*\LWR@myshorttoc[1]{  
6221 \LWR@ensuredoingapar
```

Only if the file exists:

```
6222 \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 pdflatex, 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.

```
6223 \begingroup  
6224 % \ifxetexorluatex%  
6225 % \else  
6226 % \inputencoding{latin1}% currently disabled  
6227 % \fi  
6228 \makeatletter
```

Read in the toc file:

```
6229 \@input{\jobname.\#1}  
6230 % \makeatother  
6231 \endgroup  
6232 }%  
6233 {}%  
6234 }
```

```
\LWR@subtableofcontents {⟨toc/lof/lot⟩} {⟨sectionstarname⟩}
```

Places a TOC/LOF/LOT at the current position.

```
6235 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```
6236 \ifundefined{chapter}
6237 {\LWR@closeprevious{\LWR@depthsection}}
6238 {\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
6239 \LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
6240 \ifdefined{chapter}{\section*{\#2}}{\chapter*{\#2}}
```

Create a new HTML nav containing the TOC/LOF/LOT:

```
6241 \LWR@htmlelementclass{nav}{#1}
```

Create the actual list:

```
6242 \LWR@myshorttoc{#1}
```

Close the nav:

```
6243 \LWR@htmlelementclassend{nav}{#1}
6244 }
```

`\@starttoc {<ext>}`

Patch `\@starttoc` to encapsulate the TOC inside HTML tags:

```
6245 \let\LWR@orig@starttoc\@starttoc
6246
6247 \renewcommand{\@starttoc}[1]{
6248 \LWR@htmlelementclass{nav}{#1}
6249 \LWR@orig@starttoc{#1}
6250 \LWR@htmlelementclassend{nav}{#1}
6251 }
```

`\tableofcontents` Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.

```
6252 \let\LWR@origtableofcontents\tableofcontents
6253
6254 \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:

```
6255 \ifboolexpr{bool{FormatWP} and bool{WPMarkTOC}}{  
6256  
6257 === table of contents ===  
6258  
6259 }  
6260 {
```

Copy the .toc file to .sidetoc for printing the sidetoc. The original .toc file is renewed when \tableofcontents is finished.

```
6261     \LWR@copyfile{\jobname.toc}{\jobname.sidetoc} %  
6262     \LWR@printpendingfootnotes  
6263     \LWR@origtableofcontents  
6264 }  
6265 }
```

#### \listoffigures

```
6266 \let\LWR@origlistoffigures\listoffigures  
6267  
6268 \renewcommand*{\listoffigures}{  
6269 \ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{  
6270  
6271 === list of figures ===  
6272  
6273 }  
6274 {  
6275     \LWR@printpendingfootnotes  
6276     \LWR@origlistoffigures  
6277 }  
6278 }
```

#### \listoftables

```
6279 \let\LWR@origlistoftables\listoftables  
6280  
6281 \renewcommand*{\listoftables}{  
6282 \ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{  
6283  
6284 === list of tables ===  
6285  
6286 }  
6287 {  
6288     \LWR@printpendingfootnotes  
6289     \LWR@origlistoftables
```

```
6290 }
6291 }
```

## 62.2 High-level TOC commands

\listof {⟨type⟩} {⟨title⟩}

Emulate the \listof command from the float package (section 131). Used to create lists of custom float types. Also used to redefine the standard L<sup>A</sup>T<sub>E</sub>X \listoffigures and \listoftables commands.

```
6292 \NewDocumentCommand{\listof}{m +m}{%
6293 \LWR@subtableofcontents{\nameuse{ext@#1}}{#2}
6294 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
6295 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
6296   \jobname.\csuse{ext@#1}\relax
6297 }
```

## 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.

---

```
6298 \end{warpHTML}
```

**for HTML & PRINT:** 6299 \begin{warpall}

**Ctrl SideTOCDepth** Controls how deep the side-TOC gets. Use a standard L<sup>A</sup>T<sub>E</sub>X section level similar to tocdepth.

```
6300 \newcounter{SideTOCDepth}
6301 \setcounter{SideTOCDepth}{1}
```

\sidetocname Holds the default name for the sidetoc.

```
6302 \newcommand{\sidetocname}{Contents}
6303 \end{warpall}
```

**for HTML output:** 6304 \begin{warpHTML}

\LWR@sidetoc Creates the actual side-TOC.

```
6305 \newcommand*\LWR@sidetoc{%
6306 \LWR@forcenewpage
6307 \LWR@stoppars
6308 }
```

The entire sidetoc is placed into a nav of class sidetoc.

```
6309 \LWR@htmlelementclass{nav}{sidetoc}
6310
6311 \setcounter{tocdepth}{\value{SideTOCDepth}}
6312 }
```

The title is placed into a <div> of class sidetoctitle, and may contain paragraphs.

```
6313 \begin{BlockClass}{sidetoctitle}
6314 \sidetocname
6315 \end{BlockClass}
```

The table of contents is placed into a <div> of class sidetoccontents.

```
6316 \begin{BlockClass}{sidetoccontents}
6317 \LinkHome
6318
6319 \LWR@myshorttoc{sidetoc}
6320 \end{BlockClass}
6321 \LWR@htmlelementclassend{nav}{sidetoc}
6322 }
```

## 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:

```
6323 \renewcommand*{\numberline}[1]{%
6324 \LWR@sectionnumber{#1}\quad%
6325 }
```

\hypertoc {<1: depth>} {<2: type>} {<3: name>} {<4: page>}

Called by \l@section, etc. to create a hyperlink to a section.

The autopage label is always created just after the section.

**#1** is depth

**#2** is section, subsection, etc.

**#3** the text of the caption

**#4** page number

```
6326 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
6327 \ifthenelse{\cnttest{#1}{<=}{\value{tocdepth}}}{%
6328   \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```
6329   \LWR@subhyperrefclass{%
6330     \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
6331     \LWR@stoppars%
6332 }
6333 {}
6334 }
```

Ctr **lofdepth** TOC depth for figures.

```
6335 \newcounter{lofdepth}
6336 \setcounter{lofdepth}{1}
```

Ctr **lotdepth** TOC depth for tables.

```
6337 \newcounter{lotdepth}
6338 \setcounter{lotdepth}{1}
```

\hypertocfloat {<1: depth>} {<2: type>} {<3: ext of parent>} {<4: caption>} {<5: page>}

- #1 is depth
- #2 is figure, table, etc.
- #3 is lof, lot, of the parent.
- #4 the text of the caption
- #5 page number

```
6339 \newcommand{\hypertocfloat}[5]{%
6340 \LWR@startpars
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
6341 \@ifundefined{c@#3depth}{%
6342 \newcounter{#3depth}%
6343 \setcounter{#3depth}{1}%
6344 }{}}%
```

Respond to `lofdepth`, etc.:

```
6345 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
6346 \ifthenelse{\cnttest{#1}{<=}{\arabic{#3depth}}}{%
6347     \LWR@startpars%
```

Create an HTML link to `filename#autofloat-(float number)`, with text of the caption, of the given HTML class.

```
6348     \LWR@subhyperrefclass{%
6349         \LWR@htmlrefsectionfilename{autopage-\arabic{\LWR@nextautopage}}%
6350         \#autofloat-\arabic{\LWR@nextautofloat}}%
6351         {#4}{toc#2}%
6352         \LWR@stoppars%
6353 }{}}%
6354 }
```

Automatically called by `\contentsline`:

```
6355 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
6356 \DeclareDocumentCommand{\l@chapter}{m m}
6357     {\hypertoc{0}{chapter}{#1}{#2}}
6358 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
6359 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
6360 \renewcommand{\l@subsubsection}[2]
6361     {\hypertoc{3}{subsubsection}{#1}{#2}}
6362 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
```

---

```

6363 \renewcommand{\l@subparagraph}{[2]{\hypertoc{5}{subparagraph}{#1}{#2}}}
6364 \renewcommand{\l@figure}{[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}}
6365 \renewcommand{\l@table}{[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}}

6366 \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:** 6367 \begin{warpHTML}

```

6368 \newcounter{LWR@autoindex}
6369 \setcounter{LWR@autoindex}{0}
6370
6371 \newcounter{LWR@autoglossary}
6372 \setcounter{LWR@autoglossary}{0}

```

\printindex

```

6373 \let\LWR@origprintindex\printindex
6374
6375 \renewcommand*{\printindex}
6376 {
6377 \LWR@startpars
6378 \LWR@origprintindex
6379 }

```

Env theindex

```

6380 \@ifundefined{chapter}
6381 {\newcommand*{\LWR@indexsection}[1]{\section*{#1}}}
6382 {\newcommand*{\LWR@indexsection}[1]{\chapter*{#1}}}
6383
6384 \renewenvironment*{theindex}{%
6385 \LWR@indexsection{\indexname}%
6386 \let\item\LWR@indexitem%

```

```
6387 \let\subitem\LWR@indexsubitem%
6388 \let\subsubitem\LWR@indexsubsubitem%
6389 }{}
```

\LWR@indexitem

```
6390 \newcommand{\LWR@indexitem}{%
6391
6392 \InlineClass{indexitem}{}%
6393 }
```

\LWR@indexitem

```
6394 \newcommand{\LWR@indexsubitem}{%
6395
6396 \InlineClass{indexsubitem}{}%
6397 }
```

\LWR@indexitem

```
6398 \newcommand{\LWR@indexsubsubitem}{%
6399
6400 \InlineClass{indexsubsubitem}{}%
6401 }
```

\@wrindex {*term*} Redefined to write the LWR@latestautopage counter instead of page

```
6402 \def\@wrindex#1{%
6403 \addtocounter{LWR@autoindex}{1}%
6404 \LWR@newlabel{LWRindex-\theLWR@autoindex}%
6405 \protected@write\@indexfile{}}%
6406 {\string\indexentry{#1}{\theLWR@autoindex}}%
6407 \endgroup
6408 \esphack}
```

\@wrglossary {*term*} Redefined to write the LWR@latestautopage counter instead of page

```
6409 \def\@wrglossary#1{%
6410 \addtocounter{LWR@autoglossary}{1}%
6411 \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
6412 \protected@write\@glossaryfile{}}%
6413 {\string\glossaryentry{#1}{\theLWR@autoglossary}}%
6414 \endgroup
6415 \esphack}
```

\hyperindexref {*autosecnumber*}

---

\hyperindexref{web address} is inserted into \*.ind by the xindy style file lwarp.xdy

```
6416 \newcommand*\{\hyperindexref\}[1]{\nameref{LWRindex-\#1}}
6417 \end{warpHTML}
```

**for PRINT output:** A null command for print mode, in case hyperref was not used:

```
6418 \begin{warpprint}
6419 \newcommand{\hyperindexref\}[1]{\#1}
6420 \end{warpprint}
```

**for HTML & PRINT:** For the glossaries package, try to prevent an error where \glo@name was not found:

```
6421 \begin{warpall}
6422 \providecommand{\glo@name}{}
6423 \end{warpall}
```

## 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.

```
6424 \newcommand*\{\LWR@restoreorigformatting\}{%
6425 \LWR@traceinfo{\LWR@restoreorigformatting}%
6426 \linespread{1}%
6427 \RenewDocumentCommand{\InlineClass}{o m +m}{\#\#3}%
6428 \RenewDocumentEnvironment{BlockClass}{o m}{\{}{\}}%
6429 \renewcommand{\BlockClassSingle}[2]{\#\#2}%
6430 \LetLtxMacro{\hspace}{\LWR@orighspace}%
6431 \LetLtxMacro{\rule}{\LWR@origrule}%
6432 \let\,\,\LWR@origcomma% disable HTML short unbreakable space
6433 \let\textellipsis\LWR@origtextellipsis%
6434 \let\textless\LWR@origtextless%
6435 \let\textgreater\LWR@origtextgreater%
6436 \LetLtxMacro{\textrm}{\LWR@origtextrm}%
6437 \LetLtxMacro{\textsf}{\LWR@origtextsf}%
6438 \LetLtxMacro{\texttt}{\LWR@origtexttt}%
6439 \LetLtxMacro{\textbf}{\LWR@origtextbf}%
6440 \LetLtxMacro{\textmd}{\LWR@origtextmd}%
6441 \LetLtxMacro{\textit}{\LWR@origtextit}%
6442 \LetLtxMacro{\textsl}{\LWR@origtextsl}%
```

```
6443 \LetLtxMacro{\textsc}{\LWR@origtextsc}%
6444 \LetLtxMacro{\textup}{\LWR@origtextup}%
6445 \LetLtxMacro{\textnormal}{\LWR@origtextnormal}%
6446 \LetLtxMacro{\emph}{\LWR@origemph}%
6447 \LetLtxMacro{\rmfamily}{\LWR@origrmfamily}%
6448 \LetLtxMacro{\sffamily}{\LWR@origsffamily}%
6449 \LetLtxMacro{\ttfamily}{\LWR@origttfamily}%
6450 \LetLtxMacro{\bfseries}{\LWR@origbfseries}%
6451 \LetLtxMacro{\mdseries}{\LWR@origmdseries}%
6452 \LetLtxMacro{\upshape}{\LWR@origupshape}%
6453 \LetLtxMacro{\slshape}{\LWR@origslshape}%
6454 \LetLtxMacro{\scshape}{\LWR@origscshape}%
6455 \LetLtxMacro{\itshape}{\LWR@origitshape}%
6456 \LetLtxMacro{\em}{\LWR@origem}%
6457 \LetLtxMacro{\normalfont}{\LWR@orignormalfont}%
6458 \let\sp\LWR@origsp%
6459 \let\sb\LWR@origsb%
6460 \LetLtxMacro{textsuperscript}{\LWR@origtextsupserscript}%
6461 \LetLtxMacro{@textsuperscript}{\LWR@orig@textsuperscript}%
6462 \LetLtxMacro{textsubscript}{\LWR@origtextsubscript}%
6463 \LetLtxMacro{@textsubscript}{\LWR@orig@textsubscript}%
6464 \LetLtxMacro{underline}{\LWR@origunderline}%
6465 \let~\LWR@origtilde%
6466 \let\enskip\LWR@origenskip%
6467 \let\quad\LWR@origquad%
6468 \let\qquad\LWR@origqquad%
6469 \LetLtxMacro{\tabular}{\LWR@origtabular}%
6470 \LetLtxMacro{\endtabular}{\LWR@origendtabular}%
6471 \LetLtxMacro{\toprule}{\LWR@origtoprule}%
6472 \LetLtxMacro{\midrule}{\LWR@origmidrule}%
6473 \LetLtxMacro{\cmidrule}{\LWR@origcmidrule}%
6474 \LetLtxMacro{\bottomrule}{\LWR@origbottomrule}%
6475 \LetLtxMacro{\addlinespace}{\LWR@origaddlinespace}%
6476 \LetLtxMacro{\morecmidrules}{\LWR@origmorecmidrules}%
6477 \LetLtxMacro{\specialrule}{\LWR@origspecialrule}%
6478 \let\newline\LWR@orignewline%
6479 \LetLtxMacro{\raisebox}{\LWR@origraisebox}%
6480 \LetLtxMacro{\includegraphics}{\LWR@origincludegraphics}%
6481 \LetLtxMacro{\scalebox}{\LWR@origscalebox}%
6482 \LetLtxMacro{\rotatebox}{\LWR@origrotatebox}%
6483 \let\reflectbox\LWR@origreflectbox%
6484 \LetLtxMacro{\resizebox}{\LWR@origresizebox}%
6485 \let\framebox\LWR@origframebox%
6486 \let\makebox\LWR@origmakebox%
6487 \let\fbox\LWRprint@fbox%
6488 \let\fboxBlock\LWRprint@fbox%
6489 \LetLtxMacro{\fminipage}{\LWRprint@fminipage}%
6490 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}%
6491 \LetLtxMacro{\minipage}{\LWR@origminipage}%
6492 \let\endminipage\LWR@endminipage%
```

```
6493 \LetLtxMacro{\parbox}{\LWR@origparbox}%
6494 \let\TeX\LWR@origTeX%
6495 \let\LaTeX\LWR@origLaTeX%
6496 \let\LaTeXe\LWR@origLaTeXe%
6497 \renewcommand*{\Xe}[X\textsubscript{E}]%
6498 \LetLtxMacro{\ensuredmath}{\LWR@origensuredmath}%
6499 %
6500 \LWR@restoreorigaccents%
6501 %
6502 \LWR@FBcancel%
6503 }
```

## 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  $\mathcal{M}\mathcal{S}$  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  $\mathcal{M}\mathcal{S}$  environment, although the \tag macro will still appear inside the  $\text{\LaTeX}$  math expression.

### 65.1.3 MathJax option

**MathJax math option**

Prog MathJax

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

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.

---

<sup>13</sup>See section 245 regarding fonts and fractions.

<https://groups.google.com/forum/#!topic/mathjax-users/jUtewUcE2bY>

### subequations

- MathJax itself does not support subequations. This may be improved by parsing the  $\text{\TeX}$  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

### custom MathJax macros

- 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`.
- MathJax does not automatically support custom  $\text{\TeX}$  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: 6504 `\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.

```
6505 \let\LWR@origtextdollar\$
6506
6507 \renewcommand*{\$}{%
6508 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}>{0}}{%
6509 {\LWR@origtextdollar}%
6510 {\LWR@htmllagc{span}\LWR@origtextdollar\LWR@htmllagc{/span}}%
6511 }}
```

Ctr LWR@externalfilecnt Counter for the external files which are generated and then referenced from the HTML:

```
6512 \newcounter{LWR@externalfilecnt}

6513 \LetLtxMacro{\LWR@origdollar\$}
6514 \LetLtxMacro{\LWR@secondorigdollar$}% balance for editor syntax highlighting

6515 \LetLtxMacro{\LWR@origopenparen}(%
6516 \LetLtxMacro{\LWR@origcloseparen})%
6517 \LetLtxMacro{\LWR@origopenbracket}[%
6518 \LetLtxMacro{\LWR@origclosebracket}]
```

\\$ Redefine the dollar sign to place math inside a lateximage, or use MathJax:  
\$\$

```
6519 \begingroup
6520 \catcode`\$=\active%
6521 \protected\gdef${@ifnextchar$\LWR@doubledollar\LWR@singledollar}%
```

\LWR@doubledollar Redefine the double dollar sign to place math inside a lateximage, or use MathJax:

```
6522 \gdef\LWR@doubledollar$#1$${
```

If MathJax or formatting for a word processor, print the L<sup>E</sup>T<sub>X</sub> expression:

```
6523 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{}
```

For MathJax, print the math between \[ and \]:

```
6524 {
6525
6526     \textbackslash[%
6527     \LWR@HTMLsanitize{#1}%
6528     \textbackslash]
```

```
6529
6530 }% mathjax
```

For SVG, print the math inside a `lateximage`, with an `<alt>` tag of the  $\text{\TeX}$  code:

```
6531 {%
6532   not mathjax
6533   \begin{lateximage}%
6534     [\textbackslash{}textbackslash{}lash{}[] \LWR@HTMLsanitize{#1} \textbackslash{}textbackslash{}lash{}[]]]%
6535     \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
6536   \end{lateximage}%
6537
6538 }%
6539 }%
```

`\LWR@singledollar` Redefine the single dollar sign to place math inside a `lateximage`, or use MathJax:

```
6540 \gdef\&LWR@subsingledollar#1{%
6541 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{}
```

For MathJax, print the math between `\(` and `\)`:

```
6542 {%
6543   \textbackslash{}textbackslash{}lash{(\LWR@HTMLsanitize{#1}\textbackslash{}textbackslash{}lash{})}%
6544 }%
6545 mathjax
```

For SVG, print the math inside a `lateximage`, with an `<alt>` tag of the  $\text{\TeX}$  code:

```
6545 {%
6546   not mathjax
6547   \begin{lateximage}%
6548     [\textbackslash{}textbackslash{}lash{(\LWR@HTMLsanitize{#1}\textbackslash{}textbackslash{}lash{})}]%
6549     \LWR@origdollar#1\LWR@origdollar%
6550   \end{lateximage}%
6551 }%
6552
6553 \gdef\&LWR@singledollar#1${%
6554 \&LWR@subsingledollar{#1}}%
6555 }
```

`\(` Redefine to the above dollar macros.

`\)`

```
6556 \gdef\(#1\){$#1$}
6557 \gdef\[#1\]{$$#1$$}
6558
6559 \endgroup
6560
```

6561  
6562

\@ensuredmath {*expression*} Not yet adapted to \warp.

```
6563 \LetLtxMacro{\LWR@origensuredmath}{\@ensuredmath}
6564
6565 \renewcommand{\@ensuredmath}[1]{%
6566   \ifmmode%
6567     \LWR@origensuredmath{#1}%
6568   \else%
6569     \LWR@subsingle dollar{\relax#1}%
6570   \fi%
6571 }
```

Remove the old `math` and `displaymath` environments:

```
6572 \let\math\relax
6573 \let\endmath\relax
6574 \let\displaymath\relax
6575 \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.

```
6576 \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.

```
6577 \NewEnviron{displaymath}{\expandafter[\BODY]\@ignoretrue}
```

### 65.3 MathJax support

`Ctr LWR@nextequation` Used to add one to compute the next equation number.

```
6578 \newcounter{LWR@nextequation}
```

`\LWR@syncmathjax` Sets the MathJax equation format and number for the following equations.

These MathJax commands are printed inside “\(`” and “\)`” characters. They are printed to HTML output, not interpreted by L<sup>A</sup>T<sub>E</sub>X.

```
6579 \newcommand*\{LWR@syncmathjax}\{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
6580 \ifcsdef{thechapter}{
6581 \InlineClass{hidden}{
6582 \textbackslash(
6583 \textbackslash{}seteqsection \{\thechapter\}
6584 \textbackslash)
6585 }
6586 }
6587 {}% not using chapters
```

MathJax doesn't allow setting the equation number to 1:

```
6588 \ifthenelse{\cnttest{\value{equation}}>0}{%
6589 {%
```

Tell MathJax that the next set of equations begins with the current  $\text{\LaTeX}$  equation number, plus one.

```
6590 \setcounter{LWR@nextequation}{\value{equation}}
6591 \addtocounter{LWR@nextequation}{1}
```

Place the MathJax command inside “ $\text{\(}$ ” and “ $\text{\)}$ ” characters, to be printed to  $\text{HTML}$ , not interpreted by  $\text{\LaTeX}$ .

```
6592 \InlineClass{hidden}{
6593 \textbackslash(
6594 \textbackslash{}seteqnumber \{\arabic{LWR@nextequation}\}
6595 \textbackslash)
6596 }
6597 {}% not eq > 0
6598 }
```

```
\LWR@hidelatexequation {{environment}} {{contents}}
```

Creates the  $\text{\LaTeX}$  version of the equation inside an  $\text{HTML}$  comment.

```
6599 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop  $\text{HTML}$  paragraph handling and open an  $\text{HTML}$  comment:

```
6600 \LWR@stopars
6601 \LWR@htmlopencomment
6602
```

Start the  $\text{\LaTeX}$  math environment inside the  $\text{HTML}$  comment:

```
6603 \begingroup
6604 \csuse{LWR@orig#1}
```

While in the math environment, restore various commands to their  $\text{\TeX}$  meanings.

```
6605 \LWR@restoreorigformatting
```

See `\LWR@htmlmathlabel` in section 65.5.1.

Print the contents of the equation:

```
6606 #2
```

End the  $\text{\TeX}$  math environment inside the HTML comment:

```
6607 \csuse{LWR@origend#1}
6608 \endgroup
6609
```

Close the HTML comment and resume HTML paragraph handling:

```
6610 \LWR@htmclosecomment
6611 \LWR@startpars
6612 }
```

```
\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  $\text{\TeX}$ .

```
6613 \NewDocumentCommand{\LWR@addmathjax}{m +m} {%
```

Enclose the MathJax environment inside printed “`\(`” and “`\)`” characters.

```
6614 \LWR@origtilde\LWR@orignewline
6615 \textbackslash{}begin\{\#1\}
```

Print the contents, sanitizing for HTML special characters.

```
6616 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\#2}}
```

Close the MathJax environment:

```
6617 \textbackslash{}end\{\#1\}
6618 \LWR@orignewline
6619 }
```

## 65.4 Equation environment

Remember existing equation environment:

```
6620 \let\LWR@origequation\equation
6621 \let\LWR@origendequation\endequation
```

Remove existing equation environment:

```
6622 \let\equation\relax
6623 \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  $\text{\TeX}$  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.

```
6624 \NewEnviron{equation}{%
6625 }
```

If `mathjax` or `FormatWP`, print the  $\text{\TeX}$  expression:

```
6626 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{%
```

MathJax output:

```
6627 {
```

Print commands to synchronize MathJax's equation number and format to the current  $\text{\TeX}$  chapter/section and equation number:

```
6628 \LWR@syncmathjax
```

Print the  $\text{\TeX}$  math inside an `HTML` comment:

```
6629 \LWR@hidelatexequation{equation}{\BODY}
6630 }
```

SVG output: Create the `lateximage` along with an HTML `<alt>` tag having an equation number, the `\TeX` equation environment commands, and the contents of the environment's `\BODY`.

```
6631 {%
  not mathjax
```

Begin the `lateximage` with an `<alt>` tag containing the math source:

```
6632   \begin{lateximage}[(\LWR@equationtag) \textbackslash begin\{equation\}] %
6633     \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}} %
6634   \textbackslash end\{equation\}]\% alt tag
```

Create the actual `\TeX`-formatted equation inside the `lateximage` using the contents of the environment.

```
6635   \LWR@origequation
6636   \BODY% contents collected by NewEnviron
6637   \LWR@origendequation
6638   \end{lateximage}%
6639 }%
6640 }% not mathjax
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
6641 }[%]
6642   \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6643   {%
6644     \LWR@addmathjax{equation}{\BODY}%
6645   }%
6646
6647 ]
```

## 65.5 AMS Math environments

### 65.5.1 Support macros

Bool `LWR@amsmultiline` True if processing a multiline environment.

To compensate for `multiline`-specific code, `LWR@amsmultiline` is used to add extra horizontal space in `\LWR@htmlmathlabel` if it is used in an `amsmath` environment which is not a `multiline` environment and not an equation.

```
6648 \newbool{LWR@amsmultiline}
6649 \boolfalse{LWR@amsmultiline}
```

```
\LWR@htmlmathlabel {⟨label⟩}
```

lwarp points `\ltx@label` here. This is used by `\label` when inside a  $\text{\TeX}$  AMS math environment's math display environment.

`\LWR@origltx@label` points to the  $\text{\TeX}$  original, modified by lwarp, then by amsmath, then by cleveref.

```
6650 \newcommand*{\LWR@htmlmathlabel}[1]{%
6651 \LWR@traceinfo{\LWR@htmlmathlabel #1}}%
```

If `mathjax` or `FormatWP`, print the  $\text{\TeX}$  expression:

```
6652 \ifboolexpr{bool{mathjax} \or ( bool{FormatWP} \and bool{WPMarkMath} ) }{%
6653 {}}
```

The combined  $\text{\TeX}$  & HTML label is printed in a `\text` field:

```
6654 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
6655 \ifbool{\LWR@amsmultiline}{}{\hspace*{\totwidth@}}
```

Temporarily end the HTML comment, insert the  $\text{\TeX}$  & HTML label, then resume the HTML comment. `\@firstofone` is required to remove extra braces introduced by the amsmath package.)

```
6656 \LWR@htmlclosecomment%
6657 \LWR@origltx@label{#1}%
6658 \LWR@htmlopencomment%
6659 }% text
6660 }% mathjax
6661 {%
6662 \LWR@origltx@label{#1}%
6663 }%
6664 }
```

`\LWR@beginhideamsmath` Starts hiding  $\text{\TeX}$  math inside an HTML comment.

```
6665 \newcommand*{\LWR@beginhideamsmath}{%
6666 \LWR@stopars
6667 \LWR@origtilde\LWR@orignewline
6668 \LWR@htmlopencomment
6669
6670 \begingroup
6671 \LWR@restoreorigformatting
```

---

6672 }

\LWR@endhideamsmath Ends hiding  $\text{\TeX}$  math inside an HTML comment.

```
6673 \newcommand*\LWR@endhideamsmath{  
6674 \endgroup  
6675  
6676 \LWR@htmlclosecomment  
6677 \LWR@orignewline  
6678 \LWR@startpars  
6679 }
```

### 65.5.2 Environment patches

The following amsmath environments already collect their contents in `\@envbody` for further processing. `eqnarray` is not an  $\mathcal{AM}$ S 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  $\mathcal{AM}$ S environment.

For `MathJax`: Each environment is syched with  $\text{\TeX}$ 's equation numbers, typeset with  $\text{\TeX}$  inside an `HTML` comment, then printed to `HTML` output for `MathJax` to process.

`Env eqnarray` This environment is not an  $\mathcal{AM}$ S environment and thus its body is not automatically captured, so the `environ` package is used to capture the environment into `\BODY`.

```
6680 \let\LWR@origeqnarray\eqnarray  
6681 \let\LWR@origendeqnarray\endeqnarray
```

To remember whether the starred environment was used, and thus whether to number the equations:

```
6682 \newbool{\LWR@numbereqnarray}  
6683 \booltrue{\LWR@numbereqnarray}
```

Common code used by `eqnarray` and `Beqnarray` (from `fancybox`):

```
6684 \newcommand{\LWR@eqnarrayfactor}{%
```

If `mathjax` or `FormatWP`, print the  $\text{\TeX}$  expression:

---

```
6685 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6686 {%
```

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:

```
6687     \LWR@syncmathjax
6688     \boolexpr{LWR@amsmultiline}
6689     \ifbool{LWR@numbereqnarray}
6690     {
```

If numbering the equations, execute a copy inside an HTML comment block:

```
6691     \LWR@beginhideamsmath
6692     \LWR@origeqnarray
6693     \BODY
6694     \LWR@origendeqnarray
6695     \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MathJax to interpret:

```
6696     \LWR@addmathjax{eqnarray}{\BODY}
6697     }%
6698     {%
not LWR@numbereqnarray
```

If not numbering equations, just create the contents for MathJax:

```
6699     \LWR@addmathjax{eqnarray*}{\BODY}
6700     }%
LWR@numbereqnarray
6701 }%
mathjax
6702 {%
not mathjax
6703     \ifbool{LWR@numbereqnarray}
6704     {
```

For numbered SVG equations, first create a `lateimage` with an `alt` attribute containing sanitized copy of the source code:

```
6705     \begin{lateimage}[(\LWR@startingequationtag--\LWR@equationtag)
6706             \LWR@addmathjax{eqnarray}{\BODY}]
```

Then create the image contents using an actual `eqnarray`:

```
6707     \LWR@origeqnarray
6708     \BODY
6709     \LWR@origendeqnarray
6710     \end{lateimage}
6711     }%
6712     {%
not LWR@numbereqnarray
```

If not numbered, do the same, but an extra \nonumber seems to be required:

```

6713      \begin{lateximage}[\LWR@addmathjax{eqnarray*}{\BODY}]
6714      \LWR@origeqnarray
6715      \BODY
6716      \nonumber
6717      \LWR@origendeqnarray
6718      \end{lateximage}
6719  }% \LWR@numbereqnarray
6720 }% not mathjax

```

Default to number equations in the future:

```

6721 \booltrue{\LWR@numbereqnarray}
6722 }

```

eqnarray itself is made with a blank line before and after to force it to be on its own line:

```

6723 \RenewEnviron{eqnarray}
6724 {%
6725
6726 \LWR@eqnarrayfactor
6727
6728 }

```

The starred version is patched to turn off the numbering:

```

6729 \csgpreto{eqnarray*}{\boolfalse{\LWR@numbereqnarray}}

```

The following *AMS* environments are more easily patched in-place:

Env **multiline**

```

6730 \BeforeBeginEnvironment{multiline}{
6731
6732 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{
6733 {
6734     \LWR@syncmathjax
6735     \booltrue{\LWR@amsmultiline}
6736     \LWR@beginhideamsmath
6737 }
6738 {
6739     \latexitimage[\LWR@amsmathbodynumbered{multiline}]
6740 }
6741 }
6742
6743 \AfterEndEnvironment{multiline}{}

```

```
6744  
6745 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%  
6746 {  
6747     \LWR@endhideamsmath  
6748     \boolefalse{LWR@amsmultiline}  
6749     \LWR@addmathjax{multiline}{\the\@envbody}  
6750 }  
6751 {\endlateximage}  
6752  
6753 }
```

Env  **multiline\***

```
6754 \BeforeBeginEnvironment{multiline*}{  
6755  
6756 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%  
6757 {  
6758     \LWR@syncmathjax  
6759     \booletrue{LWR@amsmultiline}  
6760     \LWR@beginhideamsmath  
6761 }  
6762 {  
6763     \lateximage[\LWR@amsmathbody{multiline*}]  
6764 }  
6765 }  
6766  
6767 \AfterEndEnvironment{multiline*}{  
6768  
6769 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%  
6770 {  
6771     \LWR@endhideamsmath  
6772     \boolefalse{LWR@amsmultiline}  
6773     \LWR@addmathjax{multiline*}{\the\@envbody}  
6774 }  
6775 {\endlateximage}  
6776  
6777 }  
6778
```

Env  **gather**

```
6779 \BeforeBeginEnvironment{gather}{  
6780  
6781 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%  
6782 {  
6783     \LWR@syncmathjax  
6784     \boolefalse{LWR@amsmultiline}  
6785     \LWR@beginhideamsmath
```

```
6786 }
6787 {
6788     \lateximage[\LWR@amsmathbodynumbered{gather}]
6789 }
6790 }
6791
6792 \AfterEndEnvironment{gather}{

6793
6794 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6795 {
6796     \LWR@endhideamsmath
6797     \LWR@addmathjax{gather}{\the\@envbody}
6798 }
6799 {\endlateximage}
6800
6801 }
```

Env **gather\***

```
6802 \BeforeBeginEnvironment{gather*}{

6803
6804 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6805 {
6806     \LWR@syncmathjax
6807     \boolfalse{\LWR@amsmultiline}
6808     \LWR@beginhideamsmath
6809 }
6810 {
6811     \lateximage[\LWR@amsmathbody{gather*}]
6812 }
6813 }
6814
6815 \AfterEndEnvironment{gather*}{

6816
6817 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6818 {
6819     \LWR@endhideamsmath
6820     \LWR@addmathjax{gather*}{\the\@envbody}
6821 }
6822 {\endlateximage}
6823
6824 }
```

Env **align**

```
6825 \BeforeBeginEnvironment{align}{

6826
6827 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

```
6828 {
6829     \LWR@syncmathjax
6830     \boolexpr{\LWR@amsmultiline}
6831     \LWR@beginhideamsmath
6832 }
6833 {
6834     \lateximage[\LWR@amsmathbody{aligned}]
6835 }
6836 }
6837
6838 \AfterEndEnvironment{aligned}{
6839
6840 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{
6841 {
6842     \LWR@endhideamsmath
6843     \LWR@addmathjax{aligned}{\the\@envbody}
6844 }
6845 {\endlatexitimage}
6846
6847 }

Env align*
6848 \BeforeBeginEnvironment{aligned*}{

6849
6850 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{
6851 {
6852     \LWR@syncmathjax
6853     \boolexpr{\LWR@amsmultiline}
6854     \LWR@beginhideamsmath
6855 }
6856 {
6857     \latexitimage[\LWR@amsmathbody{aligned*}]
6858 }
6859 }
6860
6861 \AfterEndEnvironment{aligned*}{

6862
6863 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{
6864 {
6865     \LWR@endhideamsmath
6866     \LWR@addmathjax{aligned*}{\the\@envbody}
6867 }
6868 {\endlatexitimage}
6869
6870 }
```

Env flalign

```
6871 \BeforeBeginEnvironment{flalign}{

6872
6873 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6874 {
6875     \LWR@syncmathjax
6876     \boolefalse{\LWR@amsmultiline}
6877     \LWR@beginhideamsmath
6878 }
6879 {
6880     \lateximage[\LWR@amsmathbodynumbered{flalign}]
6881 }
6882 }
6883
6884 \AfterEndEnvironment{flalign}{

6885
6886 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6887 {
6888     \LWR@endhideamsmath
6889     \LWR@addmathjax{flalign}{\the\envbody}
6890 }
6891 {\endlatexitimage}
6892
6893 }
```

Env flalign\*

```
6894 \BeforeBeginEnvironment{flalign*}{

6895
6896 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6897 {
6898     \LWR@syncmathjax
6899     \boolefalse{\LWR@amsmultiline}
6900     \LWR@beginhideamsmath
6901 }
6902 {
6903     \latexitimage[\LWR@amsmathbody{flalign*}]
6904 }
6905 }
6906
6907 \AfterEndEnvironment{flalign*}{

6908
6909 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6910 {
6911     \LWR@endhideamsmath
6912     \LWR@addmathjax{flalign*}{\the\envbody}
6913 }
6914 {\endlatexitimage}
```

---

```

6915
6916 }

6917 \end{warpHTML}

```

## 66 Lateximages

### 66.1 Description

**Env lateximage** A `lateximage` is a piece of the document which is typeset in  $\text{\TeX}$  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 preceding page and ends on following page, and instructions are written to `lateximage.txt` for `lwarpmk` to extract the `\lateximage` from the page of the `PDF` file then generate an accompanying `.svg` file image file. Meanwhile, instructions to show this image are placed into the `HTML` file after the comment.

An `HTML <span>` is created to hold both the `HTML` comment, which will have the `pdftotext` conversion, and also the link to the final `.svg` image.

A  $\text{\TeX}$  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:** 6918 `\begin{warpHTML}`

**Ctr LWR@lateximagenumber** Sequence the images.

```

6919 \newcounter{LWR@lateximagenumber}
6920 \setcounter{LWR@lateximagenumber}{0}

```

Ctr LWR@lateximagedepth Do not create \lateximage inside of \lateximage.

```
6921 \newcounter{LWR@lateximagedepth}
6922 \setcounter{LWR@lateximagedepth}{0}
```

A few utility macros to write special characters:

```
6923 \edef\LWR@hashmark{\string#} % for use in \write
6924 \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`.

```
6925 \newcounter{LWR@LIpage}
6926 \end{warpHTML}
```

### 66.3 Font size

**for HTML & PRINT:** 6927 `\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.*

```
6928 \newcommand*{\LateximageFontSizeName}[1]{\large}
6929 \end{warpall}
```

### 66.4 Sanitizing math expressions for HTML

**for HTML output:** 6930 `\begin{warpHTML}`

\LWR@HTMLsanitize `{<text>}`

Math expressions are converted to `lateximages`, and some math environments may contain "&", "<", or ">", which should not be allowed inside an HTML `<alt>` tag, so must convert them to HTML entities.

Two versions follow, depending on expansion needs. There may be a better way...

```
6931 \newcommand{\LWR@HTMLsanitize}[1]{%
```

```

6932 \begingroup%
6933 \LWR@FBcancel%
6934 \protect\StrSubstitute{\detokenize{#1}}%
6935 {\detokenize{&}}%
6936 {\detokenize{&}}[\LWR@strresult]%
6937 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6938 {\detokenize{<}}%
6939 {\detokenize{&lt;}}%
6940 [\LWR@strresult]%
6941 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6942 {\detokenize{>}}%
6943 {\detokenize{&gt;}}%
6944 [\LWR@strresult]%
6945 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6946 {\detokenize{##}}%
6947 {\#}%
6948 [\LWR@strresult]%
6949 \LWR@strresult%
6950 \endgroup%
6951 }

```

\LWR@HTMLsanitizeexpand {*text*}

This version expands the argument before sanitizing it.

```

6952 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
6953 \begingroup%
6954 \LWR@FBcancel%
6955 \protect\StrSubstitute{\detokenize\expandafter{#1}}%
6956 {\detokenize{&}}%
6957 {\detokenize{&}}%
6958 [\LWR@strresult]%
6959 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6960 {\detokenize{<}}%
6961 {\detokenize{&lt;}}%
6962 [\LWR@strresult]%
6963 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6964 {\detokenize{>}}%
6965 {\detokenize{&gt;}}%
6966 [\LWR@strresult]%
6967 \LWR@strresult%
6968 \endgroup%
6969 }

```

## 66.5 Equation numbers

Ctr LWR@startingequation For use with `lateximage` and multi-line numbered equations. Remembers the next

equation number so that it may be printed in the alt tag.

```

6970 \newcounter{LWR@startingequation}
6971
6972 \@ifundefined{chapter}
6973 {
6974 \renewcommand{\theLWR@startingequation}{%
6975 \arabic{LWR@startingequation}%
6976 }
6977 }
6978 {% chapter defined
6979 \renewcommand{\theLWR@startingequation}{%
6980 \ifnumcomp{\value{chapter}}{>}{0}{\arabic{chapter}. }{}%
6981 \arabic{LWR@startingequation}%
6982 }
6983 }
```

**Bool** True for the first equation tag, false for later tags in the same environment.  
**LWR@isstartingequation**

```
6984 \newbool{LWR@isstartingequation}
```

**\LWR@startingequationtag** Prints the starting equation number or tag.

```
6985 \let\LWR@startingequationtag\theLWR@startingequation
```

**\LWR@equationtag** Prints the ending equation number or tag.

```
6986 \let\LWR@equationtag\theequation
```

Only if `svg` math, patch `\tag` after packages have loaded, in case someone else modified `\tag`.

```

6987 \AtBeginDocument{
6988
6989 \ifbool{mathjax}{}{%
6990 \patchcmd{\tag}{\theLWR@startingequation}{\theequation}{\relax}{\relax}}
```

**\LWR@remembertag {⟨tag⟩}**

For use inside the math environments while using `svg` math. Sets `\theLWR@startingequation` and `\theequation` to the given tag.

```

6990 \NewDocumentCommand{\LWR@remembertag}{m}{%
6991 \ifbool{LWR@isstartingequation}{%
6992 }{%
6993 \global\boolfalse{LWR@isstartingequation}%
6994 \xdef\LWR@startingequationtag{\#1}}}
```

```

6995 }%
6996 { }%
6997 \xdef\LWR@equationtag{#1}%
6998 }%

```

Patches for  $\mathcal{M}\mathcal{S}$  math \tag macro to remember the first tag:

```

6999 \LetLtxMacro{\LWR@origmake@df@tag@@}{\make@df@tag@@}
7000 \LetLtxMacro{\LWR@origmake@df@tag@@@}{\make@df@tag@@@}
7001
7002 \renewcommand*{\make@df@tag@@}[1]{%
7003 \LWR@remembertag{#1}%
7004 \LWR@origmake@df@tag@@{#1}%
7005 }
7006
7007 \renewcommand*{\make@df@tag@@@}[1]{%
7008 \LWR@remembertag{#1}%
7009 \LWR@origmake@df@tag@@@{#1}%
7010 }
7011
7012 }% not mathjax
7013 }% AtBeginDocument

```

## 66.6 HTML <alt> tags

`\LWR@amsmathbody {<envname>}` For use inside the optional argument to a `lateximage` to add the contents of a AMS math environment to the <alt> tag.

```

7014 \newcommand*{\LWR@amsmathbody}[1]
7015 {%
7016 \textbackslash\begin{#1} %
7017 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}%
7018 \textbackslash\end{#1}%
7019 }

```

`\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.

```

7020 \newcommand*{\LWR@amsmathbodynumbered}[1]
7021 {%
7022 \ifnumcomp{\value{\LWR@startingequation}}{=}{\value{equation}}{%
7023 {(\LWR@equationtag)}%
7024 {(\LWR@startingequationtag--\LWR@equationtag)} %
7025 \LWR@amsmathbody{#1} %
7026 }

```

## 66.7 lateximage

```
Env  lateximage  [<alt> tag]
7027 \catcode`\$=\active%
7028
7029
7030 \NewDocumentEnvironment{lateximage}{O{image}}
7031 {%
7032 \LWR@traceinfo{lateximage: starting on page \arabic{page}}%
```

Nested lateximages remain one large lateximage:

```
7033 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}>}{0}{}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
7034 {%
7035     \addtocounter{LWR@lateximagedepth}{1}%
7036 }%
```

Otherwise, this is the outer-most lateximage:

```
7037 {% start of outer-most lateximage
```

Remember the next equation number to be allocated, in case it must be printed in a multi-equation environment:

```
7038     \setcounter{LWR@startingequation}{\value{equation}}%
7039     \addtocounter{LWR@startingequation}{1}%
7040     \booltrue{LWR@isstartingequation}%
7041     \let\LWR@startingequationtag\theLWR@startingequation%
7042     \let\LWR@equationtag\theequation%
```

Starting a new lateximage:

```
7043     \addtocounter{LWR@lateximagenumber}{1}%
7044     \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%
```

While inside a lateximage, locally do not use mathjax:

```
7045     \boolfalse{mathjax}
```

Be sure that are doing a paragraph:

```
7046     \LWR@ensuredoingapar%
```

Next file:

```
7047 \addtocounter{LWR@externalfilecnt}{1}%
7048 \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%
```

Figure out what the next page number will be:

```
7049 \setcounterpageref{LWR@LIpage}{\LWR@lateximage\theLWR@lateximagenumber}%
7050 \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:

```
7051 \LWR@htmlltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
7052 class="lateximagesource"{} } \LWR@orignewline
```

Write instructions to the lateximages.txt file:

```
7053 \LWR@traceinfo{lateximage: about to write to lateximages.txt}%
7054 \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.

```
7055 \LWR@traceinfo{lateximage: about to create open comment}%
7056 \LWR@htmlopencomment%
```

One level deeper:

```
7057 \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
7058 \LWR@traceinfo{lateximage: about to create a new page}%
7059 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
7060 \LWR@traceinfo{lateximage: about to create minipage}%
7061 \LWR@origminipage{6in}%
7062 \csuse{\LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for \hspace, etc. inside a lateximage.

```
7063 \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
7064 \LWR@restoreorigformatting%
```

Use full-page footnotes instead of minipage footnotes. These become HTML footnotes.

```
7065 \def\@mpfn{footnote}%
7066 \def\thempfn{\thefootnote}%
7067 \let\@footnotetext\LWR@footnotetext%
```

Create the LWR@lateximage<number> label:

```
7068 \LWR@traceinfo{lateximage: about to create label}%
7069 \LWR@origlabel{LWR@lateximage\arabic{LWR@lateximagenumber}}%
7070 \LWR@traceinfo{lateximage: finished creating the label}%
```

Enable print-mode math functions:

```
7071 \LetLtxMacro$\LWR@origdollar%
7072 \catcode`$=3% math shift
7073 \LetLtxMacro{(\LWR@origopenparen}%
7074 \LetLtxMacro{)\LWR@origcloseparen}%
7075 \LetLtxMacro{[\LWR@origopenbracket}%
7076 \LetLtxMacro{]\LWR@origclosebracket}%
7077 }% end of outer-most lateximage
7078 \LWR@traceinfo{lateximage: finished start of environment}%
7079 }% end of \begin{lateximage}
```

**\end{lateximage}** When the environment closes:

```
7080 {%
7081 \LWR@traceinfo{lateximage: starting end of environment}%
}
```

Nested more than one deep?

```
7082 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}>}{1}{}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
7083 {\addtocounter{LWR@lateximagedepth}{-1}}%
```

If this is the outer-most lateximage:

```
7084 {%
7085 \end{outer-most lateximage}
```

Finish the lateximage minipage and start a new PDF page:

```
7085 \LWR@origendminipage%
7086 \LWR@orignewpage%
7087 \LWR@origscriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by pdftotext:

```
7088 \LWR@htmclosecomment{}\\LWR@orignewline%
7089 \\LWR@traceinfo{lateximage}: The page after the image is \\arabic{page}%%
```

Create a link to the lateximage, allowing its natural height:

```
7090 \\LWR@subinlinelimage[#1]{lateximage}%
7091 {lateximages\\OSPathSymbol{}\\latexitimage-\\theLWR@externalfilecnt}{svg}{}%
```

Be sure that are doing a paragraph:

```
7092 \\LWR@ensuredoingapar%
```

Close the HTML span which has the pdftotext comment and also the link to the .svg image:

```
7093 \\LWR@htmntag{/span}%
7094 \\ifbool{HTMLDebugComments}{%
7095 \\LWR@htmcomment{End of latexitimage}%
7096 }{}%
7097 % \\LWR@orignewline% Removed to prevent extra space.
```

Undo one lateximage level:

```
7098 \\addtocounter{LWR@latexitagedepth}{-1}%
7099 }% end of outer-most lateximage
7100 \\LWR@traceinfo{lateximage: done}%
7101 }%
7102 \\catcode`\\$=3% math shift
7103 \\end{warpHTML}
```

**for PRINT output:** 7104 \\begin{warpprint}
7105 % \\newenvironment{lateximage}[1][]{\\minipage{\\ linewidth}}{\\endminipage}
7106 \\newenvironment{lateximage}[1][]{\\{}\\}}
7107 \\end{warpprint}

## 67 center, flushleft, flushright

**for HTML output:** 7108 \\begin{warpHTML}

Env center Replace center functionality with css tags:

```

7109 \renewenvironment*{center}
7110 {
7111 \LWR@forcenewpage
7112 \ifbool{FormatWP}
7113 {\BlockClass{text-align:center}{center}}
7114 {\BlockClass{center}}
7115 }
7116 {\endBlockClass}
```

Env **flushright**

```

7117 \renewenvironment*{flushright}
7118 {
7119 \LWR@forcenewpage
7120 \ifbool{FormatWP}
7121 {\BlockClass{text-align:right}{flushright}}
7122 {\BlockClass{flushright}}
7123 }
7124 {\endBlockClass}
```

Env **flushleft**

```

7125 \renewenvironment*{flushleft}
7126 {
7127 \LWR@forcenewpage
7128 \ifbool{FormatWP}
7129 {\BlockClass{text-align:left}{flushleft}}
7130 {\BlockClass{flushleft}}
7131 }
7132 {\endBlockClass}

7133 \end{warpHTML}
```

## 68 Pre-loaded packages

**for HTML output:** 7134 \begin{warpHTML}

If `textcomp` was loaded before `lwarp`, perhaps as part of the font-related packages, explicitly load the `lwarp` patches now:

```

7135 \@ifpackageloaded{textcomp}
7136 {
7137 \LWR@origRequirePackage{lwarp-textcomp}
7138 }
7139 {}
```

If `graphics` or `graphicx` were loaded before `l warp`, perhaps by `xunicode`, explicitly load the `l warp` patches now:

```
7140 \@ifpackageloaded{graphics}
7141 {
7142 \LWR@origRequirePackage{l warp-graphics}
7143 }
7144 {}

7145 \end{warpHTML}
```

## 69 Siunitx

Pkg `siunitx` The `l warp` 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:** 7146 \begin{warpHTML}

Options for `siunitx`:

```
7147 \PassOptionsToPackage{
7148   detect-mode=true,
7149   per-mode=symbol,% fraction is not seen by pdftotext
7150 %   text-celsius = {\protect\LWRsiunitx@degree{}C},
7151 %   text-degree = {\protect\LWRsiunitx@degree},
7152 }{siunitx}
7153

7154 \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,

- ⚠ no file extension** and also a .svg, .png, or .jpg version of the same image. These should be referred to without a 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.

##### 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 L<sup>A</sup>T<sub>E</sub>X, 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:

```
7155 \begin{warpprint}
7156 \define@key{Gin}{class}{}{}
```

Print-mode additions for the overpic package. See section 188 for the HTML version.

```
7157 \AtBeginDocument{
7158 \@ifpackageloaded{overpic}{
7159 \newcommand*\{\overpicfontsize}{12}
7160 \newcommand*\{\overpicfontskip}{14}
7161 }{}}
7162 }
7163 \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 warpall 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:** 7164 \begin{warpall}

After xparses may have been loaded ...

```
7165 \AtBeginDocument{
... and only if xcolor was loaded:
7166 \@ifpackageloaded{xcolor}{

7167 \LWR@traceinfo{patching xcolor}
```

\colorboxBlock \colorboxBlock is the same as \colorbox:

```
7168 \LetLtxMacro\colorboxBlock\colorbox
```

In HTML mode, the following is done when xcolor is loaded. Following is the print-mode action:

```
7169 \warpprintonly{
7170 \LetLtxMacro\LWRprint@colorboxBlock\colorbox
7171 \LetLtxMacro\LWRorigprint\fcolorbox\fcolorbox
```

```
7172 \LetLtxMacro{\LWRorigprint}{\fcolorbox{Block}{\fcolorbox{
```

\fcolorbox [⟨framemode⟩] {⟨framecolor⟩} [⟨boxmode⟩] {⟨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.)

```
7174 \newsavebox{\LWR@colorminipagebox}
7175
7176 \DeclareDocumentCommand{\LWRprint@fcolorbox}{o m o m +m}{%
7177 \LWR@traceinfo{\LWRprint@fcolorbox #2 #4}%

```

Pre-load the contents into an LR box so that they can be used inside a \fcolorbox:

```
7178 \begin{lrbox}{\LWR@colorminipagebox}%
7179 #5%
7180 \end{lrbox}%
```

Sort out the various optional arguments and the background color of none. In each case, the LRbox is placed inside a `\fcolorbox`.

```

7181 \ifthenelse{\equal{#4}{none}}%
7182 {%
7183     \LWR@traceinfo{background is none}%
7184     {\% scope the \colorlet
7185         \colorlet{LWR@currentcolor}{.}%
7186         \color{#2}%
7187         \fbox{%
7188             \color{LWR@currentcolor}%
7189             \usebox{\LWR@colorminipagebox}%
7190         }%
7191     }%
7192 }%
7193 {%
7194     \LWR@traceinfo{background not none}%
7195     \IfValueTF{#1}%
7196     {%
7197         \IfValueTF{#3}%
7198         {\LWRorigprint@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
7199         {\LWRorigprint@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
7200     }%
7201 {%
7202     \IfValueTF{#3}%
7203     {\LWRorigprint@fcolorbox[#2]{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
7204     {\LWRorigprint@fcolorbox[#2]{#4}{\usebox{\LWR@colorminipagebox}}}%
7205 }%
7206 {%
7207     \LWR@traceinfo{background is none}%
7208     {\% scope the \colorlet
7209         \colorlet{LWR@currentcolor}{.}%
7210         \color{#2}%
7211         \fbox{%
7212             \color{LWR@currentcolor}%
7213             \usebox{\LWR@colorminipagebox}%
7214         }%
7215     }%
7216 }%
7217 }%
7218 }%

```

```
7206 }% #4 not none
7207 \LWR@traceinfo{LWRprint@fcolorbox done}%
7208 }
```

\fcolorboxBlock [⟨framemode⟩] {⟨framecolor⟩} [⟨boxmode⟩] {⟨boxcolor⟩} {⟨text⟩}

In print mode, \fcolorboxBlock is the same as \fcolorbox.

```
7209 \LetLtxMacro\LWRprint\fcolorboxBlock\LWRprint\fcolorbox
```

**Env** **fcolorminipage** [⟨1:framemode⟩] {⟨framecolor⟩} [⟨3:boxmode⟩] {⟨4:boxcolor⟩} [⟨5:align⟩] [⟨6:height⟩] [⟨7:inner-align⟩] {⟨8:width⟩}

In print mode, becomes a \fcolorbox containing a minipage:

```
7210 \NewDocumentEnvironment{LWRprint@fcolorminipage}{o m o m O{c} O{} o m}
7211 {%
7212 \LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%

```

Pre-load the contents into an LR box so that they can be used inside a \fcolorbox:

```
7213 \begin{lrbox}{\LWR@colorminipagebox}%
```

If inner alignment is not given, use the outer alignment instead:

```
7214 \IfValueTF{#7}%
7215 {\begin{minipage}[#5][#6][#7]{#8}%
7216 {\begin{minipage}[#5][#6][#5]{#8}%
7217 }%
7218 {%
7219 \end{minipage}%
7220 \end{lrbox}%
7221 \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.

```
7222 \ifthenelse{\equal{#4}{none}}%
7223 {%
7224     % scope the \colorlet
7225     \colorlet{LWR@currentcolor}{.}%
7226     \color{#2}%
7227     \fbox{%
7228         \color{LWR@currentcolor}%
7229         \usebox{\LWR@colorminipagebox}%
7230     }%
7231 }%
7232 }% #4 none
```

```
7233 {#4 not none
7234     \IfValueTF{#1}%
7235     {%
7236         \IfValueTF{#3}%
7237             {\LWRorigprint@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
7238             {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
7239     }%
7240     {%" no value #1
7241     \IfValueTF{#3}%
7242         {\LWRorigprint@fcolorbox[#2]{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
7243         {\LWRorigprint@fcolorbox[#2]{#4}{\usebox{\LWR@colorminipagebox}}}%
7244     }%" no value #1
7245 }%" #4 not none
7246 \LWR@traceinfo{*** finished end fcolorminipage}%
7247 }
```

`\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`.

```
7248 \newcommand*{\LWR@restoreorigprintxcolor}{%
7249 \LWR@traceinfo{\LWR@restoreorigprintxcolor}%
7250 \LetLtxMacro\colorboxBlock\LWRprint@colorboxBlock\%
7251 \LetLtxMacro\fcolorbox\LWRprint@fcolorbox\%
7252 \LetLtxMacro\fcolorboxBlock\LWRprint@fcolorboxBlock\%
7253 \LetLtxMacro\fcolorminipage\LWRprint@fcolorminipage\%
7254 \LetLtxMacro\endfcolorminipage\endLWRprint@fcolorminipage\%
7255 }%
7256
7257 \appto{\LWR@restoreorigformatting}{%
7258 \LWR@restoreorigprintxcolor\%
7259 }
```

If print mode, immediately activate the print-mode enhancements for xcolor:

```
7260 \warpprintonly{\LWR@restoreorigprintxcolor}
7261
7262 \LWR@traceinfo{xcolor patches done}
7263 }{}% xcolor loaded
7264 }% AtBeginDocument

7265 \end{warpall}
```

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 386 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 l warp 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 l warp’s additional loading of cleveref will have no effect.

Table 9 on page 324 shows the data structure of the label/reference system as revised by l warp and cleveref.

A few patches allow cleveref to work as-is:

**for HTML output:** 7266 \begin{warpHTML}

\AtEndPreamble forces cleveref to be loaded last:

```
7267 \AtEndPreamble{
7268 \RequirePackage{cleveref}
7269 }
```

The following patches are applied after cleveref has loaded, and after \AtBeginDocument:

```
7270 \AfterEndPreamble{
7271 \LWR@traceinfo{Patching cleveref.}
```

```
\@@setcref {<kindofref>} {<label>}
7272 \renewcommand*\{\@@setcref}[2]{#1{\ref{#2}}{}{}}

\@@setcrefrange {<text>} {<label>} {<label>}
```

---

```
7273 \renewcommand{\@setcrefrange}[3]{%
7274 #1{\ref{#2}}{\ref{#3}}{}{}{}{}}
```

\cpagerefFor Redefinable word between “page(s)” and the page numbers.

```
7275 \newcommand*{\cpagerefFor}{for}
```

\@setcpageref {\<typeofref>} {\<label>}, where typeofref is “page” or “pages”

```
7276 \renewcommand*{\@setcpageref}[2]{%
7277 #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}{}{}{}}%
7278 }
```

```
7279 \renewcommand{\@setcpagerefrange}[3]{%
7280 #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}{}{}{}}%
7281 }% AfterEndPreamble
```

Remember and patch some label-related defintions. These will be further encased and patched by other packages later.

```
7282 \let\LWR@origlabel\label
7283 \let\label\LWR@newlabel
7284 \let\LWR@origref\ref
7285 \let\ref\LWR@newref% \end{ syntax highlighting
7286 \let\LWR@origpageref\pageref
7287 \let\pageref\LWR@newpageref
7288
7289
7290
7291 \end{warpHTML}
```

## 73 Picture

Env picture The picture environment is enclosed inside a \lateximage.

**for HTML output:** 7292 \begin{warpHTML}

Env picture

```
7293 \BeforeBeginEnvironment{picture}{\latexitimage}
7294
7295 \AfterEndEnvironment{picture}{\endlatexitimage}

7296 \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  $6 \times 9$  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:** 7297 `\begin{warpHTML}`

### 74.1 Counters and lengths

Ctr `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
7298 \newcounter{LWR@minipagedepth}
7299 \setcounter{LWR@minipagedepth}{0}
```

Len `\WR@minipagewidth` Used to convert the width into printable units.

---

```
7300 \newlength{\LWR@minipagewidth}
```

\len \W\minipageheight Used to convert the height into printable units.

```
7301 \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 L<sup>A</sup>T<sub>E</sub>X definition and modified to create a <div> of class mpfootnotes:

```
7302 \def\LWR@endminipage{%
7303   \par
7304   \unskip
7305   \ifvoid\@mpfootins\else
7306     \vskip\skip\@mpfootins
7307     \normalcolor
7308     \LWR@htmldivclass{mpfootnotes}
7309     \LWR@origmedskip
7310     \unvbox\@mpfootins
7311     \LWR@htmldivclassend{mpfootnotes}
7312   \fi
7313   \minipagefalse
7314   \color@endgroup
7315   \egroup
7316   \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.

```
7317 \newcommand*\LWR@subminipage{%
7318 \LWR@stopars
7319 \LWR@origminipage{6in}}
```

\raggedright cancels hyphenation, which will be done by HTML instead.

```
7320 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
7321 \LWR@startpars%
7322 }
```

\LWR@endsubminipage Closes the subminipage.

```
7323 \newcommand*{\LWR@endsubminipage}{%
7324 \LWR@stoppars%
7325 \LWR@endminipage% The following empty line is required:
7326
7327 }
```

Bool LWR@minipagefullwidth Should the next minipage have no HTML width?

```
7328 \newbool{LWR@minipagefullwidth}
7329 \boolfalse{LWR@minipagefullwidth}
```

\minipagefullwidth Requests that the next minipage have no width tag in HTML:

**for HTML output:** 7330 \newcommand\*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}
 7331 \end{warpHTML}

**for PRINT output:** 7332 \begin{warpprint}
 7333 \newcommand\*{\minipagefullwidth}{}%
 7334 \end{warpprint}

**for HTML output:** 7335 \begin{warpHTML}

Bool LWR@minipagethispar Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
7336 \newbool{LWR@minipagethispar}
7337 \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.

```
7338 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}
7339 {%
```

Units for printing dimensions to HTML:

```
7340 \uselengthunit{PT}%
```

Compute width, adjusted for frames:

```
7341 \setlength{\LWR@minipagewidth}{#4}%
7342 \ifthenelse{\cnttest{\value{\LWR@minipagedepth}}{=}{0}}{%
7343   \addtolength{\LWR@minipagewidth}{3em}%
7344   \setlength{\linewidth}{6in}%
7345   \setlength{\textwidth}{6in}%
7346   \setlength{\textheight}{9in}%
7347 }{}%
7348 \LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}
```

Compute height:

```
7349 \setlength{\LWR@minipageheight}{\textheight}%
7350 \IfValueT{#2}{\setlength{\LWR@minipageheight}{#2}}%
```

Track nesting depth:

```
7351 \addtocounter{\LWR@minipagedepth}{1}%
```

$\text{\LaTeX}$  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.

```
7352 \ifbool{FormatWP}{\newline}{}%
7353 \LWR@stoppars%
```

If FormatWP, add a text frame:

```
7354 \ifbool{FormatWP}{%
7355   \addtocounter{\LWR@thisfloat}{1}%
7356   \booltrue{\LWR@freezethisfloat}%
7357   \LWR@htmlltag{div id="autofloat-\arabic{\LWR@thisfloat}" class="wpminipage"}%
7359 }{}%
```

Create the <div> tag with optional alignment style:

```
7361 \LWR@traceinfo{minipage: creating div class}%
7362 \LWR@orignewpage%
7363 \LWR@htmlltag{div class="minipage" style="%"%
7364 \ifthenelse{\equal{#1}{t}}{vertical-align:bottom ; }{}%
7365 \ifthenelse{\equal{#1}{c}}{vertical-align:middle ; }{}}
```

---

```

7366 \ifthenelse{\equal{#1}{b}}{vertical-align:top ; }{}%
7367 \ifthenelse{\equal{#3}{t}}{justify-content:flex-start ; }{}%
7368 \ifthenelse{\equal{#3}{c}}{justify-content:center ; }{}%
7369 \ifthenelse{\equal{#3}{b}}{justify-content:flex-end ; }{}%
7370 \ifthenelse{\equal{#3}{s}}{justify-content:space-between ; }{}%

```

Print the width and optional height styles:

```

7371 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
7372 \uselengthunit{PT}%
7373 \ifbool{\LWR@minipagefullwidth}%
7374 {\boolfalse{\LWR@minipagefullwidth}}%
7375 {}%
7376 \ifthenelse{\lengthtest{#4}=\linewidth}%
7377 {}%
7378 {width:\rndprintlength{\LWR@minipagewidth} ; }%
7379 }%
7380 \LWR@traceinfo{minipage: about to print the height}%
7381 \IfValueT{#2}{height:\rndprintlength{\LWR@minipageheight} ; }%
7382 "{}"%

```

Finish with an empty line to start L<sup>A</sup>T<sub>E</sub>X minipage processing on a new line. Use a large minipage area to avoid the unnecessary wrapping of tags.

```

7383
7384 \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.

```

7385 \setlength{\linewidth}{#4}%
7386 \setlength{\textwidth}{6in}%
7387 \setlength{\textheight}{9in}%

```

\raggedright cancels hyphenation, which will be done by HTML instead.

```

7388 \LWR@origraggedright%

```

Resume paragraph tag handling for the contents of the minipage:

```

7389 \LWR@startpars%
7390 \ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%
7391
7392 === begin minipage ===
7393
7394 }{}%

```

```
7395 \LWR@traceinfo{minipage: finished starting the minipage}%
7396 }
```

End the environment with  $\text{\TeX}$  processing and closing tag:

```
7397 {%
7398 \ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%
7399
7400 === end minipage ===
7401
7402 }{}%
7403 \LWR@stoppars%
7404 \LWR@endminipage% The following empty line is required:
7405
7406 \ifbool{FormatWP}{%
7407
7408 \LWR@htmlelementend{div}%
7409 \boolfalse{LWR@freezethisfloat}%
7410
7411 }{}%
7412 \LWR@htmldivclassend{minipage}%
7413
7414 \LWR@origvspace{1\baselineskip}% required for subcaption
7415 \addtocounter{LWR@minipagedepth}{-1}%
7416 \LWR@startpars%
7417 \ifbool{FormatWP}{\newline}{}
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
7418 \global\booltrue{LWR@minipagethispar}%
7419 }
```

## 74.4 Parbox, makebox, framebox, fbox, raisebox

**for HTML output:**  $\backslash$ parbox [⟨pos⟩] [⟨height⟩] [⟨inner-pos⟩] {⟨width⟩} {⟨text⟩}

A parbox uses the minipage code:

```
7420 \RenewDocumentCommand{\parbox}{O{t} O{t} m +m}
7421 {
7422 \LWR@traceinfo{parbox of width #4}%
7423 \begin{minipage}[#1][#2][#3]{#4}
7424 #5
7425 \end{minipage}
7426 }
```

---

```
\makebox  (<())posn) [<width>] [<pos>] {<text>}

7427 \LetLtxMacro{\LWR@origmakebox}{\makebox}
7428
7429 \RenewDocumentCommand{\makebox}{d() o o m}{%
```

Check for the optional width:

```
7430 \IfValueTF{#2}{%
7431 {%
```

Check for the horizontal text alignment. For stretched, the best HTML can do is justified alignment.

```
7432      {%
7433      \def\LWR@align{center}%
7434      \ifstrequal{#3}{l}{\def\LWR@align{left}}{}%
7435      \ifstrequal{#3}{r}{\def\LWR@align{right}}{}%
7436      \ifstrequal{#3}{s}{\def\LWR@align{justify}}{}%
```

To print the width argument:

```
7437      \setlength{\LWR@tempwidth}{#2}{%
```

inline-block allows width and text-align to be used in a <span>.

```
7438      \uselengthunit{PT}%
7439      \InlineClass[%
7440          display:inline-block ; %
7441          text-align:\LWR@align\ ; %
7442          width:\rndprintlength{\LWR@tempwidth}%
7443      ]%
7444      {makebox}%
```

Without a width argument, the text is simply used inline:

```
7445      {\mbox{#4}}%
7446      }% scope
7447 }%
7448 {\mbox{#4}}%
7449 }
```

```
\framebox  [<width>] [<pos>] {<text>}

7450 \LetLtxMacro{\LWR@origframebox}{\framebox}
7451
7452 \RenewDocumentCommand{\framebox}{o o m}{%
```

---

```
7453 \fbox{\makebox[#1][#2]{#3}}%
7454 }
```

\LWR@forceminwidth {*length*}

Sets \LWR@atleastonept to be at least 1pt.

```
7455 \newlength{\LWR@atleastonept}
7456
7457 \newcommand*{\LWR@forceminwidth}[1]{%
7458   \setlength{\LWR@atleastonept}{#1}%
7459   \ifthenelse{%
7460     \lengthtest{\LWR@atleastonept>0pt}\AND%
7461     \lengthtest{\LWR@atleastonept<1pt}%
7462   }{%
7463     \setlength{\LWR@atleastonept}{1pt}%
7464   }%
7465 }
```

\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.

```
7466 \newcommand*{\LWR@blackborderpadding}{%
7467   \uselengthunit{PT}%
7468   border:\rndprintlength{\LWR@atleastonept} solid black ; %
7469   padding:\rndprintlength{\fboxsep}%
7470 }
```

\fbox {*text*}

Creates a framed inline span enclosing the text.

Remember the print-mode version:

```
7471 \let\LWRprint\fbox\fbox
```

Create a new HTML version, but don't use it until after xcolor may have loaded:

```
7472 \newcommand{\LWRhtml\fbox}[1]{%
7473   \LWR@traceinfo{HTML \fbox}%
7474   \LWR@forceminwidth{\fboxrule}%
7475   \InlineClass[%
7476     \LWR@blackborderpadding%
7477   ]{\fbox}{#1}%
7478 }
```

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:

```
7479 \AtBeginDocument{\let\fbox\LWRhtml\fbox}
```

\fboxBlock {<text>} Creates a framed HTML <div> of the text.

A print-output version is also supplied below.

```
7480 \newcommand{\fboxBlock}[1]{%
7481 \LWR@forceminwidth{\fboxrule}%
7482 \begin{BlockClass}[%%
7483 \LWR@blackborderpadding%
7484 ]{\fboxBlock}%
7485 #1%
7486 \end{BlockClass}%
7487 }
```

Env fminipage [<align>] [<height>] [<align>] [<width>]

Creates a framed HTML <div> around its contents.

A print-output version is also supplied below.

```
7488 \NewDocumentEnvironment{fminipage}{O{t} o O{t} m}%
7489 {%
7490 \LWR@traceinfo{fminipage #1 #2 #3 #4}%
7491 \LWR@forceminwidth{\fboxrule}%
7492 \setlength{\LWR@tempwidth}{#4}%
7493 \IfValueT{#2}{\setlength{\LWR@tempheight}{#2}}%
7494 \begin{BlockClass}[%%
7495 \LWR@blackborderpadding ; %
7496 \uselengthunit{PT}%
7497 \IfValueT{#2}{height:\rndprintlength{\LWR@tempheight} ; }%
7498 width:\rndprintlength{\LWR@tempwidth}%
7499 ]{\fminipage}%
7500 }%
7501 {%
7502 \end{BlockClass}%
7503 \LWR@traceinfo{fminipage done}%
7504 }
```

\raisebox {<raiselen>} [<height>] [<depth>] {<text>}

```
7505 \LetLtxMacro{\LWR@origraisebox}{\raisebox}%
7506 %
7507 \RenewDocumentCommand{\raisebox}{m o o m}{%
```

```

7508 #4%
7509 }

7510 \end{warpHTML}
```

**for HTML & PRINT:** 7511 \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 [⟨1:align⟩] [⟨2:height⟩] [⟨3:inner-align⟩] {⟨4:width⟩}

Creates a frame around its contents.

```

7512 \newsavebox{\LWR@fminipagebox}
7513
7514 \NewDocumentEnvironment{LWRprint@fminipage}{O{t} o O{t} m}
7515 {%
```

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:

```

7516 \IfValueTF{#3}%
7517 {\def\LWR@thisalign{#3}}
7518 {\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.

```

7519 \IfValueTF{#2}%
7520 {\minipage[#1][#2+2\fboxsep+2\fboxrule][\LWR@thisalign]{#4+2\fboxsep+2\fboxrule}}%
7521 {\minipage[#1]{#4+2\fboxsep+2\fboxrule}}%
```

Capture the contents of the environment:

```
7522 \begin{lrbox}{\LWR@fminipagebox}%
```

Nest the contents inside an inner minipage of the desired size:

```

7523 \IfValueTF{#2}%
7524 {\minipage[#1][#2][\LWR@thisalign]{#4}}%
7525 {\minipage[#1]{#4}}%
7526 }
7527 {%
```

Close the inner minipage and the LR box with the contents:

```
7528 \end{minipage}%
7529 \end{lrbox}%
```

Create a frame around the contents of the environment:

```
7530 \fbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```
7531 \end{minipage}%
7532 }
```

```
7533 \end{warpall}
```

**for PRINT output:** 7534 \begin{warpprint}

For print output, the following are \let to become active.

```
\fboxBlock {\langle text\rangle}
```

Creates a framed HTML <div> around the text.

```
7535 \let\fboxBlock\fbox
```

```
Env fminipage [\langle align\rangle] [\langle height\rangle] [\langle align\rangle] {\langle width\rangle}
```

Creates a frame around its contents.

```
7536 \LetLtxMacro{\fminipage}{\LWRprint@fminipage}%
7537 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}
```

```
7538 \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:** 7539 \begin{warpHTML}

\LWR@HTMLtextstyle {\langle FormatWP style\rangle} {\langle class\rangle} {\langle text\rangle}

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.

```
7540 \DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{%
7541 \ifbool{FormatWP}{%
7542 {\LWR@htmlspanclass[\#1]{\#2}{\#3}}{%
7543 {\LWR@htmlspanclass[\#2]{\#3}}{%
7544 }}
```

\emph {\langle text\rangle}

```
7545 \DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{\#1}}
7546 \DeclareRobustCommand{\LWR@nullemph}[1]{\#1}
7547 \LetLtxMacro{\emph}{\LWR@HTMLemph}
```

\textmd {\langle text\rangle}

```
7548 \DeclareRobustCommand{\LWR@HTMLtextmd}[1]{%
7549 \LWR@HTMLtextstyle{font-weight: normal}{textmd}{\#1}{%
7550 }
7551 \DeclareRobustCommand{\LWR@nulltextmd}[1]{\#1}
7552
7553 \LetLtxMacro{\textmd}{\LWR@HTMLtextmd}
```

\textbf {\langle text\rangle}

```
7554 \DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{\#1}}
7555 \DeclareRobustCommand{\LWR@nulltextbf}[1]{\#1}
7556 \LetLtxMacro{\textbf}{\LWR@HTMLtextbf}
```

\textrm {\langle text\rangle}

```
7557 \DeclareRobustCommand{\LWR@HTMLtextrm}[1]{%
7558 \LWR@HTMLtextstyle{font-family: serif}{textrm}{\#1}{%
7559 }
7560
7561 \DeclareRobustCommand{\LWR@nulltextrm}[1]{\#1}
7562
7563 \LetLtxMacro{\textrm}{\LWR@HTMLtextrm}
```

```
\textsf  {<text>}

7564 \DeclareRobustCommand{\LWR@HTMLtextsf}[1]{%
7565 \LWR@HTMLtextstyle{font-family:sans}{textsf}{#1}%
7566 }
7567 \DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
7568 \LetLtxMacro{\textsf}{\LWR@HTMLtextsf}

\texttt  {<text>}

7569 \DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
7570 \DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
7571 \LetLtxMacro{\texttt}{\LWR@HTMLtexttt}

\textup  {<text>}

7572 \DeclareRobustCommand{\LWR@HTMLtextup}[1]{%
7573 \LWR@HTMLtextstyle{font-variant:normal}{textup}{#1}%
7574 }
7575
7576 \DeclareRobustCommand{\LWR@nulltextup}[1]{#1}
7577
7578 \LetLtxMacro{\textup}{\LWR@HTMLtextup}

\textit  {<text>}

7579 \DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}}
7580 \DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
7581 \LetLtxMacro{\textit}{\LWR@HTMLtextit}

\textsc  {<text>}

7582 \DeclareRobustCommand{\LWR@HTMLtextsc}[1]{%
7583 \LWR@HTMLtextstyle{font-variant:small-caps}{textsc}{#1}%
7584 }
7585
7586 \DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}
7587
7588 \LetLtxMacro{\textsc}{\LWR@HTMLtextsc}

\textsl  {<text>}

7589 \DeclareRobustCommand{\LWR@HTMLtextsl}[1]{%
7590 \LWR@HTMLtextstyle{font-style:oblique}{textsl}{#1}%
7591 }
7592
```

```
7593 \DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}
7594
7595 \LetLtxMacro{\textsl}{\LWR@HTMLtextsl}

\textnormal {<text>}

7596 \DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
7597 \DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
7598 \LetLtxMacro{\textnormal}{\LWR@HTMLtextnormal}

7599 \DeclareRobustCommand{\LWR@nullrmfamily}{}
7600 \DeclareRobustCommand{\LWR@nullsffamily}{}
7601 \DeclareRobustCommand{\LWR@nullttfamily}{}
7602 \DeclareRobustCommand{\LWR@nullbfseries}{}
7603 \DeclareRobustCommand{\LWR@nullmdseries}{}
7604 \DeclareRobustCommand{\LWR@nullupshape}{}
7605 \DeclareRobustCommand{\LWR@nullslshape}{}
7606 \DeclareRobustCommand{\LWR@nullscshape}{}
7607 \DeclareRobustCommand{\LWR@nullitshape}{}
7608 \DeclareRobustCommand{\LWR@nullem}[1]{}
7609 \DeclareRobustCommand{\LWR@nullnormalfont}{}
```

\LWR@nullfonts Removes formatting during filename operations.

```
7610 \newcommand*{\LWR@nullfonts}{%
7611 \LetLtxMacro{\emph}{\LWR@nullemph}%
7612 \LetLtxMacro{\textmd}{\LWR@nulltextmd}%
7613 \LetLtxMacro{\textbf}{\LWR@nulltextbf}%
7614 \LetLtxMacro{\textrm}{\LWR@nulltextrm}%
7615 \LetLtxMacro{\textsf}{\LWR@nulltextsf}%
7616 \LetLtxMacro{\texttt}{\LWR@nulltexttt}%
7617 \LetLtxMacro{\textup}{\LWR@nulltextup}%
7618 \LetLtxMacro{\textit}{\LWR@nulltextit}%
7619 \LetLtxMacro{\textsc}{\LWR@nulltextsc}%
7620 \LetLtxMacro{\textsl}{\LWR@nulltextsl}%
7621 \LetLtxMacro{\textnormal}{\LWR@nulltextnormal}%
7622 \LetLtxMacro{\rmfamily}{\LWR@nullrmfamily}%
7623 \LetLtxMacro{\sffamily}{\LWR@nullsffamily}%
7624 \LetLtxMacro{\ttfamily}{\LWR@nullttfamily}%
7625 \LetLtxMacro{\bfseries}{\LWR@nullbfseries}%
7626 \LetLtxMacro{\mdseries}{\LWR@nullmdseries}%
7627 \LetLtxMacro{\upshape}{\LWR@nullupshape}%
7628 \LetLtxMacro{\slshape}{\LWR@nullslshape}%
7629 \LetLtxMacro{\scshape}{\LWR@nullscshape}%
7630 \LetLtxMacro{\itshape}{\LWR@nullitshape}%
7631 \LetLtxMacro{\em}{\LWR@nullem}%
7632 \LetLtxMacro{\normalfont}{\LWR@nullnormalfont}%
7633 \renewcommand*{\HTMLunicode}[1]{}%
```

```
7634 \renewcommand*{\HTMLentity}[1] {}%
```

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

```
7635 \renewcommand*{\&}{and}%
7636 \renewcommand{\textsuperscript}[1]{##1}%
7637 \renewcommand{\textsubscript}[1]{##1}%
7638 \LetLtxMacro{\underline}{\LWR@origunderline}%
7639 \RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
7640 \DeclareExpandableDocumentCommand{\InlineClass}{+o +m +m}{##3}%
7641 \DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{##3}%
7642 }
```

\mdseries

```
7643 \renewcommand*{\mdseries}{}%
```

\bfseries

```
7644 \renewcommand*{\bfseries}{}%
```

\rmfamily

```
7645 \renewcommand*{\rmfamily}{}%
```

\sffamily

```
7646 \renewcommand*{\sffamily}{}%
```

\ttfamily

```
7647 \renewcommand*{\ttfamily}{}%
```

\upshape

```
7648 \renewcommand*{\upshape}{}%
```

\itshape

```
7649 \renewcommand*{\itshape}{}%
```

\scshape

```
7650 \renewcommand*{\scshape}{}%
```

```
\normalfont
```

```
7651 \renewcommand*{\normalfont}{}{}
```

```
\sp {<text>}
```

For siunitx. Must work in math mode.

```
7652 \renewcommand{\sp}[1]{\text{\textsuperscript{#1}}}{}
```

```
\sb {<text>}
```

For siunitx. Must work in math mode.

```
7653 \renewcommand{\sb}[1]{\text{\textsubscript{#1}}}{}
```

```
\textsuperscript {<text>}
```

```
7654 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}{}
```

```
\@textsuperscript {<text>}
```

```
7655 \renewcommand{\@textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}{}
```

```
\textsubscript {<text>}
```

```
7656 \AtBeginDocument{
```

```
7657 \renewcommand{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}}{}
```

```
7658 }
```

```
\@textsubscript {<text>}
```

```
7659 \AtBeginDocument{
```

```
7660 \renewcommand{\@textsubscript}[1]{\LWR@htmlspan{sub}{#1}}{}
```

```
7661 }
```

```
\up {<text>} Prints superscript.
```

This is \let at the beginning of the document in case some other package has changed the definition.

```
7662 \AtBeginDocument{\let\up\textsuperscript}
```

```
\fup {<text>} Prints superscript.
```

Supports fmtcount package.

This is \let at the beginning of the document in case some other package has changed the definition.

```
7663 \AtBeginDocument{\let\fup\textsuperscript}

\underline {\langle text\rangle}

7664 \renewcommand{\underline}[1]{%
7665 \LWR@HTMLtextstyle{%
7666 {text-decoration:underline;text-decoration-skip}{%
7667 {underline}{#1}}%
7668 }}

\hfill

7669 \renewcommand*{\hfill}{\qquad}

\hrulefill

7670 \renewcommand*{\hrulefill}{\rule{1in}{1pt}>

\dotfill

7671 \renewcommand*{\dotfill}{\dots}

7672 \end{warpHTML}
```

## 76 Skips, spaces, font sizes

**for HTML output:** 7673 \begin{warpHTML}

\, must be redefined after \RequirePackage{printlen}

```
7674 \let\LWR@origcomma\,
7675 \let\LWR@origtilde\-
7676 \let\LWR@origskip\enskip
7677 \let\LWR@origquad\quad
7678 \let\LWR@origqquad\qquad
7679 \let\LWR@orighspace\hspace
7680 \let\LWR@origvspace\vspace
7681 \let\LWR@origrule\rule
```

```
7682 \let\LWR@origmedskip\medskip
7683 \let\LWR@origtextellipsis\textellipsis
```

Direct-formatting space commands become HTML entities:

```
7684 \renewcommand*{\,}{\HTMLunicode{202f}} % HTML thin non-breakable space
7685
7686 \renewcommand*{-}{\HTMLentity{nbsp}}
7687
7688 \renewcommand*{\textellipsis}{\HTMLunicode{2026}}
```

Direct-formatting font sizes are ignored:

```
7689 \let\LWR@orignormalsize\normalsize
7690 \let\LWR@origsmall\small
7691 \let\LWR@origfootnotesize\footnotesize
7692 \let\LWR@origscriptsize\scriptsize
7693 \let\LWR@origtiny\tiny
7694 \let\LWR@origlarge\large
7695 \let\LWR@origLarge\Large
7696 \let\LWR@origLARGE\LARGE
7697 \let\LWR@orighuge\huge
7698 \let\LWR@origHuge\Huge
7699 \renewcommand*{\normalsize} {}
7700 \renewcommand*{\small} {}
7701 \renewcommand*{\footnotesize} {}
7702 \renewcommand*{\scriptsize} {}
7703 \renewcommand*{\tiny} {}
7704 \renewcommand*{\large} {}
7705 \renewcommand*{\Large} {}
7706 \renewcommand*{\LARGE} {}
7707 \renewcommand*{\huge} {}
7708 \renewcommand*{\Huge} {}
7709
7710 \renewcommand*{\onecolumn} {}
7711
7712 \renewcommand{\twocolumn}[1] []
7713
7714 #1
7715
7716 }
```

\newline Uses the HTML <br /> element.

```
7717 \newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
7718 \let\newline\LWR@newlinebr
```

\` Redefined to \LWR@endofline or \LWR@tabularendofline.

\LWR@endofline \* [<len>]

\` is assigned to \LWR@endofline at \LWR@LwarpStart.

Inside tabular, \` is temporarily changed to \LWR@tabularendofline.

```
7719 \LetLtxMacro{\LWR@origendofline}\`  
7720 \NewDocumentCommand{\LWR@endofline}{s o}  
7721 {  
7722 \newline%  
7723 }
```

- |  |   |
|--|---|
| <p>\LWR@minipagestartpars</p> <p>\hspace</p> <p>\enskip</p> <p>\quad</p> <p>\qquad</p> | <p>Minipages are often placed side-by-side inside figures, with a bit of horizontal space to separate them. Since HTML does not allow a &lt;div&gt; to be inside a p, paragraphs must be turned off during the generation of the minipage, then turned on after the minipage is complete. When this occurs between side-by-side minipages, lwarp correctly suppresses the paragraph tags between the minipages, unless some other text is between the minipages. Such text forms its own paragraph, resulting in text after a minipage to be on its own line. Since people often place small horizontal space between minipages, it is desirable to maintain this space if possible. lwarp tries to do this by remembering that a minipage has been seen, in which case paragraph tags are suppressed around \hspace, \enskip, \quad, and \qquad until the end of the paragraph, when the closing p tag is created.</p> |
|--|---|

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.

```
7724 \newcommand*{\LWR@minipagestartpars}{%  
7725 \ifbool{\LWR@minipagethispar}{\LWR@startpars}{}}%  
7726 }
```

- |                              |   |
|------------------------------|---|
| <p>\LWR@minipagestoppars</p> | <p>Placed just after \hspace, \quad, or \qquad's HTML output.</p> |
|------------------------------|---|

```
7727 \newcommand*{\LWR@minipagestoppars}{%  
7728 \ifbool{\LWR@minipagethispar}{\LWR@stoppars}{}}%  
7729 }
```

- |              |  |
|--------------|--|
| <p>\quad</p> | <p>Handles special minipage &amp; horizontal space interactions.</p> |
|--------------|--|

```
7730 \renewcommand*{\quad}{%
```

---

```

7731 \LWR@minipagestoppars%
7732 \HTMLunicode{2001}%
7733 \LWR@minipagestartpars%
7734 }

```

\qquad Handles special minipage & horizontal space interactions.

```
7735 \renewcommand*{\qquad}{\quad\quad}
```

\enskip Handles special minipage & horizontal space interactions.

```

7736 \renewcommand*{\enskip}{%
7737 \LWR@minipagestoppars%
7738 \HTMLunicode{2000}%
7739 \LWR@minipagestartpars%
7740 }

```

Len \WR@tempwidth Used to compute span width, height, raise for \hspace and \rule:

```

Len \WR@tempheight 7741 \newlength{\LWR@tempwidth}
Len \WR@tempraise 7742 \newlength{\LWR@tempheight}
7743 \newlength{\LWR@tempraise}

```

\LWR@hspace \* {*<length>*}

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.

```

7744 \NewDocumentCommand{\LWR@hspace}{s m}{%
7745 \setlength{\LWR@tempwidth}{#2}%

```

If \fill, change to \qquad:

```

7746 \ifnum\gluestretchorder\LWR@tempwidth>0%
7747 \setlength{\LWR@tempwidth}{2em}%
7748 \fi%

```

Only if the width is not zero:

```
7749 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{0pt}}{}{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7750     \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
7751     \ifthenelse{\dimtest{\LWR@tempwidth}{=}{.16667em}}%
7752     {%
7753         \HTMLunicode{2009}% thin breakable space
7754     }%
```

Print the span with the converted width. Not rounded.

```
7755     {%
7756         \uselengthunit{PT}%
7757         \LWR@htmltagc{%
7758             span style="width:\printlength{\LWR@tempwidth}; %
7759                 display:inline-block"%
7760         }%
7761     }%
```

If formatting for a word processor, approximate with a number of \quads, in case a span of a given width is not supported:

```
7761     \ifbool{FormatWP}{%
7762         \setlength{\LWR@templengthone}{\LWR@tempwidth}%
7763         \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
7764             \quad%
7765             \addtolength{\LWR@templengthone}{-1em}%
7766         }%
7767     }{}
```

Close the span:

```
7768     \LWR@htmltagc{/span}%
7769 }
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7770     \LWR@minipagestartpars%
7771 }% width not 0
7772 }
```

```
\LWR@nohspace * {\langle length\rangle}
```

Used to disable \hspace while creating description \items.

```
7773 \NewDocumentCommand{\LWR@nohspace}{s m}{}
```

```
\hspace * {\langle length\rangle}
```

Handles special minipage & horizontal space interactions.

7774 \LetLtxMacro{\hspace}{\LWR@hspace}

\LWR@vspace \* {*length*} Nullified vspace.

7775 \NewDocumentCommand{\LWR@vspace}{s m}{}{}

\vspace \* {*length*} Nullified.

7776 \let\vspace\LWR@vspace

\linebreak [*num*] Inserts an HTML br tag.

7777 \renewcommand\*\linebreak[1][]{\newline}

\nolinebreak [*num*]

7778 \renewcommand\*\nolinebreak[1][]{}

\pagebreak [*num*] Starts a new paragraph.

7779 \renewcommand\*\pagebreak[1][]{}

7780

7781 }

\nopagebreak [*num*]

7782 \renewcommand\*\nopagebreak[1][]{}

\enlarge this page \* {*len*}

7783 \RenewDocumentCommand{\enlarge this page}{s m}{}{}

\LWR@currenttextcolor The color to use for text and \rule, defaulting to black:

7784 \newcommand\*\LWR@currenttextcolor{black}

\LWR@rule [*raise*] {*width*} {*height*}

Handles special minipage & horizontal space interactions.

Creates a span of a given width and height. Ignores the optional star.

\fill is zero-width, so \hspace{\fill} is ignored.

```
7785 \NewDocumentCommand{\LWR@rule}{o m m}{%
```

The width is copied into a temporary  $\text{\TeX}$  length, from which comparisons and conversions may be made:

```
7786 \setlength{\LWR@tempwidth}{#2}%
```

If it's zero-width then skip the entire rule:

```
7787 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}
7788 {}% zero- width
7789 {}% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
7790 \ifthenelse{%
7791   \lengthtest{\LWR@tempwidth>0pt}\AND%
7792   \lengthtest{\LWR@tempwidth<1pt}}%
7793 {}%
7794 {\setlength{\LWR@tempwidth}{1pt}}{}%
```

Likewise with height:

```
7795 \setlength{\LWR@tempheight}{#3}%
7796 \ifthenelse{%
7797   \lengthtest{\LWR@tempheight>0pt}\AND%
7798   \lengthtest{\LWR@tempheight<1pt}}%
7799 {}%
7800 {\setlength{\LWR@tempheight}{1pt}}{}%
```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
7801 \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  $\text{\TeX}$  code.

```
7802 \uselengthunit{PT}%
7803 \LWR@htmltagc{%
7804 span
7805 style=""}
```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```
7806     \ifbool{FormatWP}{}{\background:\LWR@currenttextcolor; }%
```

The width and height are printed, converted to PT:

```
7807     width:\printlength{\LWR@tempwidth}; %
7808     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  $\text{TeX}$  length allows a typical  $\text{TeX}$  expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a  $\text{TeX}$  length limits the allowable syntax. To do: A superior method would compute a ratio of  $\text{TeX}$  ex height, then print that to HTML with an ex unit.

```
7809     \IfValueT{#1}{%
7810     {%
7811         \setlength{\LWR@tempraise}{0pt-#1}%
7812         \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
7813         \LWR@orignewline%
7814         -ms-transform: translate(0pt,\printlength{\LWR@tempraise}); %
7815         \LWR@orignewline%
7816         -webkit-transform: translate(0pt,\printlength{\LWR@tempraise}); %
7817         \LWR@orignewline%
7818         transform: translate(0pt,\printlength{\LWR@tempraise}); %
7819         \LWR@orignewline%
7820     }%
7821 }
```

Display inline-block to place the span inline with the text:

```
7821     display:inline-block;"%
7822 }
```

If formatting for a word processor, approximate with a number of underscores, in case a span of a given width is not supported:

```
7823     \ifbool{FormatWP}{%
7824         \setlength{\LWR@tempLengthOne}{\LWR@tempwidth}%
7825         \whiledo{\lengthtest{\LWR@tempLengthOne>1em}}{%
7826             \_{}%
7827             \addtolength{\LWR@tempLengthOne}{-1em}%
7828         }%
7829     }{}%
```

Close the span:

---

```
7830     \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7831     \LWR@minipagestartpars%
7832 }% non-zero width
7833 }
```

```
\rule [⟨raise⟩] {⟨width⟩} {⟨height⟩}
```

Handles special minipage & horizontal space interactions.

```
7834 \renewcommand{\rule}{\LWR@rule}
```

```
7835 \end{warpHTML}
```

## 77 \phantomsection

**for HTML output:** 7836 \begin{warpHTML}

\phantomsection Emulate the hyperref \phantomsection command, often used to insert the bibliography into table of contents:

```
7837 \newcommand*{\phantomsection}{%
7838 \section*{}%
7839 }
```

```
7840 \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:** 7841 \begin{warpHTML}

\TeX \TeX

latexlogo is a css class used to properly typeset the E and A in \TeX and friends.

latexlogofont is a css class used to select the font for the rest of the logo in \TeX, LuaTeX, ConTeXt, etc.

```
7842 \let\LWR@origTeX\TeX
7843
7844 \newcommand*\{\LWR@TeX}
7845 {%
7846     \InlineClass{latexlogofont}%
7847     {%
7848         \LWR@HTMLtextstyle%
7849         {text-transform:uppercase}%
7850         {latexlogo}%
7851         {T\textsubscript{e}X}%
7852     }%
7853 }
```

\LaTeX \TeX, \TeX2\varepsilon

\LaTeXe

```
7854 \let\LWR@origLaTeX\LaTeX
7855
7856 \newcommand*\{\LWR@LaTeX}
7857 {%
7858     \InlineClass{latexlogofont}%
7859     {%
7860         \LWR@HTMLtextstyle%
7861         {text-transform:uppercase}%
7862         {latexlogo}%
7863         {L\textsuperscript{a}T\textsubscript{e}X}%
7864     }%
7865 }
7866
7867 \let\LWR@origLaTeXe\LaTeXe
7868
7869 \renewcommand*\{\LaTeXe}
7870 {\LaTeX\InlineClass{latexlogofont}%
7871 {\textit{\HTMLunicode{3B5}}}}
```

\LuaTeX \LuaTeX, \LuaTeX

\LuaLaTeX

---

```
7872 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
7873 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

\XeTeX X<sub>E</sub>T<sub>E</sub>X, X<sub>E</sub>LT<sub>E</sub>X

\XeLaTeX xetexlogo is a css class which aligns the backwards E in X<sub>E</sub>T<sub>E</sub>X and spaces T<sub>E</sub>X appropriately.

xelatexlogo is a css class which aligns the backwards E in X<sub>E</sub>LT<sub>E</sub>X and spaces LT<sub>E</sub>X appropriately.

```
7874 \newcommand*{\Xe}
7875 {X\textsubscript{\HTMLunicode{18e}}}
7876 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
7877 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}
```

\ConTeXt ConTeXt

```
7878 \newcommand*{\LWR@ConTeXt}
7879 {\InlineClass{latexlogofont}{Con}\TeX{}%}
7880 \InlineClass{latexlogofont}{t}}
```

\BibTeX BiB<sub>T</sub>E<sub>X</sub>, *MakeIndex*

\MakeIndex

```
7881 \providecommand*{\BibTeX}
7882 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}
7883
7884 \newcommand*{\MakeIndex}
7885 {\InlineClass{latexlogofont}{\textit{MakeIndex}}}
```

\AmS A<sub>M</sub>S

amslogo is a css class used for the A<sub>M</sub>S logo.

```
7886 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}%
7887 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}
```

\MiKTeX MiK<sub>T</sub>E<sub>X</sub>

```
7888 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}
```

\LyX LyX

lyxlogo is a css class used for the LyX logo.

```
7889 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
```

---

7890 \end{warpHTML}

## 78.2 Print logos

**for PRINT output:**

```

7891 \begin{warpprint}
7892 \newcommand*\{\XeTeXrevE\}
7893 {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
7894 \providecommand*\{\XeTeX\}{\mbox{X}\XeTeXrevE\TeX\}
7895 \providecommand*\{\XeLaTeX\}{\mbox{X}\XeTeXrevE\LaTeX\}
7896 \providecommand*\{\AmS\}{%
7897 \leavevmode\hbox{$\mathcal A$\kern-.2em\lower.376ex\%}
7898 \hbox{$\mathcal M$\kern-.2em$\mathcal S$}}
7899 \newcommand*\{\LyX\}{\textsf{LyX\!}}
7900 \providecommand*\{\LuaTeX\}{\mbox{Lua\TeX\!}}
7901 \providecommand*\{\LuaLaTeX\}{\mbox{Lua\LaTeX\!}}
7902 \providecommand*\{\BibTeX\}{\mbox{B\textsc{ib}\TeX\!}}
7903 \providecommand*\{\MakeIndex\}{\mbox{\textit{MakeIndex}}\!}
7904 \providecommand*\{\ConTeXt\}{\mbox{Con\TeXt\!}}
7905 \providecommand*\{\MiKTeX\}{\mbox{MiK\TeX\!}}
7906 \end{warpprint}

```

## 79 \AtBeginDocument, \AtEndDocument

**for HTML output:** 7907 \begin{warpHTML}

\LWR@LwarpStart Automatically sets up the HTML-related actions for the start and end of the document.  
 \LWR@LwarpEnd

```

7908 \AfterEndPreamble{\LWR@LwarpStart}
7909 \AtEndDocument{\LWR@LwarpEnd}

7910 \end{warpHTML}

```

## 80 Trademarks

- TeX is a trademark of American Mathematical Society.
- Adobe® and Adobe Framemaker® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.
- Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Mac OS® is a trademark of Apple Inc.
- MadCap Flare™ is the property of MadCap Software, Inc.
- MathJax is copyright 2009 and later. The MathJax Consortium is a joint venture of the American Mathematical Society (AMS) and the Society for Industrial and Applied Mathematics (SIAM) to advance mathematical and scientific content on the web.
- Microsoft®, Encarta, MSN, and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- UNIX® is a registered trademark of The Open Group.

The following adjustments apply to the l warp-\* package listings:

---

File 2 **lwarf-a4.sty**

§ 81 Package **a4**

Pkg a4 a4 is emulated.

**for HTML output:** 1 \LWR@ProvidesPackageDrop{a4}

2 \newcommand\*{\WideMargins}{}

---

File 3 **lwarf-a4wide.sty**

§ 82 Package **a4wide**

Pkg a4wide a4wide is emulated.

**for HTML output:** 1 \LWR@ProvidesPackageDrop{a4wide}

---

File 4 **lwarf-a5comb.sty**

§ 83 Package **a5comb**

Pkg a5comb a5comb is emulated.

**for HTML output:** 1 \LWR@ProvidesPackageDrop{a5comb}

---

File 5 **lwarf-abstract.sty**

§ 84 Package **abstract**

(Based on original code by PETER WILSON.)

Pkg abstract abstract is supported and patched by lwarf.

 **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 \warp-abstract:

```
1 \LWR@ProvidesPackagePass{abstract}

2 \AtBeginDocument{
3 \BeforeBeginEnvironment{abstract}{
4 \LWR@forcenewpage
5 \BlockClass{abstract}
6 }
7 \AfterEndEnvironment{abstract}{\endBlockClass}
8 }
9
10 \renewcommand{\@bsrunintitle}{%
11 \hspace*{\abstitleskip}%
12 {\abstractnamefont{%
13 \InlineClass{abstractrunintitle}{\abstractname}%
14 \@bslabeldelim}%
15 }%
16
17 \if@titlepage
18 \ renewenvironment{abstract}{%
19 % \titlepage
20 \null\vfil
21 \begin{parpenalty}\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 \end{parpenalty}\OM
33 \end{absnamepos}%
34 %% \vspace{\abstitleskip}%
35 \fi
36 \fi
37 \vspace{\abstitleskip}%
38 \fi
39 \put@bsintoc%
40 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
41 {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
42 }
43 \else
44 \renewenvironment{abstract}{%
```

---

```

45      \if@bsrunin
46      \else
47          \if@bsstyle
48              \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
49          \else
50              \ifnumber@bs
51                  \num@bs
52              \else
53 \begin{\absnamepos}%
54 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
55 \end\absnamepos%
56 %%           \vspace{\abstitleskip}%
57           \fi
58           \fi
59           \vspace{\abstitleskip}%
60           \fi
61           \put@bsintoc%
62 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
63 {\par\end{@bstr@ctlist}}
64 \fi
65

```

---

File 6 **l warp-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, l warp allows a variable number of columns up to three.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{adjmulticol}

2 \RequirePackage{multicol}

adjmulticols * {\langle numcols\rangle} {\langle left margi\rangle} {\langle right margin\rangle}

3 \NewDocumentEnvironment{adjmulticols}{s m m m}
4 {%

```

Compute the margins, and limit to positive only:

```

5 \setlength{\LWR@templengthone}{#3}%
6 \ifdimcomp{\LWR@templengthone}{<}{0pt}{\setlength{\LWR@templengthone}{0pt}}{}%
7 \setlength{\LWR@templengthtwo}{#4}%
8 \ifdimcomp{\LWR@templengthtwo}{<}{0pt}{\setlength{\LWR@templengthtwo}{0pt}}{}%

```

---

If one column is specified, use a <div> of class `singlecolumn`, else use `multicols`:

```
9 \newcommand*{\LWR@mcolltype}{multicols}%
10 \ifnumcomp{#2}{=}{1}{\renewcommand*{\LWR@mcolltype}{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@mcolltype}%
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 **l warp-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 237.1.

**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@htmlltagc{%
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@htmlltagc{/span}%
21 }%
22 %
23 \let\origComment\Comment%
24 %
25 \renewcommand{\Comment}[1]{\InlineClass{floatright}{\origComment{#1}}}%
26 }

27 \end{warpHTML}
```

File 10 **l warp-alltt.sty**§ 89 Package **alltt**

Pkg **alltt** alltt is patched for use by l warp.

**for HTML output:** 1 \LWR@ProvidesPackagePass{alltt}

```

2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching alltt.}
4 \AtBeginEnvironment{alltt}{%
5 \LWR@forcenewpage
6 \LWR@atbeginverbatim{alltt}\unskip\LWR@origvspace*{-\baselineskip}%
7 }
8 \AfterEndEnvironment{alltt}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
9 }
```

File 11 **l warp-amsthm.sty**§ 90 Package **amsthm**

(Based on original code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)

Pkg **amsthm** amsthm is patched for use by l warp.

Table 11: 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}%
8   }{%
9     \thm@style{#1}%
10    \renewcommand{\LWR@newtheoremstyle}{#1}%
11  }%
12 }
```

Patched to remember the style for this theorem type:

```
13 \def\@xnthm#1#2{%
14   \csedef{\LWR@thmstyle#2}{\LWR@newtheoremstyle}%
15   \let\@tempa\relax
16   \xp\@if definable\csname #2\endcsname{%
17     \global\xp\let\csname end#2\endcsname\@endtheorem
18     \ifx *##1% unnumbered, need to get one more mandatory arg
19       \edef\@tempa##1{%
20         \gdef\xp\@nx\csname #2\endcsname{%
21           \@nx\@thm{\xp\@nx\csname th@\the\thm@style\endcsname}%
22           {}{##1}}%
23       \else % numbered theorem, need to check for optional arg
24         \def\@tempa{\oparg{\cynthm{#2}}[]}%
25       \fi
26       \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{\LWR@thmstyle#2}}}%
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@forcenewline% new
67     \BlockClass{amsthmproof}% new
68     \pushQED{\qed}%
69     \normalfont \topsep6\p@\relax
70     \trivlist
71     \item[\hspace{-\labelsep}\hspace{0.5em}\textbf{I}\textit{nlineClass{amsthmproofname}{#1}@addpunct{.}}]\ignorespaces% changes
72 }{%
73     \InLineClass{theoremendmark}{\popQED}\endtrivlist%
74     \endBlockClass% new
75     \c@ndpefalse

```

77 }

---

File 12 l warp-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 L<sup>A</sup>T<sub>E</sub>X 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\@secCntformat%
10 }
```

---

File 13 l warp-any size.sty

§ 92 Package **any size**

Pkg any size any size is emulated.

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{any size}

2 \def\paperSize#1#2{}
3 \def\marginSize#1#2#3#4{}
```

---

File 14 l warp-appendix.sty

§ 93 Package **appendix**

Pkg appendix appendix is patched for use by l warp.

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

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{appendix}

2 \renewcommand*{\@chap@pppage}{%
3 \part*{\appendixpagename}
4 \if@dotoc@pp
5 \addappheadtotoc
6 \fi
7 }
8
9 \renewcommand*{\@sec@pppage}{%
10 \part*{\appendixpagename}
11 \if@dotoc@pp
12 \addappheadtotoc
13 \fi
14 }
```

#### File 15 `lwarf-arabicfront.sty`

### § 94 Package **arabicfront**

Pkg `arabicfront` `arabicfront` is ignored.

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{arabicfront}
```

#### File 16 `lwarf-authblk.sty`

### § 95 Package **authblk**

Pkg `authblk` `authblk` is patched for HTML.

**package support** lwarf supports the native L<sup>A</sup>T<sub>E</sub>X 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.

*(Based on original code by PATRICK W. DALY.)*

**for HTML output:** Require that `authblk` be loaded before `titling`:

---

```

1 \@ifpackageloaded{titling}{
2 \PackageError{l warp-authblk}
3 {Package authblk must be loaded before titling}
4 {Titling appends authblk's author macro, so authblk must be loaded first.}
5 }
6 {}

```

Load authblk:

```
7 \LWR@ProvidesPackagePass{authblk}
```

Patch to add a class for the affiliation:

```

8 \LetLtxMacro\LWRAB@affil\affil
9
10 \renewcommand{\affil}[2][]{%
11 \LWRAB@affil[#1]{\protect\InlineClass{affiliation}{#2}}%
12 }

```

Create an HTML break for an \authorcr:

```
13 \renewcommand*{\authorcr}{\protect\LWR@newlinebr}
```

---

#### File 17 l warp-balance.sty

##### § 96 Package **balance**

Pkg balance Emulated.

**for HTML output:** Discard all options for l warp-balance:

```

1 \LWR@ProvidesPackageDrop{balance}
2 \newcommand*{\balance}{}%
3 \newcommand*{\nobalance}{}%

```

---

#### File 18 l warp-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`** `\l delim` and `\r delim` use `\multirow`, so `\mrowcell` must be used in the proper number of empty cells in the same column below `\l delim` or `\r delim`, but not in cells which are above or below the delimiter:

---

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

---

a	b
c	d
e	f
g	h

**for HTML output:** First, remove the temporary definitions of `\l delim` and `\r delim`, which were previously defined for tabular scanning in case `bigdelim` was not loaded:

```
1 \let\l delim\relax
2 \let\r delim\relax
```

Next, load the package's new definitions:

```
3 \LWR@ProvidesPackagePass{bigdelim}
```

Remember the print-mode versions:

```
4 \LetLtxMacro{\LWR@origldelim}{\l delim}
5 \LetLtxMacro{\LWR@origrdelim}{\r delim}
```

```
\l delim {<1:delimiter>} {<2:#rows>} {<3:width>} [<4:text>]
\r delim
6 \RenewDocumentCommand{\l delim}{m m m O{} }{%
7 \renewcommand{\LWR@multirborder}{right}%
8 \multirow{#2}{#3}{#4 \InlineClass{l delim}{#1}}%
9 }
10
11 \RenewDocumentCommand{\r delim}{m m m O{} }{%
12 \renewcommand{\LWR@multirborder}{left}%
13 \multirow{#2}{#3}{\InlineClass{r delim}{#1} #4}%
14 }
```

---

When entering a `\teximage`, 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 `\teximage`, 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 **l warp-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 **l warp-boxedminipage.sty**

§ 101 Package **boxedminipage**

Pkg boxedminipage boxedminipage is superceded by `boxedminipage2e`.

**for HTML output:** 1 \LWR@loadnever{boxedminipage}{boxedminipage2e}

---

File 23 **l warp-boxedminipage2e.sty**

§ 102 Package **boxedminipage2e**

Pkg boxedminipage2e `boxedminipage2e` is emulated.

**for HTML output:** Discard all options for `l warp-boxedminipage2e`:

```
1 \LWR@ProvidesPackageDrop{boxedminipage2e}
```

---

```

2 \newenvironment{boxedminipage}{%
3 \begin{BlockClass}{framebox}%
4 \minipage{%
5 }%
6 {%
7 \endminipage%
8 \end{BlockClass}%
9 }

```

---

File 24 **lwarf-caption2.sty**

§ 103 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 25 **lwarf-ccaption.sty**

§ 104 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 26 **lwarf-changepage.sty**

§ 105 Package **changepage**

Pkg **changepage** **changepage** is ignored.

**for HTML output:** Discard all options for lwarf-changepage:

```

1 \LWR@ProvidesPackageDrop{changepage}

2 \newif\ifoddpage
3 \DeclareRobustCommand{\checkoddpage}{\oddpagetrue}
4 \DeclareRobustCommand{\changetext}[5]{}
5 \DeclareRobustCommand{\changepage}[9]{}
6 \newenvironment{adjustwidth}[2]{}{}
7 \newenvironment{adjustwidth*}[2]{}{}

```

---

File 27 **lwarf-chngpage.sty**

§ 106 Package **chngpage**

Pkg chngpage chngpage is superceded by changepage.

for HTML output: 1 \LWR@loadnever{chngpage}{changepage}

---

File 28 **lwarf-chappg.sty**

§ 107 Package **chappg**

Pkg chappg chappg is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{chappg}

2 \renewcommand{\pagenumbering}[2] [] {}  
3 \providecommand{\chappgsep}{--}

---

File 29 **lwarf-color.sty**

§ 108 Package **color**

Pkg color Allowed but ignored. xcolor is then required as well.

color is superceded by xcolor, and lwarf 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 30 **lwarf-crop.sty**

§ 109 Package **crop**

Pkg crop Emulated.

**for HTML output:** Discard all options for l warp-crop:

```
1 \LWR@ProvidesPackageDrop{crop}
2 \newcommand*{\crop}[1] []
3 \newcommand*{\cropdef}[6] []
```

---

File 31 **l warp-cuted.sty**

§ 110 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 32 **l warp-cutwin.sty**

§ 111 Package **cutwin**

Pkg cutwin Emulated.

**for HTML output:** Discard all options for l warp-cutwin:

```
1 \LWR@ProvidesPackageDrop{cutwin}
2 \newcommand*{\opencutleft}{}
3 \newcommand*{\opencutright}{}
4 \newcommand*{\opencutcenter}{}
5 \newcommand*{\cutfuzz}{}
6
7 \newenvironment{cutout}[4]
8 {\marginpar{\windowpagestuff}}
9 {}
10
11 \newcommand*{\windowpagestuff}{}
12
13 \newcommand*{\pageinwindow}{%
14 % \begin{minipage}{.3\linewidth}
15 \windowpagestuff
16 % \end{minipage}}
```

---

```

17 }
18
19 \newenvironment{shapedcutout}[3]
20 {\marginpar{\picinwindow}}
21 {}
22
23 \newcommand*{\putstuffinpic}{}
24
25 \newcommand*{\picinwindow}{%
26 \begin{picture}(0,0)
27 \putstuffinpic
28 \end{picture}}

```

---

File 33 **lwarf-dblfnote.sty**

§ 112 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 34 **lwarf-dcolumn.sty**

§ 113 Package **dcolumn**

Pkg **dcolumn** dcolumn is emulated during HTML output, and the dcolumn package is ignored.

```

1 \LWR@ProvidesPackageDrop{dcolumn}

```

---

File 35 **lwarf-draftwatermark.sty**

§ 114 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 36 **lwarf-ebook.sty**

§ 115 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 37 **lwarp-ellipsis.sty**

§ 116 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 38 **lwarp-emptypage.sty**

§ 117 Package **emptypage**

Pkg **emptypage** emptypage is ignored.

**for HTML output:** Discard all options for lwarp-emptypage:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

---

File 39 **lwarp-endnotes.sty**

§ 118 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 40 **l warp-enumerate.sty**

§ 119 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 l warp and thus must no longer be dropped.

**for HTML output:** 1 \LWR@ProvidesPackagePass{enumerate}

---

File 41 **l warp-enumitem.sty**

§ 120 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 42 **lwarf-epigraph.sty**

## § 121 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 43 **lwarf-eso-pic.sty**

§ 122 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 44 **lwarf-everypage.sty**

§ 123 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 45 **l warp-everyshi.sty**

§ 124 Package **everyshi**

Pkg **everyshi** Emulated.

**for HTML output:** Discard all options for l warp-everyshi:

```
1 \LWR@ProvidesPackageDrop{everyshi}

2 \newcommand*{\EveryShipout}[1]{}
3 \newcommand*{\AtNextShipout}[1]{}
```

---

File 46 **l warp-extramarks.sty**

§ 125 Package **extramarks**

Pkg **extramarks** extramarks is not used.

**for HTML output:** Discard all options for l warp-extramarks:

```
1 \LWR@ProvidesPackageDrop{extramarks}

2 \newcommand*{\extramarks}[2]{}
3 \newcommand*{\firstleftxmark}{}%
4 \newcommand*{\lastleftxmark}{}%
5 \newcommand*{\firstrightxmark}{}%
6 \newcommand*{\lastrightxmark}{}%
7 \newcommand*{\firstxmark}{}%
8 \newcommand*{\lastxmark}{}%
9 \newcommand*{\topxmark}{}%
10 \newcommand*{\topleftxmark}{}%
11 \newcommand*{\firstleftmark}{}%
12 \newcommand*{\lastrightmark}{}%
```

---

File 47 **lwarf-fancybox.sty**

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

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

**framing alternatives** `\fbox` works with fancybox. Also see lwarf'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 lwarf, 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** lwarf does not support the `verbatim` environment inside a `span`, `box`, or fancybox's `\Sbox`, but a `verbatim` may be placed inside a `fminipage`. The fancybox documentation's example `FramedVerb` may be defined as:

---

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

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

**indented alignment** LVerbatim, \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 \renewcommand*{\@doublebox}[1]{%
9 \ifbool{FormatWP}{%
10 {\InlineClass[border:1px double black]{doublebox}{#1}}{%
11 {\InlineClass{doublebox}{#1}}{%
12 }{%
13 \renewcommand*{\@ovalbox}[2]{%
14 \ifbool{FormatWP}{%
15 {\InlineClass[border:1px solid black; border-radius:1ex]{ovalbox}{#2}}{%
16 {\InlineClass{ovalbox}{#2}}{%
17 {\InlineClass{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 \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 48 **l warp-fancyhdr.sty**

§ 127 Package **fancyhdr**

Pkg **fancyhdr** fancyhdr is nullified.

**for HTML output:** Discard all options for l warp-fancyhdr:

```
1 \LWR@ProvidesPackageDrop{fancyhdr}

2 \newcommand*{\fancyhead}[2][]{}
3 \newcommand*{\fancyfoot}[2][]{}
4 \newcommand*{\fancyhf}[2][]{}
5 \newcommand*{\fancypagestyle}[2][]{}
6 \newcommand*{\lhead}[2][]{}
7 \newcommand*{\chead}[2][]{}
8 \newcommand*{\rhead}[2][]{}
9 \newcommand*{\lfoot}[2][]{}
10 \newcommand*{\cfoot}[2][]{}
11 \newcommand*{\rfoot}[2][]{}
12 \newcommand*{\headrulewidth}{}%
13 \newcommand*{\footrulewidth}{}%
14 \newcommand*{\fancyheadoffset}[2][]{}
15 \newcommand*{\fancyfootoffset}[2][]{}
16 \newcommand*{\fancyhfoffset}[2][]{}
17 \newcommand*{\iffloatpage}[2]{#2}
18 \newcommand*{\ifftopfloat}[2]{#2}
19 \newcommand*{\iffbotfloat}[2]{#2}
```

---

File 49 **l warp-fancyvrb.sty**

§ 128 Package **fancyvrb**

(Based on original code by TIMOTHY VAN ZANDT.)

Pkg **fancyvrb** fancyvrb is supported with some patches.

**for HTML output:**

```
1 \RequirePackage{xcolor}%
2 \LWR@ProvidesPackagePass{fancyvrb}

3 \begin{warpHTML}
```

Initial default patch for fancyvrb:

```
4 \fvset{frame=none}%
```

For \VerbatimFootnotes:

```
5 \renewcommand{\VerbatimFootnotes}{%
6 \PackageError{l warp}{%
7 {Verbatim footnotes are not yet supported by l warp.}%
8 {This may be improved some day.}%
9 }%
```

After the preamble is loaded, after any patches to Verbatim:

```
10 \AfterEndPreamble{%
11 \LWR@traceinfo{Patching Verbatim.}}
```

Remember the original defintion of Verbatim:

```
12 \let\LWRFV@origVerbatim\Verbatim
```

Env **Verbatim** Patched to place the environment in a `fancyvrb` div, and the label in a `fancyvrblabel` div. Also corrects the left margin for line numbers. Also uses `VerbatimHTMLWidth` to control placement of line numbers on the right. Aligning the right margin requires knowing the width.

```
13 \renewcommand*{\Verbatim}{%
14 \LWR@forcenewpage%
15 \LWRFV@origVerbatim%
16 }
```

**\LWR@FVstyle** Holds the style of the verbatim.

```
17 \newcommand*{\LWR@FVstyle}{}%
```

The following patches to Verbatim are executed at the start and end of the environment, depending on the choice of frame. Original code is from the `fancyvrb` package.

```
18 \newcommand*{\LWR@fvstartnone}{%
19 \LWR@traceinfo{fvstartnone}%
20 \hbox to\z@{\LWR@atbeginverbatim[\LWR@FVstyle]{verbatim}}%
21 }
22
23 \newcommand*{\LWR@f vendnone}{%
24 \LWR@traceinfo{f vendnone}%
}
```

```

25 \hbox to\z@{\LWR@afterendverbatim}%
26 }
27
28 \newcommand*{\LWR@fvstartsingle}{%
29 \LWR@traceinfo{fvstartsingle}%
30 \LWR@fvstartnone%
31 \FV@BeginListFrame@Single%
32 }
33
34 \newcommand*{\LWR@fvendsingle}{%
35 \LWR@traceinfo{fvendsingle}%
36 \FV@EndListFrame@Single%
37 \LWR@fvendnone%
38 }
39
40 \newcommand*{\LWR@fvstartline}{%
41 \LWR@traceinfo{fvstartline}%
42 \LWR@fvstartnone%
43 \FV@BeginListFrame@Lines%
44 }
45
46 \newcommand*{\LWR@fvendline}{%
47 \LWR@traceinfo{fvendline}%
48 \FV@EndListFrame@Lines%
49 \LWR@fvendnone%
50 }

```

The following patches select the start/left/right/end behaviors depending on frame. Original code is from the fancyvrb package.

```

51
52 \newcommand*{\LWR@FVfindcurrentcolor}{%
53 \protect\colorlet{LWR@current@color}{.}%
54 \protect\convertcolorspec[named]{LWR@current@color}{HTML}\LWR@tempcolor%
55 }
56
57 \newcommand*{\LWR@FVtextstyle}{%
58 \LWR@FVfindcurrentcolor
59 \ifdefstring{\LWR@tempcolor}{000000}%
60 {}%
61 {color: \#\LWR@tempcolor ; }%
62 }
63
64
65 \newcommand*{\LWR@FVfindbordercolor}{%
66 \FancyVerbRuleColor%
67 \LWR@FVfindcurrentcolor
68 \color{black}%
69 }

```

```
70
71 % border width of \FV@FrameRule
72 \newcommand*{\LWR@FVborderstyle}[1]{%
73 padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ;
74 \LWR@FVfindbordercolor
75 border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt solid \#\LWR@tempcolor ;
76 }
77
78 \def\FV@Frame@none{%
79 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle}%
80 \let\FV@BeginListFrame\LWR@fvstartnone%
81 \let\FV@LeftListFrame\relax%
82 \let\FV@RightListFrame\relax%
83 \let\FV@EndListFrame\LWR@fvendnone}
84
85 \FV@Frame@none% default values
86
87 \def\FV@Frame@singl{%
88 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{} }%
89 \let\FV@BeginListFrame\LWR@fvstartsingl%
90 \let\FV@LeftListFrame\FV@LeftListFrame@Singl%
91 \let\FV@RightListFrame\FV@RightListFrame@Singl%
92 \let\FV@EndListFrame\LWR@fvendsingl}
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\FrameFillLine\relax%
122 \else%
123 \tempdima\FrameRule\relax%
124 \multiply\tempdima-\tw@%
125 \edef\FrameFillLine{%
126 {\noexpand\FancyVerbFillColor{\vrule@width\number\tempdima sp}%
127 \kern-\number\tempdima sp}}%
128 \fi%
129 \let\BeginListFrame\LWR@fvstartnone%
130 \let\LeftListFrame\LeftListFrame@Single%
131 \let\RightListFrame\relax%
132 \let\EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the fancyverb package.

```

133 \def\SingleFrameLine#1{%
134   \hbox to\z@{%
135 %     \kern\leftmargin
136     \ifnum#1=\z@\relax
137       \let\Label\LabelBegin
138     \else
139       \let\Label\LabelEnd
140     \fi
141     \ifx\Label\relax
142 %       \FancyVerbRuleColor{\vrule \width\linewidth \height\FrameRule}%
143     \else
144       \ifnum#1=\z@
145 %         \setbox\z@\hbox{\strut\enspace\LabelBegin\enspace\strut}%
146         \ifx\LabelPositionTopLine\relax
147           \else
148             \LWR@FVfindbordercolor
149             \LWR@htmltagc{div class="fancyvrlabel" style="color: \#\LWR@tempcolor"}%
150             \LWR@origtextrm{\LabelBegin}%
151             \LWR@htmltagc{/div}
152           \fi
153         \else
154 %         \setbox\z@\hbox{\strut\enspace\LabelEnd\enspace\strut}%
155         \ifx\LabelPositionBottomLine\relax
156           \else
157             \LWR@FVfindbordercolor
158             \LWR@htmltagc{div class="fancyvrlabel" style="color: \#\LWR@tempcolor"}%
159             \LWR@origtextrm{\LabelEnd}
160             \LWR@htmltagc{/div}
161           \fi
162         \fi
163       \fi
164     }

```

```

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       \ifcsvoid{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 50 **lwarf-figcaps.sty**

§ 129 Package **figcaps**

Pkg **figcaps** Emulated.

**for HTML output:** Discard all options for lwarf-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 51 **lwarf-fix2col.sty**

§ 130 Package **fix2col**

Pkg **fix2col** fix2col is ignored.

**for HTML output:** 1 \LWR@ProvidesPackageDrop{fix2col}

---

File 52 **lwarf-float.sty**

§ 131 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 {\DeclaringFloatingEnvironment[fileext=#3,within=#4]{#1}}{%
5 {\DeclaringFloatingEnvironment[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` {*<type>*} {*(name)*}

Sets the text name of the float, such as “Figure”.

```
9 \NewDocumentCommand{\floatname}{m +m}{%
10 \SetupFloatingEnvironment{#1}{name=#2}%
11 }
```

`\floatplacement` {*<type>*} {*(placement)*}

Float placement is ignored.

```
12 \newcommand*{\floatplacement}[2]{%
13 \SetupFloatingEnvironment{#1}{placement=#2}%
14 }
```

`\floatstyle` {*<style>*}

Float styles are ignored.

```
15 \newcommand{\floatstyle}[1]{%
16 }
```

`\restylefloat` \* {*<style>*}

Float styles are ignored.

```
17 \NewDocumentCommand{\restylefloat}{s m}{%
18 }
```

---

File 53 **l warp-floatfl t.sty**

§ 132 Package **floatfl t**

Pkg **floatfl t** Emulated.

**for HTML output:** Discard all options for l warp-floatfl t:

```
1 \LWR@ProvidesPackageDrop{floatfl t}
```

Env [⟨⟩] offset {⟨type⟩} {⟨width⟩} Borrowed from the l warp version of keyfloat:

```
2 \NewDocumentEnvironment{KFLTfloatfl t@marginfloat}{O{-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{KFLTfloatfl t@marginfloat}{figure}{#2}%
21   {\end{KFLTfloatfl t@marginfloat}}
```

Env **floatingtable** [⟨placement⟩]

```
22 \DeclareDocumentEnvironment{floatingtable}{o}
23   {\begin{KFLTfloatfl t@marginfloat}{table}{1.5in}%
24   {\end{KFLTfloatfl t@marginfloat}}
```

---

File 54 **lwarf-floatpag.sty**

§ 133 Package **floatpag**

Pkg **floatpag** Emulated.

**for HTML output:** Discard all options for lwarf-floatpag:

```
1 \LWR@ProvidesPackageDrop{floatpag}
2 \newcommand*{\floatpagestyle}[1]{}
3 \newcommand*{\rotfloatpagestyle}[1]{}
4 \newcommand*{\thisfloatpagestyle}[1]{}
```

---

File 55 **lwarf-floatrow.sty**

§ 134 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. lwarf 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 {%

```

For subcaption:

```

12 \ifbool{LWR@insubfloatrow}{%
13 {%

```

subfigure and subtable environments take width as an argument.

```

14 \IfValueTF{#3}{%
15 {\@nameuse{sub#2}{#3}}{%
16 {\@nameuse{sub#2}{\linewidth}}{%
17 }% subcaption in a subfloatrow
18 {%

```

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 {%

```

For subfig:

```

29 \ifbool{LWR@insubfloatrow}{%
30 {%

```

\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 \Cnameuse{#2}
38 #6
39
40 #7
41 \Cnameuse{end#2}
42 }% subfig package, but not a subfig
43 }% assume subfig
44 }

```

Not used:

```

45 \newcommand*\nocapbeside(){}
46 \newcommand*\capbeside(){}
47 \newcommand*\capttop(){}
48 \newlength{\FBwidth}
49 \setlength{\FBwidth}{.3\linewidth}
50 \newlength{\FBheight}
51 \setlength{\FBheight}{2in}
52 \newcommand*\useFCwidth(){}
53 \newcommand{\floatsetup}[2][]{}
54 \newcommand{\thisfloatsetup}[1]({})
55 \newcommand{\clearfloatsetup}[1]({})
56 \newcommand*\killfloatstyle(){}

```

\newfloatcommand {⟨1 command⟩} {⟨2 captype⟩} [⟨3 preamble⟩] [⟨4 default width⟩]

Preamble and default width are ignored.

```

57 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
58 \Cnamedef{#1}{%
59 \floatbox{#2}{%
60 }%
61 }

```

\renewfloatcommand {⟨1 command⟩} {⟨2 captype⟩} [⟨3 preamble⟩] [⟨4 default width⟩]

Preamble and default width are ignored.

```

62 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
63 @namedef{#1}{%
64 \floatbox{#2}%
65 }%
66 }

\ffigbox  [<width>] [<height>] [<vposn>] {<caption commands>} {<contents>}%
67 \newfloatcommand{ffigbox}{figure}[\nocapbeside] []

\ttabbox  [<width>] [<height>] [<vposn>] {<caption commands>} {<contents>}%
68 \newfloatcommand{ttabbox}{table}[\capttop] [\FBwidth]

\fcapside  [<width>] [<height>] [<vposn>] {<caption commands>} {<contents>}%
69 \newfloatcommand{fcapside}{figure}[\capbeside] []

```

Env **floatrow** [<numfloats>]

The row of floats is placed into a <div> of class **floatrow**.

```

70 \newenvironment*{floatrow}[1][2]%
71 {%
72 \LWR@forcenewpage%
73 \BlockClass{floatrow}

```

While inside the floatrow, divide the \linewidth by the number of floats.

```

74 \booltrue{\LWR@infloatrow}%
75 \setlength{\linewidth}{6in/#1}%
76 }%
77 {%
78 \boolfalse{\LWR@infloatrow}%
79 \endBlockClass%
80 }

```

Keys for \DeclareNewFloatType:

```

81 \newcommand*{\LWR@frowkeyplacement}{}%
82 \newcommand*{\LWR@frowkeyname}{}%
83 \newcommand*{\LWR@frowkeyfileext}{}%
84 \newcommand*{\LWR@frowkeywithin}{}%
85 \newcommand*{\LWR@frowkeycapstyle}{}%
86 %
87 \define@key{frowkeys}{placement}{}%
88 \define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{\#1}}%
89 \define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{\#1}}%
90 \define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{\#1}}%

```

```

91 \define@key{frowkeys}{relatedcapstyle}{}%
```

\DeclareNewFloatType {⟨type⟩} {⟨options⟩}  
 Use \listof{type}{Title} to print a list of the floats.

```

92 \newcommand*\{\DeclareNewFloatType\}[2]{%
```

Reset key values:

```

93 \renewcommand*\{\LWR@frowkeyplacement\}{}%
94 \renewcommand*\{\LWR@frowkeyname\}{}%
95 \renewcommand*\{\LWR@frowkeyfileext\}{}%
96 \renewcommand*\{\LWR@frowkeywithin\}{}%
97 \renewcommand*\{\LWR@frowkeycapstyle\}{}%
```

Read new key values:

```

98 \LWR@traceinfo{about to setkeys frowkeys}%
99 \setkeys{frowkeys}{#2}%
100 \LWR@traceinfo{finished setkeys frowkeys}%
```

Create a new float with optional [within]:

```

101 \ifthenelse{\equal{\LWR@frowkeywithin}{}}%
102 {%
103   \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\%
104   \LWR@frowkeyfileext}%
105   \newfloat{\#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}%
106 }%
107 {%
108   \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\%
109   \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
110   \newfloat{\#1}{\LWR@frowkeyplacement}%
111   {\LWR@frowkeyfileext}[\LWR@frowkeywithin]%
112   \LWR@traceinfo{finished newfloat #1}%
113 }%
```

Rename the float if a name was given:

```

114 \ifthenelse{\equal{\LWR@frowkeyname}{}}%
115 {}%
116 {\floatname{\#1}{\LWR@frowkeyname}}%
117 }
```

Not used:

```

118 \newcommand{\buildFBBOX}[2]{}
119 \newcommand*\{\CenterFloatBoxes\}{}%
120 \newcommand*\{\TopFloatBoxes\}{}%
121 \newcommand*\{\BottomFloatBoxes\}{}%
122 \newcommand*\{\PlainFloatBoxes\}{}%
123
```

```
124 \newcommand{\capsubrowsettings}{}  
125  
126 \NewDocumentCommand{\RawFloats}{o o}{}  
  
\RawCaption {⟨text⟩}  
To be used inside a minipage or parbox.  
127 \newcommand{\RawCaption}[1]{#1}  
  
\floatfoot {⟨text⟩}  
Places additional text inside a float, inside a css <div> of class floatfoot.  
128 \NewDocumentCommand{\floatfoot}{s +m}{%  
129 \begin{BlockClass}{floatfoot}  
130 #2  
131 \end{BlockClass}  
132 }  
  
Used to compute \linewidth.  
133 \newbool{LWR@insubfloatrow}  
134 \boolfalse{LWR@insubfloatrow}  
  
Env subfloatrow [⟨num_floats⟩]  
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 56 l warp-flushend.sty

§ 135 Package **flushend**

Pkg flushend Emulated.

for HTML output: Discard all options for l warp-flushend:

```
1 \LWR@ProvidesPackageDrop{flushend}
2 %     \end{ma-crocode}
3 %
4 %     \begin{macrocode}
5 \newcommand*\flushend{}%
6 \newcommand*\raggedend{}%
7 \newcommand*\flushcolsend{}%
8 \newcommand*\raggedcolsend{}%
9 \newcommand*\atColsBreak[1]{}%
10 \newcommand*\atColsEnd[1]{}%
11 \newcommand*\showcolsendlrule{}%
```

---

File 57 l warp-fncychap.sty

§ 136 Package **fncychap**

Pkg fncychap fncychap is ignored.

for HTML output: Discard all options for l warp-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 58 **l warp-fnpos.sty**

§ 137 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 59 **l warp-fontenc.sty**

§ 138 Package **fontenc**

Pkg **fontenc** Error if fontenc is loaded after l warp.

Discard all options for l warp-fontenc:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{fontenc}

2 \LWR@loadbefore{fontenc}
```

---

File 60 **l warp-fontspec.sty**

§ 139 Package **fontspec**

Pkg **fontspec** Error if fontspec is loaded after l warp.

Discard all options for l warp-fontspec:

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{fontspec}
2 \LWR@loadbefore{fontspec}
```

---

File 61 **l warp-footmisc.sty**

## § 140 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\p0 \hrule \kern2.6\p0}
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 62 **lwarp-footnote.sty**

## § 141 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@htmlltagc{/}\LWR@tagregularparagraph}%
11 \LWR@orignewline%
12 }

```

Removed print-version formatting:

```

13 \def\fn@startfntext{%
14   \setbox\z@\vbox\bgroup%
15   \fn@startnote%
16   \fn@prefntext%
17   \ignorespaces%
18 }

```

Removed print-version formatting, added closing paragraph tag:

```

19 \def\fn@endfntext{%
20   \LWR@htmlltagc{/}\LWR@tagregularparagraph}%

```

---

```

21      \LWR@orignewline%
22      \fn@postfntext%
23  \egroup%
24  \begingroup%
25      \let\@makefntext\@empty%
26      \let\@finalstrut\gobble%
27      \LetLtxMacro\rule\@gobbletwo% *8* also the optional argument?
28      \@footnotetext{\unvbox\z@}%
29  \endgroup%
30 }

```

These have been redefined, so re-\let them again:

```

31 \let\endfootnote\fn@endfntext
32 \let\endfootnotetext\endfootnote

```

---

#### File 63 l warp-footnotehyper.sty

### § 142 Package **footnotehyper**

Pkg **footnotehyper** footnotehyper is a hyperref-safe version of footnote. For l warp, footnotehyper is emulated.

**for HTML output:** Discard all options for l warp-footnotehyper:

```

1 \RequirePackage{footnote}
2 \LWR@ProvidesPackageDrop{footnotehyper}

```

---

#### File 64 l warp-framed.sty

### § 143 Package **framed**

(Based on original code by DONALD ARSENEAU.)

Pkg **framed** framed is supported and patched by l warp.

**for HTML output:** Accept all options for l warp-framed:

```

1 \LWR@ProvidesPackagePass{framed}
2 \RequirePackage{xcolor}%
3 for \convertcolorspec
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 {\langle toptitle\rangle} {\langle bottitle\rangle} {\langle thicknessstop\rangle} {\langle bottom\rangle} {\langle left\rangle} {\langle right\rangle}
{\langle text contents\rangle}

62 \renewcommand{\CustomFBox}[7]{%
63   \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
64   \LWR@forcenewpage
65   \begin{BlockClass}[border: 3px solid \#\LWR@tempcolor]{framed}%
66   \ifthenelse{\isempty{#1}}{}{%
67     \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
68     \textcolor{TFTTitleColor}{\textbf{#1}}%
69     \end{BlockClass}%
70   }%
71 }
72 #7
73
74 \ifthenelse{\isempty{#2}}{}{%
75   \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
76   \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
77   \textcolor{TFTTitleColor}{\textbf{#2}}%
78   \end{BlockClass}%
79 }%
80 \end{BlockClass}
81 }

\TitleBarFrame [\langle marker\rangle] {\langle title\rangle} {\langle contents\rangle}

82 \renewcommand\TitleBarFrame[3][]{%
83   \CustomFBox
84   {#2}{}%
85   \fboxrule\fboxrule\fboxrule\fboxrule
86   {#3}%
87 }

88 \renewcommand{\TF@Title}[1]{#1}

MakeFramed {\langle settings\rangle}

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 65 **lwarp-ftnright.sty**

§ 144 Package **ftnright**

Pkg **ftnright** ftnright is ignored.

**for HTML output:** Discard all options for lwarp-ftnright:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

---

File 66 **lwarp-fullpage.sty**

§ 145 Package **fullpage**

Pkg **fullpage** Emulated.

**for HTML output:** Discard all options for lwarp-fullpage:

```
1 \LWR@ProvidesPackageDrop{fullpage}
```

---

File 67 **lwarp-fullwidth.sty**

§ 146 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 \minipage{fullwidth}{%  
4 \minipage{\linewidth}{%  
5 }  
6 {%
```

---

```
7 \endminipage%
8 }
```

---

File 68 **lwarp-geometry.sty**

§ 147 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}{}
3 \renewcommand*\{\newgeometry}{}
4 \renewcommand*\{\restoregeometry}{}
5 \renewcommand*\{\savegeometry}{}
6 \renewcommand*\{\loadgeometry}{}
```

---

File 69 **lwarp-glossaries.sty**

§ 148 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 **\nameref**s in **HTML**, which could become a very long string if many items are referenced. For now, the number list is simply turned off.

**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 lwarf 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 lwarfmk printglossary** lwarfmk has the commands `lwarfmk printglossary` and `lwarfmk htmlglossary` to process the glossaries created by `glossaries` using `xindy`.

**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   \ifdef\empty\@@glossarysecstar
9   {%
10     \csname\@@glossarysec\endcsname{#2}%
11   }%
12 }
```

In the original, the toc entry was made before the section, thus linking to the phantomsection in the printed version, but for HTML this caused the link to point to the page before the glossaries. Here, the toc entry is made after the section is created:

```
13   \csname\@@glossarysec\endcsname*{#2}%
14   \gls@toc{#1}{\@@glossarysec}%
15 }%
16 \@@glossaryseclabel
17 }
```

---

File 70 **l warp-graphics.sty**

## § 149 Package **graphics**

Pkg **graphics** **graphics** is emulated.

**for HTML output:** 1 \LWR@ProvidesPackagePass{graphics}

### § 149.1 **Graphics extensions**

\DeclareGraphicsExtensions {⟨list⟩}

\AtBeginDocument allow SVG files instead of PDF:

```
2 \AtBeginDocument{
3 \DeclareGraphicsExtensions{.svg,.SVG,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}
4 \DeclareGraphicsRule{.svg}{svg}{.svg}{}
5 \DeclareGraphicsRule{.SVG}{svg}{.SVG}{}
6 }
```

Inside a **lateximage**, allow PDF instead of SVG:

```
7 \appto\LWR@restoreorigformatting{%
8 \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
9 }
```

### § 149.2 **Length conversions and graphics options**

 **whitespace** A scaled image in L<sup>A</sup>T<sub>E</sub>X by default takes only as much space on the page as it requires, but HTML browsers use as much space as the original unscaled image would have taken, with the scaled image over- or under-flowing the area.

```
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=xyz** 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}%
30   {}% not ex
31 \IfEndWith{#1}{em}{%
32   \renewcommand*{\LWR@igwidthstyle}{width:#1}%
33   {}% not em
34 \IfEndWith{#1}{\%}{%
35   \renewcommand*{\LWR@igwidthstyle}{width:#1}%
36   {}% not percent
37 \IfEndWith{#1}{px}{%
38   \renewcommand*{\LWR@igwidthstyle}{width:#1}%
39   {}% not px
40 }{}% end of length > 0pt
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 > 0pt}}{%
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@igyscale}{#1}}
```

Numerous ignored keys:

```
73 \define@key{igraph}{bb}{}
74 \define@key{igraph}{bbllx}{}
75 \define@key{igraph}{bblly}{}
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}{}

```

### § 149.3 Printing HTML styles

\LWR@rotstyle {*prefix*} {*degrees*}

Prints the rotate style with the given prefix.

*prefix* is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:rotate` style.

```

91 \newcommand*{\LWR@rotstyle}[2]{%
92   #1 transform:rotate(-#2deg);%
93 }

```

\LWR@scalestyle {*prefix*} {*xscale*} {*yscale*}

Prints the scale style with the given prefix.

*prefix* is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:scale` style.

```

94 \newcommand*{\LWR@scalestyle}[3]{%
95   #1 transform:scale(#2,#3);%
96 }

```

### § 149.4 \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         {%
114             \setlength{\linewidth}{6in}%
115             \setlength{\textwidth}{6in}%
116             \setlength{\textheight}{9in}%
117         }%
118     }{%
119 }}
```

```

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}{Opt}%
129 \setlength{\LWR@igheight}{Opt}%
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         \%
148         \IfValueTF{#3}{%
149             \%
150             \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics*[#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  {%
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 {%
185 \LWR@traceinfo[LWR@includegraphicsb: about to LWR@htmntag]%
186 \LWR@htmntag{%
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@igyscale}{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@newline}{}%
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@igyscale}{1}}%
212 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}}%
213 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
214 \LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyscale}{}%
215 %
216 \ifthenelse{\NOT\equal{\LWR@opacity}{1}}%
217 {\opacity:\LWR@opacity;}%
218 {%
219 %
220 " \LWR@newline}{}%

```

Set the class:

```

221 \LWR@origtilde{} class="\LWR@igclass" \LWR@newline%
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

```

## § 149.5 Boxes

\LWR@rotboxorigin Holds the origin key letters.

```
243 \newcommand*\{\LWR@rotboxorigin}{}%
```

\LWR@originname {\langle letter\rangle}

Given one  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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 [⟨keyval list⟩] {⟨angle⟩} {⟨text⟩}

```
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@htmlltagc{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 {<h-scale>} [<v-scale>] {<text>}

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 {<text>}

297 \let\LWR@origreflectbox{\reflectbox}
298
299 \AtBeginDocument{
300 \renewcommand{\reflectbox}[1]{\scalebox{-1}{[1]{#1}}}
301 }
```

  

```
\resizebox {<h-length>} [<v-length>] {<text>}
```

Simply prints its text argument.

```
302 \LetLtxMacro{\LWR@origresizebox}{\resizebox}
303
304 \AtBeginDocument{
305 \renewcommand{\resizebox}[3]{#3}
306 }
```

---

#### File 71 lwarf-graphicx.sty

### § 150 Package **graphicx**

Pkg **graphicx** **graphicx** is emulated.

**graphicx** loads **graphics**, which also loads **lwarf-graphics**, which remembers the original **graphics** definitions for use inside a **lateximage**, and then patches them **\AtBeginDocument** for **HTML** output.

**lwarf-graphics** handles the syntax of either **graphics** or **graphicx**.

**for HTML output:** 1 \LWR@ProvidesPackagePass{graphicx}

---

#### File 72 lwarf-grffile.sty

### § 151 Package **grffile**

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

**lwarf-grffile** now exists as a placeholder since **grffile** used to be emulated by **lwarf**, and thus older versions of **lwarf-grffile** may exist and should be overwritten by this newer version.

**for HTML output:** 1 \LWR@ProvidesPackagePass{grffile}

---

#### File 73 lwarf-hyperref.sty

### § 152 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@htmlltag{img src="#1" alt="#2" class="hyperimage"}{}%
12 \endgroup%
13 \LWR@ensuredoingapar%
14 }%
15 %
16 \newcommand{\hyperimage}{%
17 \LWR@ensuredoingapar%
18 \begingroup\catcode`\_=12%
19 \LWR@hyperimageb%
20 }

```

\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 {*1: URL*} {*2: category*} {*3: name*} {*4: text*}

Creates an HTML link to *URL#category.name* with the given text.

```

26 \NewDocumentCommand{\LWR@hyperrefb}{m m m +m}{%
27 \LWR@htmlltag{a href="#1\LWR@hashmark#2.#3"}%
28 #4%
29 \LWR@htmlltag{/a}%
30 \endgroup%
31 }

```

\LWR@hyperrefc [*label*] {*text*}

Creates text as an HTML link to the L<sup>A</sup>T<sub>E</sub>X label.

```
32 \NewDocumentCommand{\LWR@hyperrefc}{O{label} +m}{
33 \LWR@startref{#1}%
34 #2%
35 \LWR@htmltag{/a}%
36 \endgroup%
37 }
```

\hyperref {⟨1: URL⟩} {⟨2: category⟩} {⟨3: name⟩} {⟨4: text⟩} — or —  
[⟨1: label⟩] {⟨2: text⟩}

```
38 \newcommand{\hyperref}{%
39 \LWR@ensuredoingapar%
40 \begingroup\catcode`\_=12%
41 \@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
42 }
```

\hypertarget {⟨name⟩} {⟨text⟩}

Creates an anchor to name with the given text.

```
43 \NewDocumentCommand{\hypertarget}{m +m}%
44 \label{#1}%
45 #2%
46 }
```

\hyperlink {⟨name⟩} {⟨text⟩}

Creates a link to the anchor created by hypertarget, with the given link text.

```
47 \NewDocumentCommand{\hyperlink}{s m +m}%
48 \hyperref[#1]{#2}%
49 }
```

\autoref \* {⟨label⟩}

For HTML, \cleveref is used instead.

```
50 \NewDocumentCommand{\autoref}{s m}%
51 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
52 }
```

\autopageref {⟨label⟩}

For HTML, \cleveref is used instead.

```
53 \NewDocumentCommand{\autopageref}{s m}%
54 \IfBooleanTF{#1}{\pageref{#2}}{\cref{#2}}%
55 }
```

```
\pdfstringdef  {\langle macroname\rangle} {\langle TEXstring\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 TEXstring\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 \newcommand*\TextField[2][]{}

\CheckBox [{\langle parameters\rangle}] {\langle label\rangle}
67 \newcommand*\CheckBox[2][]{}

\ChoiceMenu [{\langle parameters\rangle}] {\langle label\rangle} {\langle choices\rangle}
68 \newcommand{\ChoiceMenu}[3][]{}
```

```
\PushButton  [{parameters}]{label}  
69 \newcommand*{\PushButton}[2][]{}  
  
\Submit    [{parameters}]{label}  
70 \newcommand*{\Submit}[2][]{}  
  
\Reset    [{parameters}]{label}  
71 \newcommand*{\Reset}[2][]{}  
  
\LayoutTextField  {label} {field}  
72 \newcommand*{\LayoutTextField}[2]{}  
  
\LayoutChoiceField {label} {field}  
73 \newcommand*{\LayoutChoiceField}[2]{}  
  
\LayoutCheckField {label} {field}  
74 \newcommand*{\LayoutCheckField}[2]{}  
  
\MakeRadioField {width} {height}  
75 \newcommand*{\MakeRadioField}[2]{}  
  
\MakeCheckField {width} {height}  
76 \newcommand*{\MakeCheckField}[2]{}  
  
\MakeTextField {width} {height}  
77 \newcommand*{\MakeTextField}[2]{}  
  
\MakeChoiceField {width} {height}  
78 \newcommand*{\MakeChoiceField}[2]{}  
  
\MakeFieldButton {text}  
79 \newcommand{\MakeFieldButton}[1]{}  

```

---

File 74 **lwarf-hyperxmp.sty**

§ 153 Package **hyperxmp**

Pkg **hyperxmp** Emulated.

**for HTML output:** Discard all options for lwarf-hyperxmp:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```

---

File 75 **lwarf-idxlayout.sty**

§ 154 Package **idxlayout**

Pkg **idxlayout** Emulated.

**for HTML output:** Discard all options for lwarf-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 76 **l warp-indentfirst.sty**

§ 155 Package **indentfirst**

Pkg **indentfirst** **indentfirst** is ignored.

Discard all options for l warp-indentfirst:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{indentfirst}

---

File 77 **l warp-inputenc.sty**

§ 156 Package **inputenc**

Pkg **inputenc** Error if inputenc is loaded after l warp.

Discard all options for l warp-inputenc:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{inputenc}

2 \LWR@loadbefore{inputenc}

---

File 78 **l warp-keyfloat.sty**

§ 157 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 1w-proportional widths. (The [H] option forces the use of a minipage, which internally adjusts for a virtual 6-inch wide minipage, which then corrupts the 1w option.)

**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}{O{-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{marginable}{o}
27 {\begin{KFLT@marginfloat}{table}}
28 {\end{KFLT@marginfloat}}

29 \DeclareDocumentEnvironment{keywrap}{m +m}
30 {%
31 \LWR@ensuredoingapar%
32 \setlength{\LWR@templengthone}{#1}%
33 \uselengthunit{PT}%
34 \begin{LWR@BlockClassWP}{%
35     float:right; width:\rndprintlength{\LWR@templengthone}; %
36     margin:10pt%
37 }%
38 {%
39     width:\rndprintlength{\LWR@templengthone}%
40 }%
41 {marginblock}%
42 \setlength{\linewidth}{.95\LWR@templengthone}%
43 #2%
44 \end{LWR@BlockClassWP}%
45 }
46 {%
47 }
```

```
48 }% AtBeginDocument
```

---

File 79 **l warp-layout.sty**

§ 158 Package **layout**

Pkg **layout** layout is ignored.

**for HTML output:** Discard all options for l warp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}
2 \NewDocumentCommand{\layout}{s}{}

---


```

File 80 **l warp-letterspace.sty**

§ 159 Package **letterspace**

Pkg **letterspace** letterspace is a subset of microtype, which is pre-loaded by l warp. All user options and macros are ignored and disabled.

**for HTML output:** Discard all options for l warp-letterspace:

```
1 \LWR@ProvidesPackageDrop{letterspace}
2 \newcommand*\lsstyle{}
3 \newcommand\textls[2][]{}
4 \def\textls#1{}
5 \newcommand*\lslig[1]{#1}

---


```

File 81 **l warp-lettrine.sty**

§ 160 Package **lettrine**

(Based on original code by DANIEL FLIPO.)

Pkg **lettrine** Emulated.

**for HTML output:** Discard all options for l warp-lettrine:

```
1 \LWR@ProvidesPackageDrop{lettrine}
```

The initial letter is in a <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 82 lwarf-lips.sty

§ 161 Package **lips**

Pkg lips lips is emulated during HTML output, and the lips package is ignored.

```

1 \% \LWR@ProvidesPackageDrop{lips}
2 \PackageInfo{lwarf}{Using the lwarf version of package 'lips'.}%
3 \ProvidesPackage{lwarf-lips}
4
5 \NewDocumentCommand{\Lips}{}{\textellipsis}
6
7 \NewDocumentCommand{\BracketedLips}{}{[\textellipsis]}
8
9 \let\lips\Lips
10 \let\olips\lips

```

---

```

11
12 \DeclareOption*{}
13 \DeclareOption{mla}{%
14 \let\lips\BracketedLips
15 }
16 \ProcessOptions\relax
17
18 \newcommand \LPNobreakList {}

```

---

File 83 **lwarp-listings.sty**

§ 162 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@origlst@EveryPar\lst@EveryPar
7
8 \renewcommand{\lstlisting}[2]{\hypertocfloat{1}{\lstlisting}{\listname}{#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*{\@captype}{lstlisting}%
12 \LWR@origlst@Init{\listname}%
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\lst@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 {*t/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 #1%
41 \ifx\lst@caption\empty\expandafter\lst@HRefStepCounter \else
42 \expandafter\refstepcounter
43 \fi {\lstlisting}%
44 \LWR@traceinfo{About to assign label: !\lst@label!}%
45 \ifx\lst@label\empty\else
46 \label{\lst@label}\fi

```

```

47 \LWR@traceinfo{Finished assigning the label.}%
48     \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
49     \global\let\lst@name\lst@arg \global\let\lstname\lst@name
50     \lst@ifno{lol}{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         {\begin{group}
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 #1\allowbreak \fi
77           \ifx\lst@title\empty

```

New lwarp code to create a caption:

```

78           \lst@makecaption\fnam@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}%
83 \lst@maketitle\lst@title% lwarp
84 \end{BlockClass}%

```

```

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 lwarf:

```

94 %           \ifx #1\allowbreak \fi
95           \endgroup}{}%
96 \fi
97 \LWR@traceinfo{end of making a listings display caption}%
98 \else
99 \LWR@traceinfo{INLINE}%
100 \fi
101 \LWR@traceinfo{DONE WITH CAPTION at #1}%
102 }

```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```

103 \lst@Key{numbers}{none}{%
104   \let\lst@PlaceNumber\empty
105   \lstKV@SwitchCases{#1}%
106   {none&\\%
107     left&\def\lst@PlaceNumber{%
108 \% \llap{%
109 \LWR@orignormalfont%
110 \lst@numberstyle{\the\lstnumber}\kern\lst@numbersep%
111 \% }%
112 }%
113 \\%
114   right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont%
115             \kern\VerbatimHTMLWidth \kern\lst@numbersep%
116             \lst@numberstyle{\the\lstnumber}}}%
117   }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}%
118 \end{warpHTML}

```

---

File 84 **lwarf-longtable.sty**

§ 163 Package **longtable**

Pkg **longtable** longtable is emulated during HTML output, and the longtable package is ignored.

**for HTML output:** 1 \LWR@ProvidesPackageDrop{longtable}

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

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

instead of

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

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

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

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

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

or place it inside \warpingprintonly.

See:

<http://tex.stackexchange.com/questions/43006/why-is-input-not-expandable>

Env `longtable` \* [*horizontalignment*] {*colspec*} Emulates the longtable environment.

Per the caption package, the starred version steps the counter per caption. The unstarred version steps the counter once at the beginning, but not at each caption.

Options [c], [l], and [r] are thrown away.

```
2 \newenvironment{longtable*}[2] []{%
3 \LWR@floatbegin{table}%
4 \setcaptiontype{\LTcaptype}%
5 \caption@setoptions{longtable}%
6 \caption@setoptions{@longtable}%
7 \caption@LT@setup%
8 \booltrue{\LWR@starredlongtable}%
9 \let\captionlistentry\LWR@LTcaptionlistentry%
10 \LWR@tabular{#2}%
11 }%
12 {\endLWR@tabular\LWR@floatend}%
13 %
14 \newenvironment{longtable}[2] []{%
15 \LWR@floatbegin{table}%
16 \setcaptiontype{\LTcaptype}%
17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
```

---

```

20 \refstepcounter{\LTcaptype}%
21 \let\captionlistentry\LWR@LTcaptionlistentry%
22 \LWR@tabular{#2}
23 }
24 {\endLWR@tabular\LWR@floatend}
25

```

Provided for compatibility, but ignored:

```

26 \newcounter{LTchunksize}
27 \def\endhead{\LWR@tabularendofline}% throws away options // [dim] and /**
28 \def\endfirsthead{\LWR@tabularendofline}
29 \def\endfoot{\LWR@tabularendofline}
30 \def\endlastfoot{\LWR@tabularendofline}
31 \newcommand\tabularnewline{\LWR@tabularendofline}
32 \newcommand{\setlongtables}{}}% Obsolete command, does nothing.
33 \newlength{\LTleft}
34 \newlength{\LTright}
35 \newlength{\LTpre}
36 \newlength{\LTpost}
37 \newlength{\LTcapwidth}

38 \renewcommand*{\kill}{\LWR@tabularendofline}

```

---

#### File 85 lwarp-lscape.sty

##### § 164 Package **lscape**

Pkg **lscape** lscape is nullified.

**for HTML output:** Discard all options for lwarp-lscape.

```

1 \LWR@ProvidesPackageDrop{lscape}

2 \newenvironment*{landscape}{}{}

```

---

#### File 86 lwarp-ltcaption.sty

##### § 165 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 l warp.

longtable\* is already defined by l warp-longtable.

```
2 \newlength{\LTcapskip}
3 \newlength{\LTcapleft}
4 \newlength{\LTcapright}
5 \newcommand*{\LTcapmarginsfalse}{}

---


```

#### File 87 l warp-marginfit.sty

### § 166 Package **marginfit**

Pkg marginfit Emulated.

for HTML output: Discard all options for l warp-marginfit:

```
1 \LWR@ProvidesPackageDrop{marginfit}

---


```

#### File 88 l warp-marginfix.sty

### § 167 Package **marginfix**

Pkg marginfix Emulated.

for HTML output: Discard all options for l warp-marginfix:

```
1 \LWR@ProvidesPackageDrop{marginfix}

2 \newcommand*{\marginskip}[1] {}
3 \newcommand*{\clearmargin} {}
4 \newcommand*{\softclearmargin} {}
5 \newcommand*{\extendmargin}[1] {}
6 \newcommand*{\mparshift}[1] {}
7 \newdimen\marginheightadjustment
8 \newdimen\marginposadjustment
9 \newcommand*{\blockmargin}[1] []
10 \newcommand*{\unlockmargin}[1] []
11 \newcommand*{\marginphantom}[2] []

---


```

---

File 89 **l warp-marginnote.sty**

§ 168 Package **marginnote**

Pkg **marginnote** Emulated.

**for HTML output:** Discard all options for l warp-marginnote:

```
1 \LWR@ProvidesPackageDrop{marginnote}

2 \NewDocumentCommand{\marginnote}{o +m o}{\marginpar{#2}}
3 \newcommand*{\marginnoteleftadjust}{}%
4 \newcommand*{\marginnoterightadjust}{}%
5 \newcommand*{\marginnotetextwidth}{}%
6 \let\marginnotetextwidth\textwidth
7 \newcommand*{\marginnotevadjust}{}%
8 \newcommand*{\marginfont}{}%
9 \newcommand*{\raggedleftmarginnote}{}%
10 \newcommand*{\raggedrightmarginnote}{}%
```

---

File 90 **l warp-mcaption.sty**

§ 169 Package **mcaption**

Pkg **mcaption** mcaption is nullified.

**for HTML output:** Discard all options for l warp-mcaption:

```
1 \LWR@ProvidesPackageDrop{mcaption}

2 \newenvironment{margincap}{}{%
3 \newcommand*{\margincapalign}{}%
4 \newlength{\margincapsep}}
```

---

File 91 **l warp-mdframed.sty**

§ 170 Package **mdframed**

Pkg **mdframed** mdframed is loaded with options forced to `framemethod=none`.

## § 170.1 Package loading

**for HTML output:**

```
1 \RequirePackage{xcolor}%
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}
```

## § 170.2 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 subtitles.

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 }
```

### § 170.3 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 }
```

### § 170.4 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@htmntagc{div class="\LWR@mdthisenv" \LWR@orignewline
22 style=" \LWR@orignewline
```

Convert and print the background color:

```
23 background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```
24 border: \LWR@mdfprintlength{linewidth} solid  
25 \LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
26 border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
27 \ifbool{mdf@shadow}{%  
28     box-shadow:  
29     \LWR@mdfprintlength{shadowsize}  
30     \LWR@mdfprintlength{shadowsize}  
31     \LWR@mdfprintlength{shadowsize}  
32     \LWR@mdfprintcolor{shadowcolor} ;  
33 }  
34 {box-shadow: none ;}  
35 \LWR@orignewline  
  
36"}  
37% \LWR@htmldivclass{\LWR@mdthisenv}
```

`mdframed` environment may not work with the `HTML` versions of the following, so restore them to their originals while inside `mdframed`:

```
38 \LetLtxMacro{\hspace}{\LWR@orighspace}%  
39 \LetLtxMacro{\rule}{\LWR@origrule}%  
40 \LetLtxMacro{\makebox}{\LWR@origmakebox}%  
41 }
```

\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 }
```

## § 170.5 Titles and subtitles

\mdfframedtitleenv {*title*}

Encapsulation of the original which places the title inside a <span> of class mdframedtitle:

```
47 \LetLtxMacro{\LWR@origmdfframedtitleenv}{\mdfframedtitleenv}
48
49 \newlength{\LWR@titleroundcorner}
50
51 \renewrobustcmd{\mdfframedtitleenv[1]}{%
52 \LWR@origmdfframedtitleenv{%
```

Below, print HTML pt lengths:

```
53 \uselengthunit{PT}%
```

Open a <span> with a custom class and custom style:

```
54 \LWR@htmltagc{span class="mdframedtitle" \LWR@orignewline
55 style=" \LWR@orignewline
```

Convert and print the title background color:

```
56 background:
57 \LWR@mdfprintcolor{frametitlebackgroundcolor}
58 ; \LWR@orignewline
```

Convert and print the title rule:

```
59 \ifbool{mdf@frametitlerule}{%
60   border-bottom:
61   \LWR@mdfprintlength{frametitlerulewidth}
62   solid
63   \LWR@mdfprintcolor{frametitlerulecolor}
64   ; \LWR@orignewline
65 }{}
```

The title's top border radius is adjusted for the line width:

```
66 border-radius:
67 \setlength{\LWR@titleroundcorner}
68   {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{0pt}}
69   \rndprintlength{\LWR@titleroundcorner}
70   \rndprintlength{\LWR@titleroundcorner}
71   0pt 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%

Closes the span and unnest the paragraph tag restriction:

```
77 \LWR@htmlltagc{/span}%
78 \end{LWR@nestspan}%
79 }%
80 }
```

\LWR@mdfsubtitlecommon {<sub -or- subsub>} [<options>] {<title>}  
Common code for \LWR@mdfsubtitle and \LWR@mdfsubsubtitle.

Encapsulate the subtitle inside a <span> of class mdframedsubtitle:

```
81 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}%
82 {%
83   the following empty line is required
84 }
```

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@htmlltagc{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@htmtagc{/span}%
110 \end{[LWR@nestspan}%
111 must follow the /span or an extra <p> appears
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@mdfsubtitlecommon{subsub}%
123 }
124 \let\mdfsubsubtitle\LWR@mdfsubsubtitle
```

## § 170.6 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 \c@definecounter{#2}%
150 \expandafter\xdef\c@name the#2\endcsname{\@thmcounter{#2}}%
151 \newenvironment{#2}[1] []{%
152 \refstepcounter{#2}%
153 \ifstrempty{##1}%
154 {\let\@temptitle\relax}%
155 {%
156 \def\@temptitle{\mdf@theoremseparator%
157 \mdf@theoremspace%
158 \mdf@theoremtitlefont%
159 ##1}%
160 \mdf@thm@caption{#2}{\#4}{\c@name the#2\endcsname{##1}}%
161 }%
162 \begin{mdframed}[#1,frametitle={\strut\#4\c@name the#2\endcsname{##1}}%
163 \c@temptitle]}%
164 {\end{mdframed}}%
165 \newenvironment{#2*}[1] []{%
166 \ifstrempty{##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          \ifstrempy{##1}%
179              {\let\@temptitle\relax}%
180          {%
181              \def\@temptitle{\mdf@theoremseparator%
182                  \mdf@theoremspace%
183                  \mdf@theoremtitlefont%
184                  ##1}%
185              \mdf@thm@caption{#2}{\#4}{\csname the#2\endcsname}{##1}}%
186          }%
187          \begin{mdframed}[\#1,frametitle={\strut#4\ \csname the#2\endcsname%
188              \@temptitle}]]%\\
189          \end{mdframed}%
190      \newenvironment{#2*}[1][]{%
191          \ifstrempy{##1}%
192              {\let\@temptitle\relax}%
193          {%
194              \def\@temptitle{\mdf@theoremseparator%
195                  \mdf@theoremspace%
196                  \mdf@theoremtitlefont%
197                  ##1}%
198              \mdf@thm@caption{#2}{\#4}{\csname the#2\endcsname}{##1}}%
199          }%
200          \begin{mdframed}[\#1,frametitle={\strut#4\@temptitle}]]%\\
201          \end{mdframed}%
202      }%
203  }%
204  {##3 given -- number relationship
205  \global\@namedef{the#2}{\@nameuse{the#3}}%
206  \newenvironment{#2}[1][]{%
207      \refstepcounter{#3}%
208      \ifstrempy{##1}%
209          {\let\@temptitle\relax}%
210      {%
211          \def\@temptitle{\mdf@theoremseparator%
212              \mdf@theoremspace%
213              \mdf@theoremtitlefont%
214              ##1}%
215          \mdf@thm@caption{#2}{\#4}{\csname the#2\endcsname}{##1}}%
216      }%
217      \begin{mdframed}[\#1,frametitle={\strut#4\ \csname the#2\endcsname%
218          \@temptitle}]]%\\
219      \end{mdframed}%

```

```

220     \newenvironment{#2*}[1][]{%
221         \ifstrempty{##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}}%
226     \BeforeBeginEnvironment{#2*}{\renewcommand*{\LWR@mdthisenv}{md#2}}%
227 }%
228 }
```

\newmdtheoremenv [*mdframed-options*] envname [*numberedlike*] {*caption*} [*within*]

Modified from the original to remember the environment.

```

229 \DeclareDocumentCommand\newmdtheoremenv{O{} 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 92 lwarf-metalogo.sty

§ 171 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 93 **l warp-microtype.sty**

§ 172 Package **microtype**

Pkg **microtype** microtype is pre-loaded by l warp. All user options and macros are ignored and disabled.

**for HTML output:** Discard all options for l warp-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{\microtypsetup}{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{\lslig}{m}{#1}
23 }
24 \def\DeclareMicrotypeSet#1{\@gobbletwo}
25 \def\DeclareMicrotypeVariants#1{\@gobble}
26 \onlypreamble\DeclareMicrotypeSet
27 \onlypreamble\UseMicrotypeSet
28 \onlypreamble\DeclareMicrotypeSetDefault
29 \onlypreamble\DisableLigatures
30 \onlypreamble\DeclareMicrotypeVariants
31 \onlypreamble\DeclareMicrotypeBabelHook
```

---

File 94 **l warp-midfloat.sty**

§ 173 Package **midfloat**

Pkg midfloat midfloat is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{midfloat}
2 \newenvironment{strip}{\begingroup\def\par{\ignorespaces}\def\endpar{\endgroup}}
3 \newskip\stripsep
```

---

File 95 **l warp-moreverb.sty**

§ 174 Package **moreverb**

Pkg moreverb moreverb is supported with some patches.

for HTML output:

```
1 \begin{warpHTML}
2 \LWR@ProvidesPackagePass{moreverb}
3 \BeforeBeginEnvironment{verbatimtab}{%
4 \LWR@forcenewpage
5 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
6 }
7 \AfterEndEnvironment{verbatimtab}{%
8 \LWR@afterendverbatim%
9 }
10
11
12 \LetLtxMacro{\LWRMV@orig@verbatimtabinput}{\@verbatimtabinput}
13
14 \renewcommand{\@verbatimtabinput}[2]{%
15 \LWR@forcenewpage
16 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
17 \LWRMV@orig@verbatimtabinput[\#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 96 lwarf-morewrites.sty

§ 175 Package **morewrites**

Pkg morewrites Error if morewrites is loaded after lwarf.

Discard all options for l warp-morewrites:

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{morewrites}
2 \LWR@loadbefore{morewrites}
```

---

File 97 **l warp-mparhack.sty**

§ 176 Package **mparhack**

Pkg **mparhack** Emulated.

**for HTML output:** Discard all options for l warp-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```

---

File 98 **l warp-multicol.sty**

§ 177 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 99 **lwarp-multirow.sty**

## § 178 Package **multirow**

Pkg **multirow** multirow is emulated during HTML output, and used as-is while inside a `\teximage`.

In a `\teximage`, 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}{}  
4 \newcommand{\LWR@bigdelim}[2]{\bigdelim{#1}{#2}}
```

### § 178.1 Multirow

```
\multirow  [<vpos>] {[<numrows>]} [<bigstruts>] {[<width>]} [<fixup>] {[<text>]}
4 \RenewDocumentCommand{\multirow}{O{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@htmlltag{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@tablecols}{\theLWR@tablecolspos}%
```

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{\theLWR@tablecolspos}%
23 "%"

24 \LWR@tdstartstyles%
25 \LWR@addcmidrulewidth%
26 \LWR@addformatwpalignment%
27 \LWR@tdendstyles%
```

28 }%

The column's < spec:

29 \LWR@getexpparray{\LWR@colbeforespec}{\theLWR@tablecolspos} %

While printing the text, redefine \\ to generate a new line

```
30 \begingroup\LetLtxMacro{\\\}{\LWR@endofline}#6\endgroup%
31 \LWR@stoppars%
32 \global\boolfalse{\LWR@intabularmetadata}%
33 \renewcommand{\LWR@multirowborder}{}%
34 \LWR@traceinfo{*** multirow done}%
35 }%
```

## § 178.2 Combined multicolumn and multirow

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

`\multicolumnrow{2}{c}[c]{3}[0]{1in}[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, `\mcolrowcell` must be used for each cell of each column and each row to be skipped:
- ⚠ **empty cells**

```
... & \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 O{} m O{} m O{} +m}{%

Figure out how many extra HTML columns to add for @ and ! columns:

37 \LWR@tabularhtmlcolumns{\theLWR@tablecolspos}{#1}

Create the multicolumn/multirow tag:

---

```

38 \begingroup\LetLtxMacro{\}{\LWR@endofline}%
39 \LWR@domulticolumn[#3] [#4]{\theLWR@tabhtmlcoltotal}{#2}{#8}%
40 \endgroup%

```

Move to the next L<sup>A</sup>T<sub>E</sub>X column:

```

41 \addtocounter{LWR@tablecolspos}{#1}%
42 \addtocounter{LWR@tablecolspos}{-1}%

```

Skip any trailing @ or ! columns for this cell:

```

43 \booltrue{LWR@skipatbang}%
44 }

45 \appto{\LWR@restoreorigformatting}{%
46 \LetLtxMacro{\multirow}{\LWR@origmultirow}%
47 \renewcommand{\multicolumnrow}{\LWR@origmulticolumnrow}%
48 }

```

---

File 100 **lwarf-nameref.sty**

§ 179 Package **nameref**

Pkg **nameref** nameref is emulated by lwarf.

**for HTML output:** Discard all options for lwarf-nameref:

```

1 \typeout{Using the lwarf html version of package 'nameref' -- discarding options.}
2 \typeout{    Are not using ProvidesPackage, so that other packages}
3 \typeout{    do not attempt to patch lwarf's version of 'nameref'.}
4 \DeclareOption*{}
5 \ProcessOptions\relax

```

---

File 101 **lwarf-needspace.sty**

§ 180 Package **needspace**

Pkg **needspace** needspace is not used during HTML conversion.

**for HTML output:** Discard all options for lwarf-needspace:

```

1 \LWR@ProvidesPackageDrop{needspace}
2
3 \newcommand*{\needspace}[1]{}
4 \DeclareDocumentCommand{\Needspace}{s m}{}

```

---

File 102 **l warp-newclude.sty**

§ 181 Package **newclude**

Pkg newclude Error if newclude is loaded after l warp.

Discard all options for l warp-newclude:

for HTML output: 1 \LWR@ProvidesPackageDrop{newclude}  
2 \LWR@loadbefore{newclude}

---

File 103 **l warp-newunicodechar.sty**

§ 182 Package **newunicodechar**

Pkg newunicodechar Error if newunicodechar is loaded after l warp.

Discard all options for l warp-newunicodechar:

for HTML output: 1 \LWR@ProvidesPackageDrop{newunicodechar}  
2 \LWR@loadbefore{newunicodechar}

---

File 104 **l warp-nextpage.sty**

§ 183 Package **nextpage**

Pkg nextpage nextpage is nullified.

for HTML output: Discard all options for l warp-nextpage.

1 \LWR@ProvidesPackageDrop{nextpage}  
2 \newcommand{\cleartoevenpage}[1] {}  
3 \newcommand{\movetoevenpage}[1] {}  
4 \newcommand{\cleartooddpage}[1] {}  
5 \newcommand{\movetooddpage}[1] {}

---

File 105 **l warp-nonumonpart.sty**

§ 184 Package **nonumonpart**

Pkg nonumonpart nonumonpart is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nonumonpart}

---

File 106 **l warp-nopageno.sty**

§ 185 Package **nopageno**

Pkg nopageno nopageno is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nopageno}

---

File 107 **l warp-nowidow.sty**

§ 186 Package **nowidow**

Pkg nowidow nowidow is not used during HTML conversion.

Discard all options for l warp-nowidow:

for HTML output: 1 \LWR@ProvidesPackageDrop{nowidow}

```
\nowidow  [<lines>]
\setnowidow [<lines>]
              2 \newcommand*{\nowidow}[1] []
              3 \newcommand*{\setnowidow}[1] []

\noclub   [<lines>]
\setnoclub [<lines>]
              4 \newcommand*{\noclub}[1] []
              5 \newcommand*{\setnoclub}[1] []
```

---

File 108 **lwarf-ntheorem.sty**

## § 187 Package **ntheorem**

*(Based on original code by WOLFGANG MAY, ANDREAS SCHEDLER.)*

Pkg **ntheorem** ntheorem is patched for use by lwarf.

---

Table 12: 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.

---

### § 187.1 Limitations

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

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

### § 187.2 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 lwarf emulated `hyperref`.

**for HTML output:** Some disabled options:

```
1 \DeclareOption{thref}{}  
2  
3
```

```

4 \newbool{LWR@ntheoremmarks}
5 \boolfalse{LWR@ntheoremmarks}
6
7 \DeclareOption{thmmarks}{
8 \booltrue{LWR@ntheoremmarks}
9 \newif\ifsetendmark\setendmarktrue
10 }
11
12
13 \newbool{LWR@ntheoremamsthm}
14 \boolfalse{LWR@ntheoremamsthm}
15
16 \DeclareOption{amsthm}{\booltrue{LWR@ntheoremamsthm}}
17
18
19 \DeclareOption{amsmath}={}
20 \DeclareOption{hyperref}={}
21
22
23 \LWR@ProvidesPackagePass{ntheorem}

```

### § 187.3 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     \theorem@style{plain}
29     \renewcommand{\LWR@newtheoremstyle}{plain}% new
30   }%
31   {
32     \theorem@style{#1}
33     \renewcommand{\LWR@newtheoremstyle}{#1}% new
34   }
35 }

```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

36
37 \gdef\xnithm#1#2[#3]{%
38   \ifthm@tempif
39     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
40   \expandafter\@ifundefined{c@#1}{%

```

```
41      {\@definecounter{#1}}{}%
42      \@newctr{#1}[#3]%
43      \expandafter\xdef\csname the#1\endcsname{%
44          \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
45          {\noexpand\csname the\theoremnumbering\endcsname{#1}}{}%
46      \expandafter\gdef\csname mkheader@#1\endcsname
47          {\csname setparms@#1\endcsname
48          \@thm{#1}{#1}{#2}
49          }%
50      \global\@namedef{end#1}{\@endtheorem}
51      \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
52  \fi
53 }
54
55 \gdef\@ynthm#1#2{%
56   \ifthm@tempif
57     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
58     \expandafter\@ifundefined{c@#1}%
59       {\@definecounter{#1}}{}%
60     \expandafter\xdef\csname the#1\endcsname
61       {\noexpand\csname the\theoremnumbering\endcsname{#1}}{}%
62     \expandafter\gdef\csname mkheader@#1\endcsname
63       {\csname setparms@#1\endcsname
64       \@thm{#1}{#1}{#2}
65       }%
66     \global\@namedef{end#1}{\@endtheorem}
67     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
68  \fi
69 }
70
71 \gdef\@othm#1[#2]{#3}{%
72   \@ifundefined{c@#2}{\nocounterr{#2}}{%
73     \ifthm@tempif
74       \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
75       \global\@namedef{the#1}{\@nameuse{the#2}}%
76       \expandafter\protected\xdef\csname num@addtheoremline#1\endcsname{%
77           \noexpand@\num@addtheoremline{#1}{#3}}%
78       \expandafter\protected\xdef\csname nonum@addtheoremline#1\endcsname{%
79           \noexpand@\nonum@addtheoremline{#1}{#3}}%
80       \theoremkeyword{#3}%
81       \expandafter\protected\xdef\csname #1Keyword\endcsname
82           {\the\theoremkeyword}%
83       \expandafter\gdef\csname mkheader@#1\endcsname
84           {\csname setparms@#1\endcsname
85           \@thm{#1}{#2}{#3}
86           }%
87       \global\@namedef{end#1}{\@endtheorem}
88       \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
89  \fi}
90 }
```

## § 187.4 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@htmlltag{a id="autofloat-\arabic{LWR@thisfloat}"{} }\LWR@htmlltag{/a}%
95 \LWR@startpars%
96 }

```

## § 187.5 \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 }

```

## § 187.6 Standard styles

```

114 \renewtheoremstyle{plain}%
115   {\item[\hspace{\labelsep} \theorem@headerfont
116     \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
117   {\item[\hspace{\labelsep} \theorem@headerfont
118     \InlineClass{theoremheaderplain}{##1\ ##2\ (###3)\theorem@separator}]}%
119
120 \renewtheoremstyle{break}%
121   {\item[
122 %   \rlap{\vbox{\hbox{
123     \hspace{\labelsep} \theorem@headerfont
124     \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
125 %   }\hbox{\strut}}}}
126   ]}%
127   {\item[
128 %   \rlap{\vbox{\hbox{
129     \hspace{\labelsep} \theorem@headerfont
130     \InlineClass{theoremheaderbreak}%
131     {##1\ ##2\ (###3)\theorem@separator}\newline
132 %   }\hbox{\strut}}}}
133   ]}%
134
135 \renewtheoremstyle{change}%
136   {\item[\hspace{\labelsep}
137     \theorem@headerfont
138     \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]}%
139   {\item[\hspace{\labelsep}
140     \theorem@headerfont
141     \InlineClass{theoremheaderchange}{##2\ ##1\ (###3)\theorem@separator}]}%
142
143 \renewtheoremstyle{changebreak}%
144   {\item[
145 %   \rlap{\vbox{\hbox{
146     \hspace{\labelsep} \theorem@headerfont
147     \InlineClass{theoremheaderchangebreak}%
148     {##2\ ##1\theorem@separator}\newline
149 %   }\hbox{\strut}}}}
150   ]}%
151   {\item[
152 %   \rlap{\vbox{\hbox{
153     \hspace{\labelsep} \theorem@headerfont
154     \InlineClass{theoremheaderchangebreak}%
155     {##2\ ##1\ (###3)\theorem@separator}\newline
156 %   }\hbox{\strut}}}}
157   ]}%
158
159 \renewtheoremstyle{margin}%
160   {\item[\hspace{\labelsep}\theorem@headerfont
161     \InlineClass{theoremheadermargin}{##2 \quad ##1\theorem@separator}]}

```

```
162      ]}%
163  {\item[\hspace{\labelsep}\textnormal{\textsf{theorem@headerfont}}%
164    \textnormal{\textsf{InLineClass{theoremheadermargin}\{##2 \quad ##1\ (##3)\textnormal{@separator}}}}%
165  ]}
166
167 \renewtheoremstyle{marginbreak}%
168 {\item[\hspace{\labelsep}\textnormal{\textsf{theorem@headerfont}}%
169   \textnormal{\textsf{InLineClass{theoremheadermarginbreak}\{}}%
170   {\#2 \quad ##1\textnormal{@separator}}\newline%
171   ]}%
172 {\item[\hspace{\labelsep}\textnormal{\textsf{theorem@headerfont}}%
173   \textnormal{\textsf{InLineClass{theoremheadermarginbreak}\{}}%
174   {\#2 \quad ##1\ (##3)\textnormal{@separator}}\newline%
175   ]}%
176
177 \renewtheoremstyle{nonumberplain}%
178 {\item[\textnormal{\textsf{theorem@headerfont}}\hspace{\labelsep}%
179   \textnormal{\textsf{InLineClass{theoremheaderplain}\{##1\textnormal{@separator}}}}]%
180 {\item[\textnormal{\textsf{theorem@headerfont}}\hspace{\labelsep}%
181   \textnormal{\textsf{InLineClass{theoremheaderplain}\{##1\ (##3)\textnormal{@separator}}}}]%
182
183 \renewtheoremstyle{nonumberbreak}%
184 {\item[%
185 % \rlap{\vbox{\hbox{%
186   \hspace{\labelsep} \textnormal{\textsf{theorem@headerfont}}%
187   \textnormal{\textsf{InLineClass{theoremheaderbreak}\{##1\textnormal{@separator}}}\newline%
188 % }\hbox{\strut}}}%
189   ]}%
190 {\item[%
191 % \rlap{\vbox{\hbox{%
192   \hspace{\labelsep} \textnormal{\textsf{theorem@headerfont}}%
193   \textnormal{\textsf{InLineClass{theoremheaderbreak}\{##1\ (##3)\textnormal{@separator}}}\newline%
194 % }\hbox{\strut}}}%
195   ]}%
196
197 \renewtheoremstyle{empty}%
198 {\item[]}%
199 {\item[\textnormal{\textsf{theorem@headerfont}} \hspace{\labelsep}\relax%
200   \textnormal{\textsf{InLineClass{theoremheaderplain}\{##3}}]}%
201
202 \renewtheoremstyle{emptybreak}%
203 {\item[]}%
204 {\item[\textnormal{\textsf{theorem@headerfont}} \hspace{\labelsep}\relax%
205   \textnormal{\textsf{InLineClass{theoremheaderplain}\{##3}}] \ \newline}
```

## § 187.7 Additional objects

The following manually adjust the css for the standard configuration objects which are not a purely plain style:

```

206 \ifbool{LWR@ntheoremamsthm}{}{%
207 % upright text via CSS
208   \newtheoremstyle{plainupright}%
209   {\item[\hspace{\labelsep} \textbf{theorem@headerfont}%
210     \textit{InlineClass{theoremlheaderplain}{##1\ ##2\ theorem@separator}}]}%
211   {\item[\hspace{\labelsep} \textbf{theorem@headerfont}%
212     \textit{InlineClass{theoremlheaderplain}{##1\ ##2\ (##3)\ theorem@separator}}]}%
213
214 % upright text and small caps header via CSS
215   \newtheoremstyle{nonumberplainuprightsc}%
216   {\item[\textbf{theorem@headerfont}\hspace{\labelsep} \textit{%
217     \textit{InlineClass{theoremlheadersc}{##1\ theorem@separator}}}]%
218   {\item[\textbf{theorem@headerfont}\hspace{\labelsep} \textit{%
219     \textit{InlineClass{theoremlheadersc}{##1\ (##3)\ theorem@separator}}}}%
220 }% not amsthm

```

## § 187.8 Renewed standard configuration

The following standard configuration is renewed using the new css:

```

221 \ifbool{LWR@ntheoremamsthm}{}{%
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

```

## § 187.9 amsthm option

Only if the `amsthm` option was given:

```

248 \ifbool{LWR@ntheoremamsthm}{
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[\hspace{\labelsep}
257 % \theorem@headerfont
258   \InlineClass{theoremheaderplain}{##1\ ##2.}%
259   ]}%
260   \def\@opargbegingroup{##1##2##3{%
261 \LWR@forcenewpage% new
262     \BlockClass{theorembodyplain}% new
263     \LWR@inctheorem% new
264     \item[\hspace{\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[\hspace{\labelsep}
276 % \theorem@headerfont
277   \InlineClass{theoremheaderplain}{##1.}%
278   ]}%
279   \def\@opargbegingroup{##1##2##3{%
280 \LWR@forcenewpage% new
281     \BlockClass{theorembodyplain}% new
282     \LWR@inctheorem% new
283     \item[\hspace{\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[\hspace{-0.5em}\rule{0pt}{1.2ex}\hspace{0.5em}labelsep
295 \% \theorembodydefinition]
296 \InlineClass{\theorembodydefinition}{##1\ ##2.}
297     ]}%
298 \def\@opargbegintheorem##1##2##3{%
299 \LWR@forcenewpage% new
300     \BlockClass{\theorembodydefinition}\% new
301     \LWR@inctheorem\% new
302     \item[\hspace{-0.5em}\rule{0pt}{1.2ex}\hspace{0.5em}labelsep
303 \% \theorembodydefinition]
304 \InlineClass{\theorembodydefinition}{##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[\hspace{-0.5em}\rule{0pt}{1.2ex}\hspace{0.5em}labelsep
314 \% \theorembodydefinition]
315 \InlineClass{\theorembodydefinition}{##1.}
316     ]}%
317 \def\@opargbegintheorem##1##2##3{%
318 \LWR@forcenewpage% new
319     \BlockClass{\theorembodydefinition}\% new
320     \LWR@inctheorem\% new
321     \item[\hspace{-0.5em}\rule{0pt}{1.2ex}\hspace{0.5em}labelsep
322 \% \theorembodydefinition]
323 \InlineClass{\theorembodydefinition}{##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[\hspace{-0.5em}\rule{0pt}{1.2ex}\hspace{0.5em}labelsep
333 \% \theorembodyremark]
334 \InlineClass{\theorembodyremark}{##1\ ##2.}
335     ]}%
336 \def\@opargbegintheorem##1##2##3{%
337 \LWR@forcenewpage% new
338     \BlockClass{\theorembodyremark}\% new
```

```
339      \LWR@inctheorem% new
340      \item[\hspace{\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[\hspace{\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[\hspace{\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[\hspace{\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[\hspace{\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{\HTMLUnicod{220E}}% UTF-8 end-of-proof
400     \thm{proof}{proof}{#1}
401 }%
402 {\endtheorem}
403
404 }{}% amsthm option

```

### § 187.10 Ending a theorem

Patched for css:

```

405 \let\LWR@origendtheorem\endtheorem
406 \renewcommand{\endtheorem}{%
407 \ifbool{\LWR@ntheoremmarks}{%
408     \ifsetendmark{%
409         \InlineClass{\theoremendmark}{\csname\InTheoType Symbol\endcsname}%
410         \setendmarkfalse%
411     }%
412 }{}%
413 \LWR@origendtheorem%
414 \ifbool{\LWR@ntheoremmarks}{\global\setendmarktrue}{%
415 \endBlockClass%
416 }

```

### § 187.11 \NoEndMark

```
417 \gdef\NoEndMark{\global\setendmarkfalse}
```

### § 187.12 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@htmlelementclass{nav}{lothm}%
428 \begingroup
429 \c@tocdepth=-2%
430 \def\thm@list{#1}\thm@processlist
431 \endgroup
432 \LWR@htmlelementclassend{nav}{lothm}%
433 }
```

### § 187.13 Symbols

Proof QED symbol:

```
434 \newcommand{\qed}{\qquad\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 }
```

### § 187.14 Cross-referencing

```
\thref {\langle label \rangle}
```

```
441 \newcommand*{\thref}[1]{\cref{#1}}
```

File 109 `lwarp-overpic.sty`

## § 188 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 110 **lwarf-pagenote.sty**

## § 189 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 111 **lwarf-paralist.sty**

## § 190 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\textrm{\bfseries}#1}}
```

---

File 112 **l warp-parskip.sty**

§ 191 Package **parskip**

Pkg **parskip** **parskip** is ignored.

**for HTML output:** Discard all options for l warp-parskip.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

---

File 113 **l warp-pdflandscape.sty**

§ 192 Package **pdflandscape**

Pkg **pdflandscape** Emulated.

**for HTML output:** Discard all options for l warp-pdflandscape:

```
1 \LWR@ProvidesPackageDrop{pdflandscape}
```

---

File 114 **l warp-pdfsync.sty**

§ 193 Package **pdfsync**

Pkg **pdfsync** Emulated.

**for HTML output:** Discard all options for l warp-pdfsync:

```
1 \LWR@ProvidesPackageDrop{pdfsync}
```

```
2 \let\pdfsync\relax
3 \let\pdfsyncstart\relax
4 \let\pdfsyncstop\relax
```

---

File 115 **lwarf-pfnote.sty**

§ 194 Package **pfnote**

Pkg **pfnote** pfnote is emulated.

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

**for HTML output:** 1 \LWR@ProvidesPackageDrop{pfnote}

---

File 116 **lwarf-placeins.sty**

§ 195 Package **placeins**

Pkg **placeins** placeins is not used during HTML conversion.

Discard all options for lwarf-placeins:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{placeins}  
2 \newcommand\*{\FloatBarrier}{}

---

File 117 **lwarf-prelim2e.sty**

§ 196 Package **prelim2e**

Pkg **prelim2e** Emulated.

**for HTML output:** Discard all options for lwarf-prelim2e:

1 \LWR@ProvidesPackageDrop{prelim2e}  
2 \newcommand{\PrelimText}{}  
3 \newcommand{\PrelimTextStyle}{}  
4 \newcommand{\PrelimWords}{}

---

---

File 118 **l warp-quotchap.sty**

§ 197 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 119 **l warp-ragged2e.sty**

§ 198      Package **ragged2e**

Pkg **ragged2e** ragged2e is not used during HTML conversion.

Discard all options for l warp-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 120 **lwarf-realscripts.sty**

§ 199 Package **realscripts**

Pkg **realscripts** **realscripts** is emulated. See **lwarf.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 0{1} mm} {%
19 \LWR@setrealscriptsalign{#2}%
20 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
21 \textsuperscript{#4}\textsubscript{#3}}%
22 }%
23 }
24
25 \DeclareDocumentCommand \textsupsuperscript {s 0{1} mm} {%
26 \LWR@setrealscriptsalign{#2}%
27 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
28 \textsubscript{#4}\textsuperscript{#3}}%
29 }%
30 }
```

---

File 121 **lwarf-relsize.sty**

§ 200 Package **relsize**

Pkg **relsize** **relsize** is patched for use by **lwarf**.

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

`\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@relscaled\relscale
6 \LetLtxMacro{\LWR@origtextlarger}{\textlarger}
7 \LetLtxMacro{\LWR@origtextsmaller}{\textsmaller}
8 \let\textscale\textscale
9
10 \appto{\LWR@restoreorigformatting}{%
11 \let\relsize{\LWR@origrelsize}%
12 \LetLtxMacro{\larger}{\LWR@origlarger}%
13 \LetLtxMacro{\smaller}{\LWR@origsmaller}%
14 \let\relscale{\LWR@relscaled}%
15 \LetLtxMacro{\textlarger}{\LWR@origtextlarger}%
16 \LetLtxMacro{\textsmaller}{\LWR@origtextsmaller}%
17 \let\textscale{\LWR@textscale}%
18 }
19
20 \newcounter{LWR@relsizetemp}
21
22 \renewcommand*{\relsize}[1]{}
23 \renewcommand*{\larger}[1][]{}
24 \renewcommand*{\smaller}[1][]{}
25 \renewcommand*{\relscale}[1]{}
26
27 \renewcommand*{\textlarger}[2][1]{%
28 \setcounter{LWR@relsizetemp}{100+(#1*20)}%
29 \InlineClass{font-size:\arabic{LWR@relsizetemp}\%}{textlarger}{#2}%
30 }
31
32 \renewcommand*{\textsmaller}[2][1]{%
33 \setcounter{LWR@relsizetemp}{100-(#1*20)}%
34 \InlineClass{font-size:\arabic{LWR@relsizetemp}\%}{textsmaller}{#2}%
35 }
36
37 \renewcommand*{\textscale}[2]{%
38 \setcounter{LWR@relsizetemp}{100*\real{#1}}%

```

```
39 \InlineClass{font-size:\arabic{LWR@relsizetemp}\%}{textscale}{#2}%
40 }
```

---

File 122 **l warp-romanbar.sty**

§ 201 Package **romanbar**

Pkg romanbar romanbar is patched for use by l warp.

An inline class with an overline and underline is used.

**for HTML output:** 1 \LWR@ProvidesPackagePass{romanbar}

```
2 \DeclareRobustCommand{\Roman@bar}[1]{% #1 is in Roman, i.e. MMXII
3 \InlineClass{%
4   text-decoration: overline underline ;
5 }{romanbar}{#1}%
6 }
```

---

File 123 **l warp-romanbarpagenumber.sty**

§ 202 Package **romanbarpagenumber**

Pkg romanbarpagenumber romanbarpagenumber is ignored.

**for HTML output:** 1 \LWR@ProvidesPackageDrop{romanbarpagenumber}

---

File 124 **l warp-rotating.sty**

§ 203 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 125 **lwarf-rotfloat.sty**

## § 204 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 126 **l warp-savetrees.sty**

§ 205 Package **savetrees**

Pkg **savetrees** Emulated.

**for HTML output:** Discard all options for l warp-savetrees:

```
1 \LWR@ProvidesPackageDrop{savetrees}
```

---

File 127 **l warp-scalefnt.sty**

§ 206 Package **scalefnt**

Pkg **scalefnt** scalefnt is ignored.

**for HTML output:** 1 \LWR@ProvidesPackageDrop{scalefnt}

```
2 \DeclareRobustCommand\scalefont[1]{}
```

---

File 128 **l warp-sectsty.sty**

§ 207 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 129 **lwarf-setspace.sty**

## § 208 Package **setspace**

Pkg **setspace** setspace is not used during HTML conversion.

Discard all options for lwarf-setspace:

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{setspace}
2
3 \newcommand*{\setstretch}[1]{}
4 \newcommand*{\SetSinglespace}[1]{}
5 \newcommand*{\singespacing}{}
6 \newcommand*{\onehalfspacing}{}
7 \newcommand*{\doublespacing}{}
8
9 \newenvironment{singlespace}
10 {
11 \LWR@forcenewpage
12 \BlockClass{singlespace}
13 }
14 {\endBlockClass}
15
16 \newenvironment*singlespace*
17 {
18 \LWR@forcenewpage
19 \BlockClass{singlespace}
20 }
21 {\endBlockClass}
22
23 \newenvironment{spacing}[1]{}
24
25 }{
26
27 }
28
29 \newenvironment{onehalfspace}
30 {
```

```
31 \LWR@forcenewpage
32 \BlockClass{onehalfspace}
33 }
34 {\endBlockClass}
35
36 \newenvironment*{doublespace}
37 {
38 \LWR@forcenewpage
39 \BlockClass{doublespace}
40 }
41 {\endBlockClass}
```

---

File 130 **lwarf-shadow.sty**

§ 209 Package **shadow**

Pkg **shadow** shadow is emulated.

**for HTML output:** Discard all options for lwarf-shadow:

```
1 \LWR@ProvidesPackageDrop{shadow}

2 \newdimen\sboxsep
3 \newdimen\sboxrule
4 \newdimen\sdim
5
6 \newcommand{\shabox}[1]{%
7 \InlineClass{shabox}{#1}%
8 }
```

---

File 131 **lwarf-showidx.sty**

§ 210 Package **showidx**

Pkg **showidx** showidx is ignored.

**for HTML output:** Discard all options for lwarf-showidx:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

---

File 132 **l warp-showkeys.sty**

§ 211 Package **showkeys**

Pkg **showkeys** showkeys is ignored.

**for HTML output:** Discard all options for l warp-showkeys:

```
1 \LWR@ProvidesPackageDrop{showkeys}  
2 \NewDocumentCommand{\showkeys}{s}{}{}
```

---

File 133 **l warp-sidecap.sty**

§ 212 Package **sidecap**

Pkg **sidecap** sidecap is nullified.

**for HTML output:** Discard all options for l warp-sidecap.

```
1 \LWR@ProvidesPackageDrop{sidecap}
```

See:

[http://tex.stackexchange.com/questions/45401/  
use-the-s-star-argument-with-newdocumentenvironment](http://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment)  
regarding the creation of starred environments with xparse.

```
2 \NewDocumentEnvironment{SCtable}{soo}  
3 {\IfValueTF{#3}{\table[#3]}{\table}}  
4 {\endtable}  
5  
6 \ExplSyntaxOn  
7 \cs_new:cpx {SCtable*} {\SCtable*}  
8 \cs_new_eq:cN {\endSCtable*} \endSCtable  
9 \ExplSyntaxOff  
10  
11  
12 \NewDocumentEnvironment{SCfigure}{soo}  
13 {\IfValueTF{#3}{\figure[#3]}{\figure}}  
14 {\endfigure}  
15  
16 \ExplSyntaxOn
```

---

```

17 \cs_new:cpn {SCfigure*} {\SCfigure*}
18 \cs_new_eq:cN {\endSCfigure*} \endSCfigure
19 \ExplSyntaxOff
20
21
22 \newenvironment*{wide}{}{}

```

---

File 134 **l warp-sidenotes.sty**

§ 213 Package **sidenotes**

*(Based on original code by ANDY THOMAS, OLIVER SCHEBAUM.)*

Pkg **sidenotes** Patched for l warp.

**for HTML output:** Load the original package:

```
1 \LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with l warp:

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 l warp version of keyfloat:

```
22 \NewDocumentEnvironment{KFLTsidenotes@marginfloat}{O{-1.2ex} m}
```

```

23 {%
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{KFLT}sidenotes@marginfloat}{figure}}
33 {\end{KFLT}sidenotes@marginfloat}%
34
35 \RenewDocumentEnvironment{margintable}{o}
36 {\begin{KFLT}sidenotes@marginfloat}{table}}
37 {\end{KFLT}sidenotes@marginfloat}%

```

The following were changed by sidenotes, and now are reset back to their lwarp-supported originals:

Restoring the definition from the  $\text{\LaTeX}$  `article.cls` source:

```

38 \renewenvironment{figure*}
39           {\@dblfloat{figure}}
40           {\end@dblfloat}%
41
42 \renewenvironment{table*}
43           {\@dblfloat{table}}
44           {\end@dblfloat}%

```

File 135 `lwarp-siunitx.sty`

§ 214 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 136 **l warp-soul.sty**

§ 215 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\}{}%
```

\soo {\langle text\rangle}

Basic markup with css:

```
10 \newcommand{\soo}[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   {capspacing}{#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 \ifcsempty{#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 137 **lwarf-stabular.sty**

§ 216 Package **stabular**

Pkg **stabular** `stabular` is emulated.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{stabular}

Env stabular [vpos] {[colspec}]

2 \newenvironment{stabular}[2][c]
3 {
4 \renewcommand{\noalign}{[1]{}}
5 \begin{tabular}[#1]{#2}
6 }
7 \end{tabular}

Env stabular {[width} [vpos] {[colspec}]

8 \NewDocumentEnvironment{stabular*}{m o m}
9 {
10 \renewcommand{\noalign}{[1]{}}
11 \begin{tabular}[#2]{#3}
12 }
13 \end{tabular}

```

---

File 138 **lwarf-subfig.sty**

§ 217 Package **subfig**

(Based on original code by STEVEN DOUGLAS COCHRAN.)

Pkg **subfig** `subfig` is supported and patched by `lwarf`.

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

**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 \warp-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}%
4 \LWR@stopars%
5   \ifundefined{FBsc@max}{}%
6     {\FB@readaux{\let\FBsubheight\relax}}%
7   \tempcpta=\ne
8   \if@minipage
9     \tempcpta=z%
10    \else\ifdim\lastskip=z@\else
11      \tempcpta=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\tempcpta
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@captionadj
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\FBsubohight\relax
60 %         \box\@tempboxa
61          #4
62      \else
63 %         \vbox to \FBsubohight{\FBafil\box\@tempboxa\FBbfil}%
64          #4
65      \fi}%
66      \LWR@stoppars% new
67      \egroup
68      \ifx \empty#3\relax \else
69          \vskip\sf@capskip
70          \hrule width0pt height0pt depth0pt
71          \sf@subcaption{\#1}{\#2}{\#3}%
72          \fi
73      }%
74      \vskip\sf@bottom
75      \egroup
76      \@ifundefined{FBsc@max}{}%
77      {\addtocounter{FRobj}{-1}%
78      \ifnum\c@FRobj=0\else
79          \subfloatrowsep
80          \fi}%
81      \ifmaincaptiontop\else
82          \global\advance\@nameuse{c@\@capttype}\m@ne
83      \fi
84 \end{minipage}%
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{\@capttype}}{\the\value{\#1}}%
97      }%
98    \egroup
99  \fi
100 \bgroup
101   \ifx \relax#3\relax
102     \let\captionlabelsep=\relax
103   \fi
104 %   \setbox0\vbox{%
105 %     \hb@xt@{\the\@tempdima}{%
106 %       \hss
107 %       \parbox[t]{\the\@tempdima}{%
108 %         \caption@make
109 %           {\@nameuse{sub\@capttype name}}%
110 %           {\@nameuse{thesub\@capttype}}%
111 %           {#3}%
112 %       }%
113 %       \hss
114 %     }
115 %   }%
116   \@ifundefined{FBsc@max}%
117   { \box0 }%
118   {
119 %   \parbox[t]{\the\@tempdima}{%
120 \LWR@traceinfo{sfsubcap B1}%
121   \LWR@figcaption% new
122   \caption@make
123     {\@nameuse{sub\@capttype name}}%
124     {\@nameuse{thesub\@capttype}}%
125     {#3}%
126   \LWR@figcaption% new
127 \LWR@traceinfo{sfsubcap B2}%
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}%
151           \caption@make
152           {\@nameuse{sub@\capttype name}}%
153           {\@nameuse{thesub@\capttype}}%
154           {#3}%
155 \LWR@traceinfo{sfsubcap C1}%
156 % }%
157 %           \hss
158
159 % }
160       }%
161     }%
162   \egroup
163 \LWR@startpars%
164 }

\caption@@make  {\langle caption label\rangle} {\langle caption text\rangle}

165 \renewcommand\caption@@make[2]{%
166 \LWR@startpars%
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}%
177   \caption@applyfont
178   \caption@fmt
179   {\ifcaption@star\else
180     \begingroup
181       \captionlabelfont
182       #1%
183     \endgroup
184   \fi}%
185   {\ifcaption@star\else

```

```

186      \begingroup
187          \caption@iflf\captionlabelfont
188          \relax\caption@lsep
189      \endgroup
190  \fi}%
191 {{\captiontextfont
192     \caption@ifstrut
193     {\vrule\@height\ht\strutbox\@width\z@}%
194     {}%
195     \nobreak\hskip\z@skip % enable hyphenation
196     \caption@tfmt{#2}
197 \LWR@ensuredoingapar% new
198     \caption@ifstrut
199     {\ifhmode\@finalstrut\strutbox\fi}%
200     {}%
201     \par}}}
202 \LWR@stoppars% new
203 }

```

\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\@capttype\space
209     \ifundefined{thechapter}{}{\nameuse{thechapter}\space}%
210     \nameuse{p@sub\@capttype}%
211     \nameuse{thesub\@capttype}.)}}}

```

Patches for \subref.

\sf@subref {\langle label \rangle}

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 {\langle label \rangle}

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>]} {\langle float name \rangle}

```

216 \LetLtxMacro{\LWR@orig}{\newsubfloat}{\newsubfloat}
217
218 \def\newsubfloat[#1]{%
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 {\text} {\pagenum}
222 \renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2} }

\l@subtable {\text} {\pagenum}
223 \renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}

```

---

File 139 **l warp-supertabular.sty**

## § 218 Package **supertabular**

Pkg **supertabular** supertabular is emulated during HTML output.

**for HTML output:** 1 \LWR@ProvidesPackageDrop{supertabular}

⚠ misplaced alignment  
alignment tab character &

```

\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{mpsuptabular}{s o m}
51 {\minipage{\linewidth}\supertabular{#3}%
52 {\endsupertabular\endminipage}
```

---

File 140 **l warp-syntonly.sty**

§ 219      Package **syntonly**

Pkg    syntonly    Emulated.

**for HTML output:** Discard all options for l warp-syntonly:

```
1 \LWR@ProvidesPackageDrop{syntonly}

2 \newif\ifsyntax@
3 \syntax@false
4
5 \newcommand*{\syntaxonly}{}
6
7 \@onlypreamble\syntaxonly
```

---

**File 141 l warp-tables.sty****§ 220 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 142 l warp-tabularx.sty****§ 221 Package **tabularx****

Pkg **tabularx** tabularx is emulated by l warp.

**for HTML output:** Discard all options for l warp-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 143 **l warp-tabulary.sty**

§ 222 Package **tabulary**

Pkg **tabulary** tabulary is emulated by l warp.

**for HTML output:** Discard all options for l warp-tabulary.

Column types L, C, R, and J are emulated by l warp core code.

```
1 \LWR@ProvidesPackageDrop{tabulary}

2 \NewDocumentEnvironment{tabulary}{m o m}
3 {\begin{array}{#3}}
4 {\end{array}}
5
6 \NewDocumentEnvironment{tabulary*}{m o m}
7 {\begin{array}{#3}}
8 {\end{array}}
9
10 \newdimen\tymin
11 \newdimen\tymax
12 \def\tyformat{}

---


```

File 144 **l warp-textarea.sty**

§ 223 Package **textarea**

Pkg **textarea** textarea is emulated.

**for HTML output:** \LWR@ProvidesPackageDrop{textarea}

```
1 \newcommand\StartFromTextArea{}
2 \newcommand\StartFromHeaderArea{}
3 \newcommand*\RestoreTextArea{}
4 \newcommand*\ExpandTextArea[1][*]{}
5 \let\NCC@restoretarea\empty

---


```

---

File 145 **l warp-textcomp.sty**

§ 224 Package **textcomp**

Pkg **textcomp** **textcomp** is patched for use by **l warp**.

§ 224.1 **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.

§ 224.2 **Package loading**

**for HTML output:** 1 \LWR@ProvidesPackagePass{textcomp}

§ 224.3 **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@origtextrquill\textrquill
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}

```

## § 224.4 HTML symbols

For HTML, use HTML entities or direct Unicode, depending on the engine.

`\AtBeginDocument` improves support for `LuaLTEX` and `XLTEX`.

### § 224.4.1 pdf<sup>LT</sup>EX 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*{\texttwelvedash}{\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

```

### § 224.4.2 X<sup>LT</sup>EX and Lua<sup>LT</sup>EX 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\}{\ell}
60 \renewcommand*\{\textrquill\}{r}
61 \renewcommand*\{\textcircledP\}{\circledP}
62 \renewcommand*\{\texttwelveudash\}{\text{--}}% emdash
63 \renewcommand*\{\textthreequartersemdash\}{\text{---}}% emdash
64 \renewcommand*\{\textmho\}{\mho}
65 \renewcommand*\{\textnaira\}{\text{₦}}
66 \renewcommand*\{\textpeso\}{\text{₱}}
67 \renewcommand*\{\textrecipe\}{\text{℞}}
68 \renewcommand*\{\textinterrobangdown\}{\text{`}~\text{`}}
69 \renewcommand*\{\textpertenthousand\}{\text{‰}}
70 \renewcommand*\{\textbaht\}{\text{฿}}
71 \renewcommand*\{\textdiscount\}{\text{؎}}
72 \renewcommand*\{\textservicemark\}{\text{™}}
73 \fi

```

## § 224.5 HTML dicitrics

For HTML, Unicode diacritical marks are used:

```

74 \renewcommand*\{\capitalcedilla\}[1]{\#1\HTMLunicode{0327}}
75 \renewcommand*\{\capitalogonek\}[1]{\#1\HTMLunicode{0328}}
76 \renewcommand*\{\capitalgrave\}[1]{\#1\HTMLunicode{0300}}
77 \renewcommand*\{\capitalacute\}[1]{\#1\HTMLunicode{0301}}
78 \renewcommand*\{\capitalcircumflex\}[1]{\#1\HTMLunicode{0302}}
79 \renewcommand*\{\capitaltilde\}[1]{\#1\HTMLunicode{0303}}
80 \renewcommand*\{\capitaldieresis\}[1]{\#1\HTMLunicode{0308}}
81 \renewcommand*\{\capitalhungarumlaut\}[1]{\#1\HTMLunicode{30B}}
82 \renewcommand*\{\capitalring\}[1]{\#1\HTMLunicode{30A}}
83 \renewcommand*\{\capitalcaron\}[1]{\#1\HTMLunicode{30C}}
84 \renewcommand*\{\capitalbreve\}[1]{\#1\HTMLunicode{306}}
85 \renewcommand*\{\capitalmacron\}[1]{\#1\HTMLunicode{304}}
86 \renewcommand*\{\capitaldotaccent\}[1]{\#1\HTMLunicode{307}}

```

\textcircled becomes a span with a rounded border:

```

87 \renewcommand*\{\textcircled\}[1]{%
88 \InlineClass[border: 1px solid \LWR@currenttextcolor]{textcircled}{\#1}%
89 }%
90 }% AtBeginDocument

```

## § 224.6 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\textrquill{\LWR@origtextrquill}%
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 146 `lwarf-textpos.sty`

## § 225 Package **textpos**

Pkg `textpos` `textpos` is emulated during HTML output, and the `textpos` package is ignored.

**for HTML output:** `\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 147 **lwarf-theorem.sty**

§ 226 Package **theorem**

*(Based on original code by FRANK MITTELBACH.)*

Pkg **theorem** theorem is patched for use by lwarf.

---

Table 13: 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}

§ 226.1 **Remembering the theorem style**

Storage for the style being used for new theorems:

2 \newcommand{\LWR@newtheoremstyle}{plain}

Patched to remember the style being used for new theorems:

3 \gdef\theoremstyle#1{%
4 \@ifundefined{th@#1}{\@warning

```

5      {Unknown theoremstyle '#1'. Using 'plain'}%
6      \theorem@style{plain}%
7      \renewcommand{\LWR@newtheoremstyle}{\plain}%
8      }%
9      {%
10     \theorem@style{#1}%
11     \renewcommand{\LWR@newtheoremstyle}{#1}%
12     }%
13     \begingroup
14     \csname th@\the\theorem@style \endcsname
15     \endgroup}

```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

16 \gdef\xnthy#1#2[#3]{%
17   \expandafter\@ifdefinable\csname #1\endcsname
18   {%
19     \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}%
20     \definecounter{#1}\@newctr{#1}[#3]%
21     \expandafter\xdef\csname the#1\endcsname
22     {\expandafter\noexpand\csname the#3\endcsname
23      \@thmcntersep \@thmcnter{#1}}%
24     \def\@tempa{\global\@namedef{#1}}%
25     \expandafter\@tempa \expandafter{%
26       \csname th@\the\theorem@style
27         \expandafter\endcsname \the\theorem@bodyfont
28       \@thm{#1}{#2}}%
29     \global\expandafter\let\csname end#1\endcsname \endtheorem
30     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{\LWR@thmstyle#1}}}%
31   }%
32 }
33 \gdef@ynthy#1#2{%
34   \expandafter\@ifdefinable\csname #1\endcsname
35   {%
36     \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}%
37     \definecounter{#1}%
38     \expandafter\xdef\csname the#1\endcsname{\@thmcnter{#1}}%
39     \def\@tempa{\global\@namedef{#1}}\expandafter\@tempa
40     \expandafter{\csname th@\the\theorem@style \expandafter
41       \endcsname \the\theorem@bodyfont \@thm{#1}{#2}}%
42     \global\expandafter\let\csname end#1\endcsname \endtheorem
43     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{\LWR@thmstyle#1}}}%
44   }%
45 }
46 \gdef@othy#1[#2]{%
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}%
53       \expandafter \xdef \csname the#1\endcsname
54         {\expandafter \noexpand \csname the#2\endcsname}%
55       \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
56         \expandafter{\csname th@\the \theorem@style \expandafter
57           \endcsname \the\theorem@bodyfont \thm{#2}{#3}}%
58       \global \expandafter \let \csname end#1\endcsname \endtheorem
59       \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}%
60     }%
61   \fi}

```

## § 226.2 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%
65     \BlockClass{theorembody\LWR@thisthmstyle}%
66       \item[\hspace{-\labelsep}%
67         \InlineClass{theoremheader}{##1\ ##2}%
68       ]}%
69   \def\@opargbegintheorem##1##2##3{%
70     \LWR@forcenewpage%
71     \BlockClass{theorembody\LWR@thisthmstyle}%
72       \item[\hspace{-\labelsep}%
73         \InlineClass{theoremheader}{##1\ ##2\ (##3)}%
74       ]}%
75   }
76
77 \gdef\th@break{\% \normalfont \slshape
78   \def\@begintheorem##1##2{%
79     \LWR@forcenewpage%
80     \BlockClass{theorembody\LWR@thisthmstyle}%
81       \item[\hspace{-\labelsep}%
82         \InlineClass{theoremheader}{##1\ ##2}\newline%
83       ]}%
84   \def\@opargbegintheorem##1##2##3{%
85     \LWR@forcenewpage%
86     \BlockClass{theorembody\LWR@thisthmstyle}%
87       \item[\hspace{-\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[\hspace{\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[\hspace{\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[\hspace{\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[\hspace{\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[\hspace{\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[\hspace{\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[\hspace*{1em}\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[\hspace*{1em}\labelsep
150     \InlineClass{theoremheader}{##2 \quad ##1\ (#3)}]
151   ]}
152 }

```

Patched for css:

```
153 \gdef\@endtheorem{\endBlockClass\endtrivlist}
```

File 148 **lwarf-threeparttable.sty**

## § 227 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 {\langle text\rangle}
Prints the table note item header inside a css class of tnoteitemheader.
```

```
2 \newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}
```

Env **threeparttable** [*alignment*] To emulate threeparttable:

```
3 \newenvironment*{threeparttable}[1][b]{}{}
```

Env **tablenotes** [*options*]

```
4 \newenvironment*{tablenotes}[1]{}{}
```

```

5 {%
6 \LWR@forcenewpage
7 \BlockClass{tnotes}{%
8 \ltx@ifpackageloaded{enumitem}{%
9 \setlist[description]{format=\LWR@printtablenote}{%
10 }{}}{%
11 \description{%
12 }{%
13 {%
14 \enddescription{%
15 \endBlockClass{%
16 }

```

---

```

\tnote{<(text)>}{%
17 \newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}}

```

---

File 149 **lwarf-tikz.sty**

§ 228 Package **tikz**

Pkg **tikz** tikz is supported.

Accept all options for lwarf-tikz:

```
1 \LWR@ProvidesPackagePass{tikz}
```

**catcodes** lwarf 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, lwarf 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}%
6 {\usepackage[babel]{tikz}\booltrue{\LWR@tikzbabel}}
7 {\boolfalse{\LWR@tikzbabel}}

```

Env **tikzpicture** tikzpicture environment is enclosed inside a \latextimage. May be used as-is, and its contents will be converted to an image.

```

8 \BeforeBeginEnvironment{tikzpicture}{%
9 \latextimage%
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 \endlatextimage%
17 \ifbool{LWR@tikzbabel}{% Test for Tikz version v3.0.0
18 {}%
19 {\catcode`\$=\active}%
20 }%
21 \end{warpHTML}

```

File 150 **l warp-titleps.sty**

## § 229 Package **titleps**

Pkg **titleps** titleps is loaded and used by l warp during HTML output. All user options and macros are ignored and disabled.

Discard all options for l warp-titleps:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{titleps}

\pagestyle and \thispagestyle are already disabled in the l warp code.

```

\newpagestyle {<name>} [<style>] {[<commands>]}
2 \NewDocumentCommand{\newpagestyle}{m o m}{}{}

\renewpagestyle {<name>} [<style>] {[<commands>]}
3 \NewDocumentCommand{\renewpagestyle}{m o m}{}{}

\sethead {[<el>] [<ec>] [<er>] {[<ol>]} {[<oc>]} {[<or>]}
4 \NewDocumentCommand{\sethead}{o o m m m}{}{}

\setfoot {[<el>] [<ec>] [<er>] {[<ol>]} {[<oc>]} {[<or>]}

```

```
5 \NewDocumentCommand{\setfoot}{o o o m m m}{}  
  
\settitemarks * {\langle names\rangle}  
6 \NewDocumentCommand{\settitemarks}{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}{}  
  
\bottitemarks  
15 \newcommand*{\bottitemarks}{}  
  
\toptitemarks  
16 \newcommand*{\toptitemarks}{}  
  
 
```

```
\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 extra\rangle} [\langle which\rangle]
24 \NewDocumentCommand{\setfloathead}{s o o o m m m m}{}

\setfloatfoot * [.] [.] [.] {.} {.} {.} {\langle extra\rangle} [\langle which\rangle]
25 \NewDocumentCommand{\setfloatfoot}{s o o o m m m m}{}

\nextfloathead * [.] [.] [.] {.} {.} {.} {\langle extra\rangle} [\langle which\rangle]
26 \NewDocumentCommand{\nextfloathead}{s o o o m m m m}{}

\nextfloatfoot * [.] [.] [.] {.} {.} {.} {\langle extra\rangle} [\langle which\rangle]
27 \NewDocumentCommand{\nextfloatfoot}{s o o o m m m m}{}

\newmarkset {\langle markset\rangle}
28 \newcommand{\newmarkset}[1]{}
```

```
\newextramark * {\langle markset\rangle} {\langle macro-name\rangle}
29 \NewDocumentCommand{\newextramarkset}{s m m}{}{}

\botextramarks {\langle markset\rangle}
30 \newcommand{\botextramarks}[1] {}

\topextramarks {\langle markset\rangle}
31 \newcommand{\topextramarks}[1] {}

\firstextramarks {\langle markset\rangle}
32 \newcommand{\firstextramarks}[1] {}

\nextextramarks {\langle markset\rangle}
33 \newcommand{\nexttopextramarks}[1] {}

\outerextramarks {\langle markset\rangle}
34 \newcommand{\outerextramarks}[1] {}

\innerextramarks {\langle markset\rangle}
35 \newcommand{\innerextramarks}[1] {}
```

---

File 151 **lwarf-titleref.sty**

§ 230 Package **titleref**

Pkg titleref titleref is superceded by hyperref and nameref.

for HTML output: 1 \LWR@loadnever{titleref}{hyperref and nameref}

---

File 152 **lwarf-titlesec.sty**

§ 231 Package **titlesec**

Pkg titlesec titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for \warp-titlesec:

```
for HTML output: 1 \LWR@ProvidesPackageDrop{titlesec}

\titelabel {⟨label-format⟩}
2 \newcommand*\titlename{⟨label-format⟩}[1]{}

\titlformat* {⟨command⟩} {⟨format⟩}

\titlformat {⟨command⟩} [⟨shape⟩] {⟨format⟩} {⟨label⟩} {⟨sep⟩} {⟨begfore⟩} [⟨after⟩]
3 \newcommand\titlformat{%
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 * {⟨command⟩} {⟨left⟩} {⟨before⟩} {⟨after⟩} [⟨right⟩]
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 \oplus
18   \fontdimen\thr@@\font \ominus \fontdimen4\font}

\ttitleline * [<align>] {<material>}
19 \NewDocumentCommand{\ttitleline}{s o m}[]

\titlerule [<height>]
20 \providecommand*\titlerule{\@ifstar{\ttl@row}{\ttl@rule}}
21 \newcommand*\{\ttl@rule}[1][]{}
22 \newcommand*\{\ttl@row}[2][]{}

\iftitlemeasuring {\<true>} {\<false>}
23 \newcommand{\iftitlemeasuring}[2]{#2}

\assignpagestyle {\<command>} {\<pagestyle>}
24 \newcommand{\assignpagestyle}[2]{#2}

\ttitleclass {\<name>} [<startlevel>] {\<class>} [<cmd>]
25 \NewDocumentCommand{\ttitleclass}{m o m o}

```

---

File 153 lwarf-toc.sty

## § 232 Package **titletoc**

Pkg titletoc titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarf-toc:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{titletoc}

```

\dottedcontents {\<section>} [<left>] {\<above>} {\<label>} {\<leader>}
2 \NewDocumentCommand{\dottedcontents}{m o m m}[]

```

```
\titlecontents * {\langle section\rangle} [{\langle left\rangle}] {\langle above\rangle} {\langle numbered\rangle} {\langle numberless\rangle} {\langle filler\rangle} [{\langle below or begin\rangle}] [{\langle separator\rangle}] [{\langle end\rangle}]
  3 \newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
  4 \NewDocumentCommand{\ttl@tcstar}{m o m m m m o o o}{}%
  5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m m o}{}%

\contentsmargin [{\langle correction\rangle}] {\langle right\rangle}
  6 \newcommand{\contentsmargin}[2][]{}

\thecontentslabel
  7 \newcommand*{\thecontentslabel}{\thecontentslabel}

\thecontentspage
  8 \newcommand*{\thecontentspage}{\thecontentspage}

\contentslabel [{\langle format\rangle}] {\langle space\rangle}
  9 \newcommand{\contentslabel}[2][]{\thecontentslabel}

\contentspage [{\langle format\rangle}]
  10 \newcommand{\contentspage}[1][]{\thecontentspage}

\contentspush {\langle text\rangle}
  11 \newcommand{\contentspush}[1]{}%

\contentsuse {\langle name\rangle} {\langle text\rangle}
  12 \newcommand{\contentsuse}[2]{}%

\startcontents [{\langle name\rangle}]
  13 \newcommand*{\startcontents}[1][]{}

\stopcontents [{\langle name\rangle}]
  14 \newcommand*{\stopcontents}[1][]{}

\resumecontents [{\langle name\rangle}]
  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 154 **lwarp-titling.sty**

§ 233 Package **titling**

Pkg **titling**

**package support** l warp supports the native L<sup>A</sup>T<sub>E</sub>X 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.

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  $\Rightarrow$  \nameuse{arabic}{footnote}, or  
\thefootnote  $\Rightarrow$  \nameuse{fnsymbol}{footnote}

Redefine the footnote text:

35 \long\def\@makefntext##1{%

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

36 \makethanksmark \LWR@orighspace{1in}

\makethanksmark  $\Rightarrow$  \thanksfootmark  $\Rightarrow$  \tamark  $\Rightarrow$   
\@thefnmark  $\Rightarrow$  \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   {
51     \maketitlehooka
52     {
53       \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
54       \@bspretitle \@title \@bsposttitle
55       \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
56     }
57     \maketitlehookb
58     {
59       \begin{BlockClass}{author}
60       \renewcommand{\and}{%
61         \end{BlockClass}
62         \begin{BlockClass}{oneauthor}
63       }
64       \begin{BlockClass}{oneauthor}
65       \@bspreauthor \@author \@bspostauthor
66       \end{BlockClass}
67     }
68     \maketitlehookc
69     {
70       \begin{BlockClass}{titledate}
71       \@bspredate \@date \@bspostdate
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 \c@thanks
```

Select which kind of footnote marks to use:

```
78 \c@bsmarkseries
```

Set up special patches:

```
79 \LWR@maketitlesetup
```

Typeset the title, etc:

```
80 \c@maketitle
```

Immediately generate any \thanks footnotes:

```
81 \c@thanks
```

Reset the footnote counter:

```
82 \c@bscontmark  
83 }
```

```
\thanksmarkseries {\{series\}}
```

Sets the type of footnote marks used by \thanks, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc.

```
84 \renewcommand{\thanksmarkseries}[1]{%  
85 \def\c@bsmarkseries{\renewcommand{\thefootnote}{\c@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 155 **l warp-tocbibind.sty**

§ 234 Package **tocbibind**

Pkg **tocbibind** **tocbibind** is patched for use by **l warp**.

Opt **IndexLanguage** The **l warp** 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 **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.

**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 156 **lwarp-tocloft.sty**

§ 235 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 **LATEX** 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 {\langle style\rangle}
2 \newcommand{\tocloftpagestyle}[1]{}

\cftmarktoc
3 \newcommand*\cftmarktoc{}

\cfttoctitlefont
4 \newcommand*\cfttoctitlefont{}

\cftaftertoctitle
5 \newcommand*\cftaftertoctitle{}

6 \newlength{\cftbeforetoctitleskip}
7 \newlength{\cftaftertoctitleskip}

\cftmarklof
8 \newcommand*\cftmarklof{}

```

```
\cftloftitlefont
 9 \newcommand*{\cftloftitlefont}{}{}

\cftafterloftitle
10 \newcommand*{\cftafterloftitle}{}{}

11 \newlength{\cftbeforeloftitleskip}
12 \newlength{\cftafterloftitleskip}

\cftmarklot
13 \newcommand*{\cftmarklot}{}{}

\cftlottitlefont
14 \newcommand*{\cftlottitlefont}{}{}

\cftafterlottitle
15 \newcommand*{\cftafterlottitle}{}{}

16 \newlength{\cftbeforelottitleskip}
17 \newlength{\cftafterlottitleskip}

\cftdot
18 \newcommand*{\cftdot}{.}{}

\cftdotsep
19 \providecommand*{\cftdotsep}{1}{}

\cftnodots
20 \newcommand*{\cftnodots}{5000}{}

\cftdotfill {⟨sep⟩}
21 \providecommand{\cftdotfill}[1]{}

\cftsetpnumwidth {⟨length⟩}
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{\cftpartindent}
28 \newlength{\cftpartnumwidth}
29 \newcommand*{\cftpartfont}={}
30 \newcommand*{\cftpartpresnum}{}
31 \newcommand*{\cftpartaftersnum}{}
32 \newcommand*{\cftpartaftersnumb}{}
33 \newcommand*{\cftpartleader}{}
34 \newcommand*{\cftpartdotsep}{1}
35 \newcommand*{\cftpartpagefont}{}
36 \newcommand*{\cftpartafterpnum}{}

37 \newlength{\cftbeforechapskip}
38 \newlength{\cftchapindent}
39 \newlength{\cftchapnumwidth}
40 \newcommand*{\cftchapfont}{}
41 \newcommand*{\cftchappresnum}{}
42 \newcommand*{\cftchapaftersnum}{}
43 \newcommand*{\cftchapaftersnumb}{}
44 \newcommand*{\cftchapleader}{}
45 \newcommand*{\cftchapdotsep}{1}
46 \newcommand*{\cftchappagefont}{}
47 \newcommand*{\cftchapafterpnum}{}

48 \newlength{\cftbeforesecskip}
49 \newlength{\cftsecindent}
50 \newlength{\cftsecnumwidth}
51 \newcommand*{\cftsecfont}={}
52 \newcommand*{\cftsecpresnum}{}
53 \newcommand*{\cftsecaftersnum}{}
54 \newcommand*{\cftsecaftersnumb}{}
55 \newcommand*{\cftsecleader}{}
56 \newcommand*{\cftsecdotsep}{1}
57 \newcommand*{\cftsecpagefont}{}
58 \newcommand*{\cftsecafterpnum}{}

59 \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{\cftpaindent}
83 \newlength{\cftparanumwidth}
84 \newcommand*{\cftpafont}{}}
85 \newcommand*{\cftpapresnum}{}}
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}
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     \protect\begin{quote}#1\protect\end{quote}}%
177   }%
178 }%
179 }

```

---

File 157 **lwarf-transparent.sty**

§ 236 Package **transparent**

(Based on original code by HEIKO OBERDIEK.)

Pkg **transparent** Emulated. \texttransparent works for inline objects. \transparent only works

for \includegraphics.

⚠ Not  $\text{\LaTeX}$ : Note that transparent does not work with  $\text{\LaTeX}$ .

**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 158 `lwarp-trivfloat.sty`

### § 237 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}
```

### § 237.1 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 159 **lwarp-typearea.sty**

### § 238 Package **typearea**

Pkg typearea typearea is emulated.

**for HTML output:** 1 \LWR@ProvidesPackageDrop{typearea}

```
2 \newcommand*\typearea[2][\z@]{}
3 \newcommand*\recalctypearea(){}
4 \@ifundefined{footheight}{\newlength\footheight}{}
5 \newcommand*\areaset[3][\z@]{}
```

---

File 160 **l warp-ulem.sty**

§ 239 Package **ulem**

(Based on original code by DONALD ARSENEAU.)

Pkg **ulem** Emulated.

**for HTML output:** Original l warp definitions:

```
1 \LetLtxMacro{\LWR@ulemorig}{\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 161 **l warp-upref.sty**

§ 240 Package **upref**

Pkg **upref** Ignored.

**for HTML output:** Discard all options for l warp-upref:

```
1 \LWR@ProvidesPackageDrop{upref}
```

---

File 162 **l warp-verse.sty**

§ 241 Package **verse**

(Based on original code by PETER WILSON.)

Pkg **verse** verse is supported and patched by l warp.

**for HTML output:** Pass all options for l warp-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. l warp provides \attribution, which works for both print and HTML output. To combine the two so that \attrib is used for print and \attribution is used for HTML:

---

```
\begin{warpHTML}
\let\attrib\attribution
\end{warpHTML}
```

---

```
Len  \leftskip
Len  \leftmargini
Len  \TMLleftskip
Len  \TMLleftmargini
```

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.

`Env verse` The `verse` environment will be placed inside a `HTML pre`.

```
2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching verse.}
```

At the beginning of the `verse` environment:

```
4 \AtBeginEnvironment{verse}
5 {%
```

```
Pkg  verse   The verse or memoir packages can place stanza numbers to the left with their
Pkg  memoir  \flagverse command. Do not allow them to go into the left margin, which would
\flagverse cause pdfcrop to crop the entire page further to the left:
Len  \leftskip
     6 \ifdef{\vleftskip}{%
      7 \setlength{\vleftskip}{\HTMLvleftskip}%
      8 \setlength{\leftmargini}{\HTMLleftmargini}%
      9 }{}%
     10 \LWR@forcenewpage%
     11 \LWR@atbeginverbatim{verse}%
     12 \unskip\LWR@origvspace{-\baselineskip}%
     13 }
```

After the end of the `verse` environment, which places the `pre` tag at the regular left margin:

```
14 \AfterEndEnvironment{verse}%
15 \unskip\LWR@origvspace{-\baselineskip}%
16 \LWR@afterendverbatim%
17 }
```

Patch to place `poemtitle` inside an `HTML span` of class `poemtitle`:

---

```

18 \ifdefined{\poemtitle}{%
19   \DeclareDocumentCommand{\@vstypepoemtitle}{m}{%
20     \vspace{\beforepoemtitleskip}%
21     {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
22     \vspace{\afterpoemtitleskip}%
23   }%
24 }{}%
25
26 \LWR@traceinfo{Finished patching verse.}%
27 }% AfterEndPreamble

```

---

File 163 **lwarf-wallpaper.sty**

§ 242 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 164 **lwarf-wrapfig.sty**

§ 243 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 165 **lwarp-xcolor.sty**

## § 244 Package **xcolor**

Pkg **xcolor** **xcolor** is supported by **lwarp**.

### § 244.1 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.

### § 244.2 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 349.

`\color:`

**Print:** Used as-is.

**HTML:** Ignored by `pdftotext`, and will not appear.

**HTML `lateximage`:** Colors will appear in a `lateximage`.

`\textcolor:`

**Print:** Used as-is.

**HTML:** Redefined by `lwarp-xcolor`, page 599.

**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 599.

**HTML `lateximage`:** Remembers and reuses the print version.

\colorboxBlock:

**Print:** Becomes \colorbox.

**HTML:** Newly defined by lwarp-xcolor to use a <div>, page 600.

**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 600.

**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 601.

**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 601.

**HTML lateximage:** Uses the print version.

\boxframe:

**Print:** Used as-is.

**HTML:** Redefined by lwarp-xcolor, page 602.

**HTML lateximage:** Remembers and reuses the print version.

### § 244.3 Package loading

for HTML output: 1 \LWR@ProvidesPackagePass{xcolor}

2 \begin{warpHTML}

#### § 244.4 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 }

```

#### § 244.5 HTML color style

`\LWR@tempcolor` The color converted to HTML colorspace.

```

19 \newcommand*{\LWR@tempcolor}{}%
20 \newcommand*{\LWR@tempcolortwo}{}%

```

`\LWR@colorstyle` {*1: styletext*} {*2: model*} {*3: color*}

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 }
```

## § 244.6 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 }
```

## § 244.7 High-level macros

\color \color appears in the L<sup>A</sup>T<sub>E</sub>X PDF output, but is ignored by pdftotext and thus is ignored in the HTML file. Text styling by local group is not yet supported.

Each of the following macros is given a temporary name, and is \let to the final name once the HTML conversion starts.

\textcolor [⟨model⟩] {{color}} {{text}}

Converted into an HTML hex color span.

```
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     }%
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 {%
80     yes background color
81     \InlineClass[%
82         \LWR@borderpadding{#1}{#2} ; %
83         background:\LWR@colorstyle{#3}{#4}%
84     ]{fcolorbox}{#5}%
85 }%
86 \endgroup%
87 }
```

\fcolorboxBlock [⟨framemode⟩] {⟨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}{O{named} m O{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 {%
102     yes background color
103     \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
104     \begin{BlockClass}[%]
105         background:#\LWR@tempcolortwo; %
106         \LWR@borderpadding{#1}{#2}%
107     ]{fcolorboxBlock}
108     #5
109     \end{BlockClass}%
110 }%
111 \endgroup%
112 }
```

Creates a framed HTML <div> around its contents.

A print-output version is defined in the lwarp core: section 71

```

\lWR@subfcolorminipage {⟨framemode⟩} {⟨framecolor⟩} {⟨background tag⟩} {⟨height⟩}
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 [⟨1:framemode⟩] {⟨2:framecolor⟩} [⟨3:boxmodel⟩] {⟨4:boxcolor⟩} [⟨5:align⟩] [⟨6:height⟩]
[⟨7:inner-align⟩] {⟨8:width⟩}

121 \NewDocumentEnvironment{fcolorminipage}{O{named} m O{named} m O{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 {⟨width⟩} {⟨height⟩} {⟨depth⟩}

The depth is added to the height, but the box is not descended below by the depth.
\textrmcolor 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@currenttexcolor{} ; %
147 width:\rndprintlength{\lWR@tempwidth} ; %
148 height:\rndprintlength{\lWR@tempheight}}%

```

---

```

149 ] {boxframe}{}%
150 }%
151 }

152 \end{warpHTML}

```

---

File 166 **lwarf-xfrac.sty**

§ 245 Package **xfrac**

Pkg **xfrac** Supported by adding xfrac instances.

**for HTML output:** 1 \LWR@ProvidesPackagePass{xfrac}

**⚠ font size** In the user's document preamble, lwarf should be loaded after font-related setup. During HTML conversion, this font is used by lwarf to generate its initial PDF output containing HTML tags, later to be converted by pdftotext to a plain text file. While the text may be in any font which pdftotext can read, the math is directly converted into SVG images using this same user-selected font. xfrac below is set for the Latin Modern (lmr) font. If another font is used, it may be desirable to redefine \xfracHTMLfontsize with a different em size.

**\sfrac** [*<instance>*] [*<num>*] [*<sep>*] [*<denom>*]

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. \scalebox is made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a `lateximage`, no adjustments are necessary.

**for HTML & PRINT:** 2 \begin{warpall}

**\xfracHTMLfontsize** User-redefinable macro which controls the font size of the fraction.

3 \newcommand\*\{\xfracHTMLfontsize\}{.6em}

4 \end{warpall}

**for HTML output:** 5 \begin{warpHTML}

**font size** A span for a small font, used in the numerator and denominator:

6 \newcommand\*\{\LWR@htmlsmallfontstart\}{%

7 \LWR@htmntagc{span style="font-size:\xfracHTMLfontsize"{}%}

```

8 \LWR@nestspan%
9 %
10 }
11
12 \newcommand*{\LWR@htmlsmallfontend}{%
13 \LWR@htmlltagc{/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 167 **lwarp-xltxttra.sty**

## § 246 Package **xltxttra**

Pkg **xltxttra** **xltxttra** is emulated.

*(Based on original code by WILL ROBERTSON, JONATHAN KEW.)*

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{xltxttra}

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 168 **l warp-xmpincl.sty**

§ 247 Package **xmpincl**

Pkg xmpincl Emulated.

**for HTML output:** Discard all options for l warp-xmpincl:

```

1 \LWR@ProvidesPackageDrop{xmpincl}

2 \newcommand*\{\\include{xmp}[1]{}

```

---

File 169 **l warp-xtab.sty**

§ 248 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  
\dots  
\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 170 **lwarf-zwpagelayout.sty**

§ 249 Package **zwpagelayout**

Pkg zwpagelayout zwpagelayout is ignored.

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{zwpagelayout}

2 \def\noBboxes{}
3 @onlypreamble\noBboxes
4
5 \expandafter\ifx\csname definecolor\endcsname\relax \else
6   \definecolor{cmykblack}{cmyk}{0,0,0,1}
7   \definecolor{grblack}{gray}{0}
8 %   \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@redefineblack
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#1{\gdef\ZW@oddwarning}  
40 \def\SetEvenPageMessage#1{\gdef\Z@evenwarning}  
41 \def\ZW@oddwarning{Empty page inserted}\let\Z@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]{}  
62
```

# Change History and Index

## § 249 Change History

---

v0.10	General: 2016/03/08 Initial version . . 1	Docs: Table: Float data structures. 333	
v0.11	General: 2016/03/11 . . . . . 1	Docs: Trademarks section. . . . . 415	
	Added section: Operating-System portability. . . . . 122	Docs: Troubleshooting cross-references. . . . . 114	
	Added section: Selecting the operating system. . . . . 74	Test Suite: Assigned cleveref name for Test Float. . . . . 1	
	Test Suite: Images and index in README.txt . . . . . 1	Test Suite: Floatrow . . . . . 1	
	Test Suite: MS-Windows in README.txt . . . . . 1		
v0.12	\LWR@newhtmlfile: Bugfix: TOC with numbered files. . . . . 238	General: 2016/04/06 . . . . . 1	
	General: 2016/03/14 . . . . . 1	Added . . . . . 458	
	Global: Uses \p@{type} in float captions. . . . . 1	Ampersand (&): Fixed handling when passed as an argument. . 283	
	Test Suite: Sub-figures . . . . . 1	Docs: Added warning icons for items needing special attention. 119	
v0.13	\CaptionSeparator: Fix for newer babel package. . . . . 336	Docs: Clarify print/HTML output. 75	
	\LWR@LwarpStart: \up and \fup . . 254	Docs: Moved the supported functions table to the introduction. . . . . 40	
	General: 2016/03/24 . . . . . 1	Files: lwarp_formal.css added. . . 1	
	Removed package: subfig . . . . . 1	Fix: steps counter . . . . . 458	
	Test Suite: Ordinals, Subcaption . . 1	Fixed & handling. . . . . 456	
	tikzpicture: Fix dollar-redefined bug for newer package. . . . . 567	Test Suite: test_suite_formal.css file added. . . . . 1	
v0.14	\LWR@htmlsectionfilename: Fix: Links to home page. . . . . 207	v0.16	General: 2016/04/11 . . . . . 1
	General: 2016/03/31 . . . . . 1	\titlingpage: Improved print-output spacing. . . . . 261	
	floatrow: Added. . . . . 453	Added XeLaTeX, LuaLaTeX support. . . . . 132	
	Docs: Commands for a successful HTML conversion. . . . . 79	Docs: Font and UTF-8 support. . 72	
	Docs: Commands into a warpprint environment. . . . . 76	Docs: Moved location of \usepackage{lwarp}. . . . . 73	
	Docs: Newclude limitations. . . . 102	Docs: Text not converting. . . . . 114	
	Docs: Table: Cross-referencing data structures. . . . . 323	Fix: amsmath options clash . . . . . 136	
		Fix: newtxmath compatibility. . . . . 136	
		Lwarp no longer selects fonts. 72, 132	
		Removed package: suffix . . . . . 1	
		Test Suite: Improved titlingpage. 261	
		Test Suite: Lwarp no longer selects fonts. . . . . 1	

Test Suite: Supports XeLaTeX,	
LuaTeX. . . . .	1
xfrac: Adjusted for the use of any	
font. . . . .	604
v0.17	
\LWR@htmlsectionfilename: Fix:	
Links when entire doc is one	
HTML page. . . . .	207
General: 2016/04/14 . . . . .	1
mdframed: Added. . . . .	497
Test Suite: Fix: Print-version	
front-matter page numbers. . . . .	1
Test Suite: Mdframed . . . . .	1
v0.18	
\LWR@hspace: \hspace supported. . . . .	406
\LWR@includegraphicsb: Add: svgz	
file extension. . . . .	473
em, ex, %, px dimensions	
preserved. . . . .	473
Fix: \linewidth, \textwidth,	
\textheight inside a minipage. . . . .	473
Improved HTML output linebreaks. . . . .	473
\LWR@myshorttoc: Reorganize	
\HomeHTMLfilename logic. . . . .	340
\LWR@newhtmlfile: sidetoc after	
title, improving responsive	
design. . . . .	237
\LWR@requesttoc: Reorganize	
\HomeHTMLfilename logic. . . . .	255
\LWR@subhyperref: Improved HTML	
output linebreaks. . . . .	331
\LWR@subhyperrefclass: Improved	
HTML output linebreaks. . . . .	331
\LWR@subinlineimage: Surpress	
extra space. . . . .	332
General: 2016/05/19 . . . . .	1
File: l warp.css: Improved TOC	
outline display. . . . .	1
Files: l warp.css and	
l warp_formal.css: Improved	
responsive design. . . . .	1
Microtype disabled during HTML	
generation . . . . .	132
PDF Unicode input characters. . . . .	120
Test Suite: Verse package . . . . .	1
latextimage: pdfcrop: --hires added. . . . .	374
Reorganize \HomeHTMLfilename	
logic. . . . .	374
Surpress extra space. . . . .	374
verse: Supports verse, memoir	
packages. . . . .	592
minipage: Fix: \linewidth,	
\textwidth, \textheight	
inside a minipage. . . . .	389
v0.19	
\HTMLfilename: Docs: Escape	
filename underscores. . . . .	207
\HomeHTMLfilename: Docs: Escape	
filename underscores. . . . .	207
\LWR@l warpStart: Enabled \\ equal	
to \newline. . . . .	252
\LWR@doubledollar: MathJax	
support. . . . .	354
\LWR@filestart: l warp_mathjax.txt	
loaded. . . . .	250
\LWR@hspace: Fix: \hspace length	
computations. . . . .	406
\LWR@minipagestartpars:	
Surpresses paragraph tags	
between minipages. . . . .	405
\LWR@singledollar: MathJax	
support. . . . .	355
\LateximageFontSizeName: Add:	
User-adjustable math/latextimage	
font size. . . . .	370
\minipagefullwidth: Added: No	
width tag for the next minipage	
in HTML. . . . .	389
\rule: Added . . . . .	411
\warpHTMLonly: Added. . . . .	126
\warpprintonly: Replaces	
\rowprintedonly. . . . .	126
\xfracHTMLfontsize: Added. . . . .	603
General: 2016/06/08 . . . . .	1
Added optional args. . . . .	512
Avoids MathJax. . . . .	354
cleveref: Loaded	
\AtEndPreamble. . . . .	385
CSS for table note item. . . . .	565
Docs: Math options. . . . .	73
Docs: Table: Cross-referencing	
data structures, updated. . . . .	323
File: l warp_mathjax.txt added. . . . .	1
File: l warp.css: tnoteitemheader	
added . . . . .	1
Introduction: MathJax support	
mentioned. . . . .	39
MathJax support added. . . . .	356, 360, 362
Options: mathsvg and mathjax . . . . .	124

Supports colored \rule. . . . .	598
titleps: null \pagestyle and \thispagestyle for HTML . . .	567
tikzpicture: Adapts to tikz version.	567
equation: MathJax support. . . . .	359
v0.20	
\BlockClassSingle: Renamed from "LWR@htmldivclassline". . . . .	219
\HTMLDescription: Added \NewHTMLdescription. (Renamed in v0.30.) . . . . .	230
\HTMLFilename: No longer escape underscores. . . . .	207
\HomeHTMLFilename: No longer escape underscores. . . . .	207
\InlineClass: Renamed from "inlineclass". . . . .	220
\LWR@LwarpStart: Fix: math cross references. . . . .	254
\LWR@closeparagraph: \unskip extra spaces. . . . .	223
No break tags in the start/end of a tabular . . . . .	223
\LWR@endofline: Fix: \\ . . . . .	405
\LWR@filestart: Adds meta description. . . . .	250
\LWR@hspace: Add: Supports HTML thin breakable space. . . . .	406
\LWR@htmldivclass: Added optional style. . . . .	218
\LWR@htmlelementclass: Added optional style. . . . .	217
\LWR@htmlsectionfilename: HTMLFilename: removed additional trailing ' ', and may be empty. . . . .	207
Sections called "Index" or "index" have an underscore prepended to their filenames if no prefix. . . .	207
\LWR@includegraphicsb: Fix: \linewidth in a floatrow. . . . .	473
Fix: Expands filename. . . . .	473
\LWR@longtabledatacaptiontag: Fix: Pars in captions. . . . .	309
\LWR@section: Combined higher-level sections together into files. . . . .	243
\LWR@setOSWindows: Auto-detects operating system. . . . .	122
\LWR@subhtmlelementclass: Factored code. . . . .	217
\SetHTMLFileName: Add: Control file numbers. . . . .	207
\cpagerefFor: User-redefinable word for page references. . . . .	386
\dotfill: Inserts an ellipsis. . . . .	403
\hfill: Inserts a \quad. . . . .	403
\hrulefill: Inserts a short rule. . . . .	403
\hyperindexref: Print mode provided in case hyperref not used. . . . .	349
\pageref: Added. . . . .	330
\tracingl warp: Added. . . . .	141
General: 2017/02/09 . . . . .	1
afterpage: Added. . . . .	419
alltt: Added. . . . .	421
bookmark: Added. . . . .	428
caption and subcaption supported. . . . .	1
cleveref and referencing patches: Applied \AfterEndPreamble. . . . .	385
draftwatermark: Added. . . . .	434
eso-pic: Added. . . . .	438
everypage: Added. . . . .	438
extramarks: Added. . . . .	439
fancyhdr: Added. . . . .	444
hyperref: Additional user macros. .	480
keyfloat: Added. . . . .	486
letterspace: User-interface emulated. . . . .	488
listings: Added. . . . .	490
ltcaption: Added. . . . .	495
lwarp-newproject: Added. . . . .	149
microtype: User-interface emulated. . . . .	507
needspace: Added. . . . .	514
nowidow: Added. . . . .	516
placeins: Added. . . . .	531
ragged2e: Added. . . . .	533
setspace: Improved support. . . . .	539
textpos: Added. . . . .	560
titleps: Added. . . . .	567
titlesec: Added. . . . .	570
titletoc: Added. . . . .	572
titling: Improved compatibility. .	574
tocloft: Added. . . . .	581
wallpaper: Added. . . . .	593
wrapfig: Added. . . . .	593
Added @, <, > columns. . . . .	278

Added single-expansion data arrays. . . . .	205	\lwarpmk: Fix: \lwarpmk limages for Windows. . . . .	190
Code factored into independent \l warp_html files. . . . .	415	\lwarpmk: Fix: \lwarpmk uses lateximages text file instead of shell script. . . . .	190
Docs: Examples for generating HTML file names. . . . .	65	Add: Errors for misplaced packages. . . . .	127
Docs: Improved index. . . . .	1	Docs: Added internet class. . . . .	44
Enhanced titling support. . . . .	260	Docs: Added TeX2page, GladTeX. . . . .	44
File: lwarf.css: Minor fixes for validation. . . . .	1	Docs: Installing on Windows. . . . .	50
File: lwarpmk used to compile print, HTML, indexes, and lateximages. . . . .	1	File \l warp_tutorial.txt added. . . . .	54
Fix: \linewidth in a floatrow. . . . .	456		
Improved float caption type handling. . . . .	450		
Moved sidebar and example code to test suite. . . . .	1		
Page geometry set to 6in wide with large margins. . . . .	133		
Parallel versions of aux files for print/HTML. . . . .	1		
Removed reliance on make, grep, gawk. . . . .	1		
Tabular: \unskip extra spaces. . . . .	278		
Test Suite: HTML meta descriptions. . . . .	1		
Verbatim: Added. . . . .	445		
verbatim: Added. . . . .	271		
BlockClass: Added optional style. . . . .	219		
Renamed from "blockclass". . . . .	219		
\LWR@nestspan: Fix: Minipages inside a span. . . . .	214		
<b>v0.21</b>			
\LWR@LwarpStart: Changed lateximages to a .txt file. . . . .	252	\LWR@parsetablecols: Fix for vert bar column type. . . . .	293
\LWR@filestart: Skip title if not given. . . . .	250	\LWR@printmccoldata: Fix for vert bar column type. . . . .	305
\LWR@newhtmlfile: Skip title if not given. . . . .	237	General: 2017/03/02 . . . . .	1
\marginpar: Fixed source listing. . . . .	234		
General: 2017/02/23 . . . . .	1		
fontenc: Added. . . . .	460		
fontspec: Added. . . . .	460		
inputenc: Added. . . . .	486		
newclude: Added. . . . .	515		
newunicodechar: Added. . . . .	515		
\lwarpmk: Fix: \lwarpmk again for Windows. . . . .	190		
<b>v0.23</b>			
\LWR@parsetablecols: Fix for vert bar column type. . . . .	293		
\LWR@printmccoldata: Fix for vert bar column type. . . . .	305		
General: 2017/03/02 . . . . .	1		
<b>v0.24</b>			
\LWR@hspace: Add: \hspace \fill converts to 2em . . . . .	406		
\LWR@htmlfileref: Fix: Index links while \tracinglwarf. . . . .	326		

\hypertocfloat: List of floats	
responds to <code>lofdepth</code> ,	
<code>lotdepth</code> . . . . .	346
General: 2017/03/15 . . . . .	1
<code>floatrow</code> : Support for <code>subfig</code> . . . . .	453
<code>subfig</code> : Added. . . . .	547
<code>tikz</code> : For <code>tikz v3.0.0</code> or later,	
auto-loads <code>tikz babel</code> library if	
necessary. . . . .	566
Docs: Filename underscore. . . . .	63, 80
No longer preloads <code>subcaption</code> ;	
conflicted with <code>subfig</code> . . . . .	135
<code>picture</code> : Fix for inline images. . . . .	386
<code>tikzpicture</code> : Fix for inline images.	567
v0.25	
\LWR@loadnever: Added the ability	
to prevent conflicting packages.	128
\addcontentsline: Handles	
theorems. . . . .	338
General: 2016/03/22 . . . . .	1
<code>amsthm</code> : Added. . . . .	421
<code>ccaption</code> : Prevented. . . . .	430
<code>ellipsis</code> : Added. . . . .	435
<code>emptypage</code> : Added. . . . .	435
<code>framed</code> : Added. . . . .	463
<code>lips</code> : Added. . . . .	489
<code>mdframed</code> : Help avoid	
hyphenation. . . . .	499
<code>ntheorem</code> : Added. . . . .	517
<code>showidx</code> : Added. . . . .	540
<code>theorem</code> : Added. . . . .	561
Basic <code>L<small>A</small>T<small>E</small>X</code> theorems: improved	
css. . . . .	271
Docs: Adds credits for patched	
code. . . . .	1
Docs: Testing <code>lwarp</code> . . . . .	112
Fix: Allows <code>XE<small>A</small>T<small>E</small>X</code> and <code>L<small>A</small>T<small>E</small>X</code> to	
preload <code>graphics</code> and <code>graphicx</code> .	128
v0.26	
General: 2017/03/31 . . . . .	1
<code>cutwin</code> : Added. . . . .	432
<code>endnotes</code> : Added. . . . .	435
<code>floatflt</code> : Added. . . . .	452
<code>footmisc</code> : Added. . . . .	461
<code>footnotehyper</code> : Added. . . . .	463
<code>footnote</code> : Added. . . . .	462
<code>marginfix</code> : Added. . . . .	496
<code>marginnote</code> : Added. . . . .	497
<code>mparhack</code> : Added. . . . .	510
<code>pagenote</code> : Supported as-is. . . . .	529
sidenotes: Added. . . . .	542
1warp.css: Improved responsive	
<code>marginpar</code> and <code>marginblock</code> . . . . .	151
Docs: Improved MiK <small>T</small> <small>E</small> X install	
instructions. . . . .	47, 50
Dollar span avoided in a	
<code>lateximage</code> . . . . .	354
Footnotes now are <code>L<small>A</small>T<small>E</small>X</code> boxes	
instead of <code>pagenotes</code> . . . . .	230
lateximage: Labels track page	
numbers of <code>lateximages</code> . . . . .	374
Print mode now uses a <code>minipage</code>	
of <code>\ linewidth</code> . . . . .	374
picture: Fix for <code>\makebox</code> in	
<code>picture</code> . . . . .	386
v0.27	
\LWR@footnotetext: Fix for table	
footnote par tags. . . . .	231
General: 2017/04/04 . . . . .	1
<code>letterine</code> : Added. . . . .	488
<code>microtype</code> : Fix with <code>Xe<small>A</small>T<small>E</small>X</code> ,	
<code>L<small>A</small>T<small>E</small>X</code> . . . . .	507
<code>soul</code> : Added. . . . .	545
<code>ulem</code> : Added. . . . .	589
Docs: Installing utilities for	
MacOS. . . . .	52
Docs: Limitations of <code>saveboxes</code> . . . . .	81
Page geometry modified to reduce	
line overflow. . . . .	133
v0.28	
\@rindex: Improved indexing. . . . .	348
\HTMLAuthor: Added <code>\HTMLauthor</code> .	
(Renamed in v0.30.) . . . . .	229
\LWR@LwarpEnd: If <code>FormatEPUB</code> or	
<code>FormatWordProcessor</code> , no	
bottom nav. . . . .	256
\LWR@LwarpStart:	
<code>FormatWordProcessor</code> forces	
single-file output. . . . .	252
\LWR@filestart: Adds <code>HTML meta</code>	
<code>author</code> . . . . .	250
\LWR@forcenewpage: Forces new	
PDF page before major	
environments. . . . .	211
\LWR@htmlcomment: Breaks ligatures	
in <code>HTML</code> comments. . . . .	216
\LWR@includegraphicsb: Adapts to	
<code>graphics</code> syntax. . . . .	473

\LWR@newhtmlfile: If FormatEPUB or FormatWordProcessor: skips headers, footers, nav. . . . .	237	\HTMLDescription: Renamed from \NewHTMLdescription. . . . .	230
\LWR@parsetablecols: Added L, C, R, J column types. . . . .	293	\HTMLFirstPageTop: Renamed from \SetFirstPageTop. . . . .	228
\LWR@startref: Removed space. . . . .	328	\HTMLLanguage: Renamed from \MetaLanguage. . . . .	250
\chapter: If EPUB, prints footnotes before each section. . . . .	248	\HTMLPageBottom: Renamed from \SetPageBottom. . . . .	228
\hyperindexref: Improved indexing. . . . .	349	\HTMLPageTop: Renamed from \SetPageTop. . . . .	228
\textup: Fixed span class. . . . .	399	General: 2017/04/14 . . . . .	1
General: 2017/04/14 . . . . .	1	l warp-newproject removed, and combined with l warp. . . . .	149
glossaries: Added. . . . .	468	l warpmk: Add: xdyfile configuration option. . . . .	190
graphics: Added. . . . .	469	l warpmk: Fix: xindy and texindy adjusted for pdflatex, xelatex and lualatex. . . . .	190
tabularx: Fix for optional pos. . . . .	555	l warpmk: Fix: xindy now used for print index generation with latexmk. . . . .	190
tabulary: Added. . . . .	556	l warpmk: language now used for both index and glossary generation. . . . .	190
l warpmk: Add: printglossary and htmlglossary commands. . . . .	190	File: l warp_html.xdy renamed to l warp.xdy. . . . .	188
Added boolean FormatEPUB. . . . .	142	Fix: *.css files only written in print mode. . . . .	151
Added boolean FormatWP. . . . .	142	Fix: l warp.xdy only written in print mode. . . . .	188
Added boolean HTMLDebugComments. . . . .	142	Fix: l warp_mathjax.txt: Only written in print mode. . . . .	188
Added boolean HTMLMarkFloats, changed to WPMarkFloats as of v0.42. . . . .	143	Option OSWindows replaces macro \warpOSwindows. . . . .	125
Docs: Modfyng l warpmk and index processing. . . . .	112	Option latexmk replaces macro \UseLatexmk. . . . .	125
File l warp_mathjax.txt: Updated CDN repository. . . . .	188	Option l warpmklang changed to IndexLanguage. . . . .	124
Forced oneside to maintain large right margin. . . . .	133	Option xdyFilename added. . . . .	124
v0.29		Options HomeHTMLFilename and HTMLFilename replace macros \HomeHTMLFilename and \HTMLFilename. . . . .	125
\LWR@includegraphicsb: Fix: Error when no optional arguments. . . . .	473	v0.31	
General: 2017/04/15 . . . . .	1	General: 2017/05/15 . . . . .	1
l warpmk: Add: language option for config files. . . . .	190	keyfloat: Improved compatibility. . . . .	486
Add: l warpmklang option for l warp. . . . .	124	v0.32	
Docs: Using a glossary . . . . .	70	\RequirePackage: Fix: Ignores blanks in package list. . . . .	137
File *.l warpmkconf: Add: language option for config files. . . . .	150	General: 2016/06/09 . . . . .	1
File l warpmk.conf: Add: language option for config files. . . . .	150		
v0.30			
\CSSFilename: Renamed from \NewCSS. . . . .	229		
\HTMLAuthor: Renamed from \HTMLauthor. . . . .	229		

glossaries:	Prevent error with \glo@name not defined. . . . .	349
\lwarpmk:	Fix: io.lines() changed to file:lines() due to luatex changes. . . . .	190
v0.33		
\HTMLAuthor:	Fix: Provides empty default author if none given. . .	229
\LWR@loadbefore:	Fix: No \PackageError if already loaded. . . . .	127
\LWR@parseatcolumn:	Fix: Column alignment with leftmost @. . .	287
\LWR@tabledatasinglecolumntag:	Fix: Macros in tabular could cause extra data cell. . . . .	297
\LWR@vspace:	Add: \vspace nullified. . . . .	408
\StartDefiningTabulars:	Add: Avoids error: misplaced alignment tab character &. .	282
General:	2017/07/10 . . . . .	1
amsmath:	Removed fleqn option. .	136
fancyhdr:	Fix: Optional args for \lhead, etc. . . . .	444
Add:	Tabular at and bang columns now have their own HTML columns. . . . .	278
cleveref:	Fix: Loaded \AtEndPreamble. . . . .	385
Fix:	Incorrectly-inline math environments. . . . .	362
New handling of & to localize catcode changes. . . . .	278	
v0.34		
\fnsymbol:	Text symbols instead of math. . . . .	263
\InlineClass:	Moved optional argument in front of mandatory. .	220
\LWR@htmldivclass:	Moved optional argument in front of mandatory. . . . .	218
\LWR@htmlelementclass:	Moved optional argument in front of mandatory. . . . .	217
\LWR@htmlelementclassline:	Moved optional argument in front of mandatory. . . . .	218
\LWR@htmlspanclass:	Moved optional argument in front of mandatory. . . . .	215
\LWR@nullfonts:	Improved font control. . . . .	400
\LWR@restoreorigformatting:	booktabs: Works inside \lateximage. . . . .	349
Improved font control. . . . .	349	
\LWR@subhtmelementclass:	Moved optional argument in front of mandatory. . . . .	217
\LWR@tabledatacolumntag:	booktabs: Works inside \lateximage. . . . .	316
\fboxBlock:	Added. . . . .	395
\makebox:	Fix: Handles paren arg. .	393
General:	2017/08/08 . . . . .	1
babel-french:	Adds fixed-width HTML spaces to punctuation. .	212
balance:	Added. . . . .	426
booktabs:	Works inside \lateximage. . . . .	318, 429
boxedminipage2e:	Added. . . . .	429
boxedminipage:	Prevented. . . . .	429
crop:	Added. . . . .	431
enumerate:	Added. . . . .	436
enumitem:	Added, no longer required. . . . .	436
everyshi:	Added. . . . .	439
fancybox:	Added. . . . .	441
fancyvrb:	Added, no longer required. . . . .	444
figcaps:	Added. . . . .	450
filecontents:	Required. Patched for morewrites. . . . .	134
floatpag:	Added. . . . .	453
flushend:	Added. . . . .	459
fullpage:	Added. . . . .	466
hyperxmp:	Added. . . . .	485
idxlayout:	Added. . . . .	485
marginfit:	Added. . . . .	496
mdframed:	Improved mdtheorem patch. . . . .	504
moreverb:	Added. . . . .	508
paralist:	Added. . . . .	529
pdfescape:	Added. . . . .	530
pdfsync:	Added. . . . .	530
prelim2e:	Added. . . . .	531
rotfloat:	Added. . . . .	537

savetrees: Added.	538
shadow: Added.	540
syntonly: Added.	554
titleps: No longer required.	567
titleref: Prevented.	570
xmpincl: Added.	606
Added.	601
Docs: Horizontal space limitations.	1
Docs: Misplaced alignment character.	114
File: README.txt: updated.	1
File: l warp_mathjax.txt: Version change.	188
Fix: Added the eqnarray environments.	362
Improved font control.	398
Lists refactored to remove enumitem requirement.	273
Verbatim refactored to remove fancyvrb requirement.	269
lateximage: Fix: lateximage with minipage, \parbox, \makebox, \fbox, \framebox, \raisebox, \scalebox, \reflectbox.	374
BlockClass: Moved optional argument in front of mandatory.	219
fminipage: Added.	395
LWR@nestspan: Fix: Minipages, BlocksClass, and lists inside a span.	214
LWR@tabular: booktabs: Works inside lateximage.	319
v0.35	
General: 2017/08/08	1
Fix: \textbf and related.	398
v0.36	
\LWR@HTMLsanitize: Fix for babel-french.	370
\LWR@HTMLsanitizeexpand: Fix for babel-french.	371
\LWR@closeparagraph: Extra HTML source space after paragraphs.	223
\LWR@currenttextcolor: Fix for \rule when xcolor not loaded.	408
\LWR@footnotetext: Extra HTML source space after paragraphs. Force HTML superscripts.	231
\LWR@nullfonts: Fix: Filenames while using MathJax.	400
\LWR@restoreorigformatting:	
siunitx: Improved super/subscripts in a latexitimage.	349
\LWR@section: Improved spacing.	243
\LWR@stopars: Extra HTML source space after paragraphs.	226
\fbox: Fix: Uses \fboxrule and \fboxsep.	394
\framebox: Fix: Handles width and horiz position.	393
\makebox: Fix: Handles width and horiz position.	393
General: 2017/08/17	1
babel-french: Adjustements for French variants, load order, footnotes, ellipses.	212
footnote: Extra HTML source space after paragraphs.	462
siunitx: Fix for babel-french.	379
siunitx: Improved symbol support.	543
transparent: Added.	587
upref: Added.	591
xcolor: Added \fcolorboxBlock, \colorboxBlock.	595
xcolor: Fix: Background none in print mode.	595
xcolor: Refactored \LWR@colorstyle.	598
xcolor: Uses \fboxrule and \fboxsep.	595
xcolor: \fcolorbox etc. now work inside lateximage.	595
Docs: Reorganized: Special cases and limitations.	79
Source: Improved formatting.	1
lateximage: Footnotes appear in regular text instead of the lateximage minipage.	374
LWR@tabular: Fix for babel-french.	319
v0.37	
\include: Maintains independent .aux files for HTML.	140
General: 2017/08/19	1
\TeX accents: Added.	147
babel-french: Adjustment for load order.	212
color: Prevented.	431
comment: Maintains independent cutfiles for print, HTML.	126

siunitx: Improved symbol support.	543	Supports authblk with <div>s of class oneauthor instead of tabular.	264, 577
textcomp: Improved support.	557	\AddSubtitlePublished: Added.	265
\lwarpmk: Removes additional HTML aux files.	190	\LWR@domulticolumn: Add: Optional vpos and # rows.	307
File handles reorganized.	139	\LWR@restoreorigformatting: Appended with \appto instead of calling various macros.	349
v0.38		\LWR@tabledatacolumntag: Don't start a data cell if see \TabularMacro.	316
\@seccntformat: Added for appendix.	242	\ResumeTabular: Added.	314
\ForceHTMLPage: Added.	240	\TabularMacro: Added.	314
\ForceHTMLTOC: Added.	240	\multicolumnrow: Added.	313, 513
\lWR@section: \part* starts a new HTML page, for appendix.	243	\printauthor: Removed minipages.	260
Modified spacing, uses \numberline.	243	Supports authblk with <div>s of class oneauthor instead of tabular.	260
\numberline: Added trailing \quad.	344	\thanksmarkseries: Removed minipage footnotes.	578
\part: Fix with article class.	248	General: 2017/08/27	1
General: 2017/08/27	1	a4wide: Added.	416
appendix: Added.	425	a4: Added.	416
arabicfront: Added.	425	a5comb: Added.	416
caption2: Prevented.	430	addlines: Added.	419
chappg: Added.	431	anysize: Added.	424
color: Forces xcolor as well.	431	authblk: Added.	425
fix2col: Added.	450	bigdelim: Added.	427
fncychap: Added.	459	bigstrut: Added.	428
grffile: Added.	480	chngpage: Prevented.	431
metalogo: Added.	506	ebook: Added.	434
nonumonpart: Added.	516	fullwidth: Added.	466
nopageno: Added.	516	supertabular: Added.	553
pagenote: Option page disabled.	529	textarea: Added.	556
realscripts: Added.	534	titling: Improved compatibility.	574
relsize: Added.	535	titling: Removed extraneous center environments.	575
romanbarpagenumber: Added.	536	typearea: Added.	588
romanbar: Added.	536	xtabular: Added.	606
scalefont: Added.	538	zwpagelayout: Added.	608
siunitx: Removed from lwarpm core.	543	Add: New optional vpos argument.	512
textcomp: Removed from lwarpm core.	557	Add: Supports left/right border for bigdelim.	512
tocbibind: Added.	580	Docs: Reorganized tabular discussion.	96
xltextra: Added.	605	Fix: Long text argument.	512
\lwarpmk: Added print1 and html1 actions.	190	Titlepage \published and \subtitle removed.	
Added \markboth, \sloppy, etc.	210		
Docs: Enhanced <i>Supported Functions</i> table.	40		
Docs: Index, tocbibind.	89		
Docs: Starred sections.	86		
v0.39			
\@maketitle: titling version.	577		
Native L <sup>A</sup> T <sub>E</sub> X version.	264		
Removed minipages.	264, 577		

\AddSubtitlePublished	532
restores. ....	265
titlepage: Clear pending footnotes.	259
Removed minipages. ....	259
titlingpage: Clear pending	
footnotes. ....	575
v0.40	
{@chapcntformat: Added for	
tocbibind, anonchap. ....	243
\LWR@hline: Added. ....	318
\LWR@includegraphicsb: Add: Full	
\graphicspath support. ....	473
\LWR@nullfonts: Fix: Long	
arguments for expandable	
command. ....	400
\LWR@restoreorigformatting:	
Improved L <sup>A</sup> T <sub>E</sub> X logos inside a	
lateximage. ....	349
Improved symbols inside a	
lateximage. ....	349
Nullified \InlineClass, etc.	
inside a lateximage. ....	349
\LWR@tabledatacolumntag: Fix for	
bigdelim: \ldelem, \rdelem. . .	316
\chapter: Added support for	
quotchap. ....	248
\multicolumnrow: Fix: Adapts to	
older multirow and xparse. . .	313
\simplechapterdelim: Added for	
tocbibind, anonchap. ....	243
\underline: Added. ....	403
General: 2017/09/25	1
adjmulticol: Added. ....	418
anonchap: Added. ....	424
bigdelim: Improved	
documentation. ....	427
cuted: Added. ....	432
dblfnote: Added. ....	433
fnpos: Added. ....	460
graphics: Moved out of the lwrap	
core. ....	469
graphics: Restores	
\includegraphics and	
\DeclareGraphicsExtensions	
in a lateximage. ....	469
graphicx: Moved out of the lwrap	
core. ....	480
grffile: Directly supported. ....	480
midfloat: Added. ....	508
pfnote: Added. ....	531
quotchap: Added. ....	532
sectsty: Added. ....	538
stabular: Added. ....	547
tabls: Added. ....	555
textcomp: Additional symbols,	
improved XeLaTeX and LuaLaTeX	
support. ....	557
tocbibind: Improved for	
\simplechapter. ....	580
xtextra: Fix for \showhyphens with	
XeLaTeX. ....	605
Improved bigdelim borders. ....	512
No longer preloads xfrac. ....	136
v0.41	
\LWR@addcmidruletrim: Add:	
\cmidrule trims. ....	302
\LWR@clearmidrules: Add:	
\cmidrule trims. ....	300
\LWR@closetabledatacell: Add:	
Mute > for \bottomrule. ....	281
Fix: At/bang column with	
\multirow. ....	281
Fix: Cancel < for \multicolumn. ....	281
\LWR@domulticolumn: Add:	
\cmidrule trims. ....	307
Added vertical rules. ....	308
\LWR@nullifyNoAutoSpacing: Fix:	
\NoAutoSpacing in a tabular	
with babel-french. ....	319
\LWR@parsebarcolumn: Added	
vertical rules. ....	290
\LWR@printatbang: Add: \cmidrule	
trims. ....	296
Add: Mute at and bang columns	
for \bottomrule. ....	296
\LWR@printbartag: Added vertical	
rules. ....	296
\LWR@subaddcmidruletrim: Added. 301	
\LWR@subcmidrule: Add: \cmidrule	
trims. ....	300
\LWR@tabledatasinglecolumntag:	
Add: \cmidrule trims. ....	297
Add: Mute < for \bottomrule. . .	297
\LWR@tabularfinishrow:	
Unfinished tabular rows	
automatically filled. ....	284
\mcolrowcell: Added for	
\multicolumnrow cells. ....	318
General: 2017/10/07	1
Add: \cmidrule trims. ....	512

Added vertical rules. . . . .	512
Fix: < spec. . . . .	513
LWR@tabular: Fix: \NoAutoSpacing in a tabular with babel-french. . .	320
Improved rules. . . . .	321
v0.42	
\@ensuredmath: Improved \ensuremath. . . . .	356
\@textsubscript: Added. . . . .	402
\@textsuperscript: Added. . . . .	402
\LWR@HTMLtextstyle: Added. . . . .	398
\LWR@addformatwpalignment: If FormatWP add explicit style for cell alignment. . . . .	303
\LWR@addrulewidth: If FormatWP force explicit border. . . . .	302
\LWR@amsmathbody: Fix: Numbering and naming AMS math environments. . . . .	373
\LWR@amsmathbodynumbered: Fix: Numbering and naming AMS math environments. . . . .	373
\LWR@domulticolumn: If FormatWP add cell alignment. . . . .	308
\LWR@doubledollar: If FormatWP print LaTeX expression. . . . .	354
Improved \ensuremath. . . . .	354
Improved line spacing with mathjax. . . . .	354
\LWR@floatbegin: If FormatWP add a text frame. . . . .	334
\LWR@floatend: If FormatWP add a text frame. . . . .	335
\LWR@hline: If FormatWP force explicit border. . . . .	318
\LWR@hspace: If FormatWP add \quads. . . . .	407
\LWR@htmlmathlabel: If FormatWP print LaTeX expression. . . . .	361
\LWR@includegraphicsb: Fix: Filename expansion. . . . .	473
If FormatWP, use explicit size. . .	474
\LWR@remembertag: Fix: Numbering and naming AMS math environments. . . . .	372
\LWR@restoreorigformatting: Improved \ensuremath. . . . .	349
\LWR@rule: If FormatWP add \quads. . . . .	410
\LWR@singledollar: If FormatWP print LaTeX expression. . . . .	355
\LWR@subaddcmidruletrim: Opt if no rule given. . . . .	301
\LWR@tabledatasinglecolumntag: If FormatWP add cell alignment. .	298
\LaTeX: If FormatWP use explicit style. . . . .	412
\TeX: If FormatWP use explicit style.	412
\listoffigures: Added boolean WPMarkLOFT. . . . .	342
\listoftables: Added boolean WPMarkLOFT. . . . .	342
\marginpar: If FormatWP emulate a wrapfig. . . . .	234
\tableofcontents: Added boolean WPMarkTOC. . . . .	342
\underline: If FormatWP, use explicit styles for \underline, etc. . . . .	403
General: 2017/10/30 . . . . .	1
\textbf and related: If FormatWP, use explicit styles for \textsc, etc. . . . .	398
algorithmic: If FormatWP add \quads. . . . .	420
epigraph: If FormatWP add HTML styles. . . . .	437
fancybox: If FormatWP add HTML styles. . . . .	441
floatfl: Added width. . . . .	452
includegraphics: Fix: Class key. .	473
keyfloat: If FormatWP add explicit HTML style. . . . .	487
moreverb: Simplified formatting of listings. . . . .	508
morewrites: Added. . . . .	509
multirow: If FormatWP add cell alignment. . . . .	512
overpic: Added. . . . .	528
realscripts: Fix for subscripts in a lateimage. . . . .	534
sidenotes: If FormatWP add explicit HTML style. . . . .	542
siunitx: Improved \ensuremath. .	544
soul: If FormatWP, add explicit styles. . . . .	545
textcomp: Improved \interrobangdown. . . . .	557

wrapfig: If FormatWP add explicit	
HTML style. . . . .	594
Added boolean WPMarkLOFT. . . . .	143
Added boolean WPMarkMath. . . . .	144
Added boolean	
WPMarkMinipages. . . . .	143
Added boolean WPMarkTOC. . . . .	143
Added boolean WPTitleHeading. . . . .	144
Docs: Added support page. . . . .	2
Docs: Improper \prevdepth. . . . .	114
Docs: Reorganized math	
limitations . . . . .	90
File: l warp_mathjax.txt:	
Updated siunitx script. . . . .	188
Fix: Numbering and naming AMS	
math environments. . . . .	372
If FormatWP, shift section	
headings. . . . .	144
tabbing: Added. . . . .	271
latextimage: Fix: Numbering and	
naming AMS math	
environments. . . . .	374
center: If FormatWP use explicit	
text-align. . . . .	377
minipage: Added boolean	
WPMarkMinipages. . . . .	391, 392
If FormatWP add a text frame. . . . .	390
eqnarray: Fix: Numbering and	
naming AMS math	
environments. . . . .	363
If FormatWP print LaTeX	
expression. . . . .	362
equation: If FormatWP print LaTeX	
expression. . . . .	359
LWR@BlockClassWP: Added to factor	
code. . . . .	220
LWR@figcaption: If FormatWP forces	
italic captions. . . . .	337
LWR@tabular: If FormatWP force	
explicit border. . . . .	321

## § 249 Index

---

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols	
\\$ . . . . .	<u>354</u>
\& . . . . .	<u>206</u> , <u>5002</u>
\( . . . . .	<u>6556</u>
\) . . . . .	<u>6556</u>
\, . . . . .	<u>79</u>
\@csetcpageref . . . . .	<u>7276</u>
\@csetcref . . . . .	<u>7272</u>
\@csetcrerange . . . . .	<u>7273</u>
\@author . . . . .	<u>258</u>
\@begintheorem . . . . .	<u>4832</u>
\@capttype . . . . .	<u>6129</u>
\@chapcntformat . . . . .	<u>4197</u>
\@currentlabelname . . . . .	<u>5902</u>
\@date . . . . .	<u>258</u>
\@dldbfloat . . . . .	<u>6105</u>
\@endtheorem . . . . .	<u>4842</u>
\@ensuredmath . . . . .	<u>6563</u>
\@float . . . . .	<u>6105</u>
\@fnssymbol . . . . .	<u>4642</u>
\@footnotetext . . . . .	<u>3960</u>
\@include . . . . .	<u>457</u>
\@makecaption . . . . .	<u>6131</u>
\@makefnmark . . . . .	<u>3932</u>
\@makefntext . . . . .	<u>3931</u>
\@maketitle . . . . .	<u>49</u> , <u>4665</u>
\@mpfootnotetext . . . . .	<u>3961</u>
\@opargbegintheorem . . . . .	<u>4837</u>
\@secCntformat . . . . .	<u>4195</u>
\@starttoc . . . . .	<u>6245</u>
\@textsubscript . . . . .	<u>7659</u>
\@textsuperscript . . . . .	<u>7655</u>
\@title . . . . .	<u>258</u>
\@wrglossary . . . . .	<u>6409</u>
\@wrindex . . . . .	<u>6402</u>
\\" . . . . .	<u>405</u>
\\$ . . . . .	<u>354</u>
\\$\\$ . . . . .	<u>354</u>

~	79	\backmatter	4179
		balance (package)	426
		\BaseJobname	3349
		BaseJobname (option)	124
		baseline	
		tabular	292
		\bfseries	7644
		\BibTeX	7881
		bigdelim (package)	426
		bigstrut (package)	428
		BlockClass (environment)	3680
		\BlockClassSingle	3692
		bookmark (package)	428
		booktabs (package)	429
		booleans:	
		CombineHigherDepths	64, 235
		FileSectionNames	65, 207
		FormatEPUB	103, 142
		FormatWP	105, 142
		HTMLDebugComments	142
		LWR@amsmultiline	360
		LWR@doingapar	222
		LWR@doingcmidrule	280
		LWR@doinghline	279
		LWR@doingstartpars	222
		LWR@doingtbrule	280
		LWR@emptyatbang	280
		LWR@exittingtabular	315
		LWR@freezethisfloat	335
		LWR@infloatrow	472
		LWR@intabularmetadata	280
		LWR@isstartingequation	372
		LWR@minipagefullwidth	389
		LWR@minipagethispar	389
		LWR@skipatbang	280
		LWR@skippingmcolrowcell	280
		LWR@skippingmrowcell	280
		LWR@starredlongtable	309
		LWR@startedrow	279
		LWR@tableparcell	280
		LWR@tabularmutemods	315
		LWR@tracinglwarp	141
		LWR@validtablecol	291
		LWR@verbtags	269
		mathjax	123
		usingOSWindows	122
		warpingHTML	123
		warpingprint	123
		WPMarkFloats	107, 142
		WPMarkLOFT	108, 143
		B	
		babel (package)	102

WPMMarkMath . . . . .	108, 144	LWR@midrulecounter . . . . .	299	
WPMMarkMinipages . . . . .	107, 143	LWR@minipagedepth . . . . .	387	
WPMMarkTOC . . . . .	107, 143	LWR@nextautofloat . . . . .	336	
WPTitleHeading . . . . .	108, 144	LWR@nextautopage . . . . .	336	
boxedminipage (package) . . . . .	429	LWR@nextequation . . . . .	356	
boxedminipage2e (package) . . . . .	429	LWR@prevFileDepth . . . . .	242	
\boxframe . . . . .	137	LWR@spandepth . . . . .	222	
bugs . . . . .	114	LWR@startingequation . . . . .	371	
BVerbatim (environment) . . . . .	183	LWR@tablecolspos . . . . .	286	
<b>C</b>				
calc (package) . . . . .	135	LWR@tablecolswidth . . . . .	286	
Calibre . . . . .	103	LWR@tabletotalcols . . . . .	287	
caption (package) . . . . .	135	LWR@tabletotalcolsnext . . . . .	287	
caption2 (package) . . . . .	430	LWR@thisfloat . . . . .	335	
\caption@begin . . . . .	6174	SideTOCDepth . . . . .	64, 343	
\caption@end . . . . .	6174	tocdepth . . . . .	63	
\captionlistentry . . . . .	6178	\cpagerefFor . . . . .	7275	
\captionof . . . . .	6208	crop (package) . . . . .	431	
\CaptionSeparator . . . . .	6130	cross-references		
ccaption (package) . . . . .	430	missing or incorrect . . . . .	114	
center (environment) . . . . .	7109	CSS		
\centering . . . . .	254	file selection . . . . .	78	
changepage (package) . . . . .	430	lwarf.css . . . . .	78	
chappg (package) . . . . .	431	per HTML page . . . . .	78	
\chapter . . . . .	4344	project-specific changes . . . . .	78	
chngpage (package) . . . . .	431	\CSSFilename . . . . .	62, 65, 78, 3902	
\citetitle . . . . .	4745	cuted (package) . . . . .	432	
class:		cutwin (package) . . . . .	432	
internet . . . . .	44	<b>D</b>		
cleveref (package) . . . . .	384	danger icon . . . . .	119	
cmap (package) . . . . .	73	\date . . . . .	77	
\color . . . . .	599	dblfnote (package) . . . . .	433	
color (package) . . . . .	431	dcolumn (package) . . . . .	433	
\colorbox . . . . .	45	debugging . . . . .	114	
\colorboxBlock . . . . .	55, 7168	HTML debug comments . . . . .	142	
CombineHigherDepths (boolean) . . . . .	64, 235	\DeclareGraphicsExtensions . . . . .	2	
comment (package) . . . . .	126	Deja Vu . . . . .	72	
Computer Modern . . . . .	72	description		
\ConTeXt . . . . .	7878	HTML meta tag . . . . .	77, 230	
counters:		description (environment) . . . . .	4896	
FileDepth . . . . .	64, 235	displaymath (environment) . . . . .	6577	
lofdepth . . . . .	345	\dotfill . . . . .	7671	
lotdepth . . . . .	345	draftwatermark (package) . . . . .	434	
LWR@externalfilecnt . . . . .	354	<b>E</b>		
LWR@htmlfilenumber . . . . .	207	ebook (package) . . . . .	434	
LWR@lateautopage . . . . .	336	ellipsis (package) . . . . .	435	
LWR@lateimagedepth . . . . .	370	\emph . . . . .	7545	
LWR@lateimagenumber . . . . .	369	emptypage (package) . . . . .	435	
LWR@LIPage . . . . .	370	\end@dblfloat . . . . .	6121	

\end@float . . . . .	6121	verse . . . . .	2
\EndDefiningTabulars . . . . .	4996	warpall . . . . .	75, 100
endnotes (package) . . . . .	435	warpHTML . . . . .	50, 67, 75, 101
\enlargethispage . . . . .	7783	warpprint . . . . .	66, 75, 101
\enskip . . . . .	405, 7736	epigraph (package) . . . . .	437
enumerate (environment) . . . . .	4868	EPUB	
enumerate (package) . . . . .	436	conversion software . . . . .	103
enumitem (package) . . . . .	436	HTML conversion settings . .	103, 142
environ (package) . . . . .	135	eqnarray (environment) . . . . .	6680
environments:		equation	
abstract . . . . .	4728	miss-numbered . . . . .	93, 517
align . . . . .	6825	equation (environment) . . . . .	6624
align* . . . . .	6848	error messages . . . . .	114
BlockClass . . . . .	3680	eso-pic (package) . . . . .	438
BVerbatim . . . . .	183	etoolbox (package) . . . . .	121
center . . . . .	7109	everyhook (package) . . . . .	134
description . . . . .	4896	everypage (package) . . . . .	438
displaymath . . . . .	6577	everyshi (package) . . . . .	439
enumerate . . . . .	4868	expl3 (package) . . . . .	134
eqnarray . . . . .	6680	export	
equation . . . . .	6624	to word processor . . . . .	105
fcolorminipage . . . . .	121, 7210	extramarks (package) . . . . .	439
flalign . . . . .	6871		F
flalign* . . . . .	6894	fancybox (package) . . . . .	440
flushleft . . . . .	7125	fancyhdr (package) . . . . .	444
flushright . . . . .	7117	fancyvrb (package) . . . . .	444
fminipage . . . . .	82, 7488, 7536	FAQ . . . . .	114
gather . . . . .	6779	fbox	
gather* . . . . .	6802	around a minipage . . . . .	82
itemize . . . . .	4854	\fbox . . . . .	82, 7471
lateximage . . . . .	369, 7027	\fboxBlock . . . . .	82, 7480, 7535
longtable . . . . .	2	\fcolorbox . . . . .	67, 7174
LWR@BlockClassWP . . . . .	3708	\fcolorboxBlock . . . . .	87, 7209
LWR@figcaption . . . . .	6140	fcolorminipage (environment) .	121, 7210
LWR@nestspan . . . . .	3524	figcaps (package) . . . . .	450
LWR@tabular . . . . .	5794	file	
LWRprint@fminipage . . . . .	7512	inaccessible . . . . .	64
math . . . . .	6576	underscore . . . . .	63, 80
minipage . . . . .	7338	filecontents (package) . . . . .	134
multiline . . . . .	6730	FileDepth (counter) . . . . .	64, 235
multiline* . . . . .	6754	files:	
picture . . . . .	386, 7293	glyptounicode . . . . .	73
quote . . . . .	4753	lwarf.css . . . . .	78, 151
tabbing . . . . .	4820	lwarf.xdy . . . . .	78, 188
theindex . . . . .	6380	lwarf_formal.css . . . . .	183
tikzpicture . . . . .	8	lwarf_mathjax.txt . . . . .	188
titlepage . . . . .	76, 4569	lwarf_sagebrush.css . . . . .	178
titlingpage . . . . .	14, 76	lwarf_tutorial.txt . . . . .	54
Verbatim . . . . .	13	lwarfpmk.conf . . . . .	149
verbatim . . . . .	4810		

lwarpmk.lua	112	\framebox	7450
project.css	78	framed (package)	463
project.lwarpmkconf	150	framed objects	82
project_html.tex	149	FrameMaker (program)	46
sample_project.css	78, 187	Frequently Asked Questions	114
tutorial.tex	54	\frontmatter	4176
FileSectionNames (boolean)	65, 207	\fnright (package)	466
fix2col (package)	450	fullpage (package)	466
\flagverse	592	fullwidth (package)	466
flalign (environment)	6871	\fup	7663
flalign* (environment)	6894	\fussy	3445
Flare (program)	46		
float (package)	450	<b>G</b>	
floatflt (package)	452	gather (environment)	6779
\floatname	9	gather* (environment)	6802
floatpag (package)	453	GELLMU (program)	44
\floatplacement	12	generator	
floatrow (package)	453	HTML meta tag	251
\floatstyle	15	geometry (package)	133, 467
\flushbottom	3443	gettitlestring (package)	134
flushend (package)	459	GladTeX (program)	45
flushleft (environment)	7125	glossaries (package)	88, 467
flushright (environment)	7117	glossary	
fminipage (environment)	82, 7488, 7536	language	70
fnchap (package)	459	processing	70
fnpos (package)	460	glyptounicode (file)	73
font		graphics (package)	469
bitmapped	72	graphicx (package)	480
Computer Modern	72	grffile (package)	480
Deja Vu	72		
ligatures	72, 73	<b>H</b>	
packages	72	heading, word processor	108
selection	72	Hevea (program)	44
size — lateximage	91, 352, 369	\hfill	7669
size — math, SVG	91, 352, 369	\HomeHTMLFilename	3351
size — xfrac	603	HomeHTMLFilename (option)	63, 125
fontenc (package)	72, 460	horizontal space	79
fontspec (package)	72, 460	between minipages	405
footmisc (package)	461	\href	6045
footnote (package)	462	\hrulefill	7670
footnotehyper (package)	463	\hskip	79
footnotes	230	\hspace	79, 405, 7774
for		HTML	
HTML & PRINT	119	commands for a successful conver-	
HTML output	119	sion	79
PRINT output	119	conversion settings	63
\ForceHTMLPage	87, 4154	debug comments	142
\ForceHTMLTOC	87, 4160	EPUB	103, 142
FormatEPUB (boolean)	103, 142	word processor	105, 142, 143
FormatWP (boolean)	105, 142	filename generation	65

headings . . . . .	118
in place of SVG images . . . . .	60, 115
meta tag	
author . . . . .	77, 229
description . . . . .	77, 230
generator . . . . .	251
viewport . . . . .	251
tabular column conversion . . . . .	292
\HTMLAuthor . . . . .	66, 77, 3916
HTMLDebugComments (boolean) . . . . .	142
\HTMLDescription . . . . .	66, 77, 3922
\HTMLEntity . . . . .	3332
\HTMLFilename . . . . .	3350
HTMLFilename (option) . . . . .	63, 125
\HTMLFirstPageTop . . . . .	66, 3879
htmlglossary (option) [lwarpmk] . . . . .	89, 468
\HTMLLanguage . . . . .	66, 4403
HTMLleftmargini (length) . . . . .	101, 268, 269, 592
\HTMLPageBottom . . . . .	66, 258, 3895
\HTMLPageTop . . . . .	66, 3887
\HTMLUnicode . . . . .	3340
HTMLvleftskip (length) . . . . .	101, 268, 592
\hyperindexref . . . . .	6416
hyperref (package) . . . . .	330, 480
\hypertoc . . . . .	6326
\hypertocfloat . . . . .	6339
hyperxmp (package) . . . . .	485
<b>I</b>	
icon	
warning . . . . .	119
idxlayout (package) . . . . .	485
ifplatform (package) . . . . .	121
images	
graphicx package . . . . .	379
in strange places . . . . .	115
showing as HTML . . . . .	60, 115
\include . . . . .	457
\includegraphics . . . . .	228
indentfirst (package) . . . . .	486
InDesign (program) . . . . .	46
index	
language . . . . .	70
placement and toc options . . . . .	89
processing . . . . .	58, 59
UTF-8 . . . . .	73
IndexLanguage (option) . . . . .	70, 89, 124, 468, 579
\InlineClass . . . . .	3700
inputenc (package) . . . . .	72, 486
internet (class) . . . . .	44
<b>J</b>	
item	
empty . . . . .	80, 272
itemize (environment) . . . . .	4854
\itshape . . . . .	7649
<b>K</b>	
JavaScript	
MathJax . . . . .	90, 351
<b>L</b>	
label	
in HTML . . . . .	254
math environment . . . . .	361
language	
glossary . . . . .	70
index . . . . .	70
language HTML metadata . . . . .	250
\LaTeX . . . . .	7854
\LaTeX2HTML (program) . . . . .	44
\LaTeXe . . . . .	7854
lateximage (environment) . . . . .	369, 7027
\LateximageFontSizeName . . . . .	6928
lateximages	
font size . . . . .	91, 352, 369
latexmk (option) . . . . .	63, 125
\TeX (program) . . . . .	44
layout (package) . . . . .	488
lengths:	
HTMLleftmargini . . . . .	101, 268, 269, 592
HTMLvleftskip . . . . .	101, 268, 592
LWR@minipageheight . . . . .	388
LWR@minipagewidth . . . . .	387
LWR@tempheight . . . . .	406
LWR@tempraise . . . . .	406
LWR@tempwidth . . . . .	406
\LWR@cmidrulewidth . . . . .	299
\LWR@heavyrulewidth . . . . .	299
\LWR@lightrulewidth . . . . .	299
\LWR@thiscmidrulewidth . . . . .	300
\VerbatimHTMLWidth . . . . .	269
vleftmargini . . . . .	101, 268, 592
vleftskip . . . . .	101, 268, 592
letltxmacro (package) . . . . .	121
letterspace (package) . . . . .	488
lettrine (package) . . . . .	488

LibreOffice	
conversion recommendations	108
import into	105
section headings	108
LibreOffice (program)	45
ligatures	72, 73, 133
line numbers	119
\linebreak	7777
\LinkHome	3401
Linux (program)	74, 122
lips (package)	489
list item, empty	80, 272
listings (package)	490
\listof	6292
\listoffigures	6266
\listoftables	6279
lmodern (package)	72
lofdepth (counter)	345
longtable (environment)	2
longtable (package)	493
lotdepth (counter)	345
lscape (package)	495
ltcaption (package)	495
LuaLaTeX	
detection	132
file & section names	240
\LuaLaTeX	7872
LuaLaTeX (program) [requirement]	48
\LuaTeX	7872
l warp	
loading	73
options	73
l warp (package)	73
l warp.css (file)	78, 151
l warp.xdy	
customizing	78
l warp.xdy (file)	78, 188
l warp_formal.css (file)	183
l warp_mathjax.txt (file)	188
l warp_sagebrush.css (file)	178
l warp_tutorial.txt (file)	54
l warpmk	
customizing	112
[l warpmk]:	
htmlglossary (option)	89, 468
printglossary (option)	89, 468
l warpmk (option)	124
l warpmk (program)	112, 190
l warpmk.conf (file)	149
l warpmk.lua (file)	112
\LWR@addcmidruletrim	5400
\LWR@addcmidrulewidth	5425
\LWR@addformatwpalignment	5432
\LWR@addleftmostbartag	5283
\LWR@addmathjax	6613
\LWR@addrulewidth	5405
\LWR@afterendverbatim	4794
\LWR@amsmathbody	7014
\LWR@amsmathbodynumbered	7020
\LWR@amsmultiline (boolean)	360
\LWR@atbeginverbatim	4780
\LWR@beginhideamsmath	6665
\LWR@blackborderpadding	7466
\LWR@BlockClassWP (environment)	3708
\LWR@botnavigation	3409
\LWR@caption@begin	6155
\LWR@caption@end	6165
\LWR@clearmidrules	5346
\LWR@closeparagraph	3794
\LWR@closeprevious	3429
\LWR@closetabledatamaxcell	4957
\LWR@cmidrulewidth (length)	299
\LWR@colafterspec	287
\LWR@colatspec	287
\LWR@colbangspec	287
\LWR@colbarspec	287
\LWR@colbeforespec	287
\LWR@copyfile	492
\LWR@createautosec	4185
\LWR@currentcss	3901
\LWR@currenttextcolor	7784
\LWR@descitem	4882
\LWR@docmidrule	5371
\LWR@doingapar (boolean)	222
\LWR@doingcmidrule (boolean)	280
\LWR@doinghline (boolean)	279
\LWR@doingstartpars (boolean)	222
\LWR@doingtbrule (boolean)	280
\LWR@domulticolumn	5518
\LWR@doubledollar	6522
\LWR@emptyatbang (boolean)	280
\LWR@endhideamsmath	6673
\LWR@endminipage	7302
\LWR@endofline	7719
\LWR@endsubminipage	7323
\LWR@ensuredoingapar	3767
\LWR@epubprintpendingfootnotes	3998
\LWR@equationtag	6986
\LWR@existingtabular (boolean)	315
\LWR@externalfilecnt (counter)	354

\LWR@figcaption (environment) . . . . .	6140	\LWR@lateautopage (counter) . . . . .	336
\LWR@filenamenoblanks . . . . .	4024	\LWR@lateximagedepth (counter) . . . . .	370
\LWR@filestart . . . . .	4410	\LWR@lateximagedepthref . . . . .	5932
\LWR@findword . . . . .	386	\LWR@lateximagenumber (counter) . . . . .	369
\LWR@floatbegin . . . . .	6084	\LWR@lateximagenumberref . . . . .	5935
\LWR@floatend . . . . .	6107	\LWR@lateximagesfile . . . . .	454
\LWR@footnotetext . . . . .	3933	\LWR@lightrulewidth (length) . . . . .	299
\LWR@forceminwidth . . . . .	7455	\LWR@LIpage (counter) . . . . .	370
\LWR@forcenewpage . . . . .	3435	\LWR@loadafter . . . . .	111
\LWR@freezethisfloat (boolean) . . . . .	335	\LWR@loadbefore . . . . .	122
\LWR@futurenonspacelet . . . . .	4922	\LWR@loadnever . . . . .	131
\LWR@FVstyle . . . . .	17	\LWR@longtabledatacaptiontag . . . . .	5559
\LWR@getexpparray . . . . .	3328	\LWR@lookforpackagename . . . . .	389
\LWR@getmynexttoken . . . . .	4929	\LWR@LwarpEnd . . . . .	4529, 7908
\LWR@heavyrulewidth (length) . . . . .	299	\LWR@LwarpStart . . . . .	4460, 7908
\LWR@hidelatexequation . . . . .	6599	\LWR@makteitlesetup . . . . .	33, 4634
\LWR@hline . . . . .	5764	\LWR@maybeinc>thisfloat . . . . .	6126
\LWR@hspace . . . . .	7744	\LWR@maybenewtablerow . . . . .	5231
\LWR@htmlblockcomment . . . . .	3620	\LWR@midrulecounter (counter) . . . . .	299
\LWR@htmlblocktag . . . . .	3628	\LWR@midrules . . . . .	299
\LWR@htmclosecomment . . . . .	3592	\LWR@minipagedepth (counter) . . . . .	387
\LWR@htmlcomment . . . . .	3613	\LWR@minipagefullwidth (boolean) . . . . .	389
\LWR@htmldivclass . . . . .	3657	\LWR@minipageheight (length) . . . . .	388
\LWR@htmldivclassend . . . . .	3660	\LWR@minipagestartpars . . . . .	7724
\LWR@htmlelement . . . . .	3670	\LWR@minipagestoppars . . . . .	7727
\LWR@htmlelementclass . . . . .	3644	\LWR@minipagethispar (boolean) . . . . .	389
\LWR@htmlelementclassend . . . . .	3649	\LWR@minipagewidth (length) . . . . .	387
\LWR@htmlelementclassline . . . . .	3663	\LWR@multicolother . . . . .	5469
\LWR@htmlelementend . . . . .	3673	\LWR@multicolpartext . . . . .	5464
\LWR@htmfilename (counter) . . . . .	207	\LWR@multicolskip . . . . .	5475
\LWR@htmfileref . . . . .	5928	\LWR@multirowborder . . . . .	3
\LWR@htmlmathlabel . . . . .	6650	\LWR@mynexttoken . . . . .	4921
\LWR@htmlemulticolumn . . . . .	5548	\LWR@myshorttoc . . . . .	6220
\LWR@htmlopencomment . . . . .	3592	\LWR@nameref . . . . .	5925
\LWR@htmrefsectionfilename . . . . .	3394	\LWR@nestspan (environment) . . . . .	3524
\LWR@HTMLsanitize . . . . .	6931	\LWR@newhtmlfile . . . . .	4090
\LWR@HTMLsanitizeexpand . . . . .	6952	\LWR@newlabel . . . . .	5962
\LWR@htmlsectionfilename . . . . .	3361	\LWR@newref . . . . .	6012
\LWR@htmllspan . . . . .	3565	\LWR@nextautofloat (counter) . . . . .	336
\LWR@htmllspanclass . . . . .	3573	\LWR@nextautopage (counter) . . . . .	336
\LWR@htmlltag . . . . .	3583	\LWR@nextequation (counter) . . . . .	356
\LWR@htmlltagc . . . . .	3514	\LWR@nohspace . . . . .	7773
\LWR@HTMLtextstyle . . . . .	7540	\LWR@nullfonts . . . . .	7610
\LWR@includegraphicsb . . . . .	105	\LWR@nullifyNoAutoSpacing . . . . .	5781
\LWR@indexitem . . . . .	6390, 6394, 6398	\LWR@openparagraph . . . . .	3772
\LWR@infloatrow (boolean) . . . . .	472	\LWR@origcolspec . . . . .	5064
\LWR@instertatbangcols . . . . .	4949	\LWR@originname . . . . .	244
\LWR@intabularmetadata (boolean) . . . . .	280	\LWR@originnames . . . . .	251
\LWR@isstartingequation (boolean) . . . . .	372	\LWR@parseaftercolumn . . . . .	5124
\LWR@itemizeitem . . . . .	4847	\LWR@parseatcolumn . . . . .	5070

\LWR@parsebangcolumn .....	5094	\LWR@strresult .....	5062
\LWR@parsebarcolumn .....	5134	\LWR@subaddcmidruletrim .....	5389
\LWR@parsebeforecolumn .....	5115	\LWR@subcmidrule .....	5360
\LWR@parseDcolumn .....	5161	\LWR@subhtmelementclass .....	3633
\LWR@parseenormalcolumn .....	5145	\LWR@subhyperref .....	6036
\LWR@parsepcolumn .....	5157	\LWR@subhyperrefclass .....	6040
\LWR@parsetablecols .....	5166	\LWR@subinlineimage .....	6077
\LWR@patchlists .....	4910	\LWR@subminipage .....	7317
\LWR@prevFileDepth (counter) .....	242	\LWR@subnewref .....	6006
\LWR@printatbang .....	5261	\LWR@subsublabel .....	5941
\LWR@printbartag .....	5253	\LWR@subtableofcontents .....	6235
\LWR@printmcoldata .....	5478	\LWR@syncmathjax .....	6579
\LWR@printmcoltpe .....	5448	\LWR@tablecolspos (counter) .....	286
\LWR@printpendingfootnotes .....	3988	\LWR@tablecolswidth (counter) .....	286
\LWR@printthetitle .....	4593	\LWR@tabledatacolumnntag .....	5693
\LWR@ProvidesPackageDrop .....	441	\LWR@tabledatasinglecolumnntag .....	5296
\LWR@ProvidesPackagePass .....	429	\LWR@tableparcell (boolean) .....	280
\LWR@pushoneclose .....	4188	\LWR@tabletotalcols (counter) .....	287
\LWR@quickfile .....	451	\LWR@tabletotalcolsnext (counter) .....	287
\LWR@remembertag .....	6990	\LWR@tabular (environment) .....	5794
\LWR@requesttoc .....	4522	\LWR@tabularendofline .....	5038
\LWR@requirepackagenames .....	385	\LWR@tabularfinishrow .....	5013
\LWR@restoreorigaccents .....	701	\LWR@tabularhtmlcolumns .....	5625
\LWR@restoreorigformatting .....	6424	\LWR@tabularleftedge .....	5288
\LWR@restoreorigprintxcolor .....	7248	\LWR@tabularlarmutemods (boolean) .....	315
\LWR@rotboxorigin .....	243	\LWR@tdaddstyle .....	5377
\LWR@rotstyle .....	91	\LWR@tdendstyles .....	5383
\LWR@rule .....	7785	\LWR@tdstartstyles .....	5376
\LWR@scalestyle .....	94	\LWR@tempheight (length) .....	406
\LWR@section .....	4198	\LWR@tempraise (length) .....	406
\LWR@sectionnumber .....	4182	\LWR@tempwidth (length) .....	406
\LWR@setexparray .....	3323	\LWR@thiscmidrulewidth (length) .....	300
\LWR@setlatestname .....	5904	\LWR@thisfilename .....	4022
\LWR@setOSWindows .....	29	\LWR@thisfloat (counter) .....	335
\LWR@sidetoc .....	6305	\LWR@thisnewfilename .....	4023
\LWR@singledollar .....	6540	\LWR@titlingmaketitle .....	76, 4682
\LWR@skipatbang (boolean) .....	280	\LWR@topnavigation .....	3406
\LWR@skippingmcrowcell (boolean) .....	280	\LWR@traceinfo .....	513
\LWR@skippingmrowcell (boolean) .....	280	\LWR@tracingl warp (boolean) .....	141
\LWR@spandepth (counter) .....	222	\LWR@triml rules .....	299
\LWR@splabel .....	5938	\LWR@trimr rules .....	299
\LWR@starredlongtable (boolean) .....	309	\LWR@validtablecol (boolean) .....	291
\LWR@startedrow (boolean) .....	279	\LWR@verbtags (boolean) .....	269
\LWR@startingequation (counter) .....	371	\LWR@vspace .....	7775
\LWR@startingequationtag .....	6985	\LWR@WPcell .....	5428
\LWR@startnewdepth .....	4189	\LWRprint@fminipage (environment) .....	7512
\LWR@startpars .....	3831	\LWRsetnextfloat .....	6134
\LWR@startref .....	5975	\LyX .....	7889
\LWR@stoppars .....	3847		
\LWR@stripperiod .....	5903		

M	
Mac OS (program) . . . . .	<i>74, 122</i>
Madcap (program) . . . . .	<i>46</i>
\mainmatter . . . . .	<i>4172</i>
\makebox . . . . .	<i>7427</i>
makeidx (package) . . . . .	<i>135</i>
\MakeIndex . . . . .	<i>7881</i>
\maketitle . . . . .	<i>40, 76, 4646</i>
margin	
numbers . . . . .	<i>119</i>
tags . . . . .	<i>119</i>
marginfix (package) . . . . .	<i>496</i>
marginfix (package) . . . . .	<i>496</i>
marginnote (package) . . . . .	<i>497</i>
\marginpar . . . . .	<i>4005</i>
\markboth . . . . .	<i>3440</i>
\markright . . . . .	<i>3441</i>
markup languages . . . . .	<i>46</i>
math	
font size — SVG . . . . .	<i>91, 352, 369</i>
mathjax option . . . . .	<i>124</i>
MathJax summary . . . . .	<i>91, 352</i>
mathsvg option . . . . .	<i>124</i>
showing as HTML . . . . .	<i>60, 115</i>
SVG summary . . . . .	<i>91, 352</i>
word processor conversion . . . . .	<i>108</i>
math (environment) . . . . .	<i>6576</i>
MathJax	
custom macros . . . . .	<i>92, 353</i>
mathjax option . . . . .	<i>124</i>
subequations . . . . .	<i>92, 353</i>
summary . . . . .	<i>91, 352</i>
tagged equations . . . . .	<i>92, 352</i>
MathJax (program) . . . . .	<i>91, 92, 352</i>
MathJax (program) [requirement] . . . . .	<i>48</i>
mathjax (boolean) . . . . .	<i>123</i>
mathjax (option) . . . . .	<i>63, 73, 124</i>
mathsvg (option) . . . . .	<i>63, 73, 124</i>
\mcaption (package) . . . . .	<i>497</i>
\mcolrowcell . . . . .	<i>5754</i>
mdframed (package) . . . . .	<i>497</i>
\mdseries . . . . .	<i>7643</i>
memoir	
verse . . . . .	<i>592</i>
memoir (package) . . . . .	<i>592</i>
meta tag, HTML	
author . . . . .	<i>77, 229</i>
description . . . . .	<i>77, 230</i>
generator . . . . .	<i>251</i>
viewport . . . . .	<i>251</i>
metalogo (package) . . . . .	<i>506</i>
microtype (package) . . . . .	<i>73, 132, 507</i>
midfloat (package) . . . . .	<i>508</i>
\MiKTeX . . . . .	<i>7888</i>
minipage	
framed . . . . .	<i>82</i>
horizontal space between . . . . .	<i>405</i>
minipage (environment) . . . . .	<i>7338</i>
\minipagewidth . . . . .	<i>7330</i>
misplaced \noalign . . . . .	<i>98, 278</i>
misplaced alignment tab character & . . . . .	<i>96, 276, 282</i>
missing sections . . . . .	<i>64</i>
moreverb (package) . . . . .	<i>508</i>
morewrites (package) . . . . .	<i>509</i>
\mparhack (package) . . . . .	<i>510</i>
\mrowcell . . . . .	<i>5751</i>
MS-Windows (program) . . . . .	<i>74, 122</i>
multicol (package) . . . . .	<i>510</i>
multicolumn	
with multirow . . . . .	<i>513</i>
\multicolumnrow . . . . .	<i>36, 5646</i>
multirow	
with multicolumn . . . . .	<i>513</i>
\multirow . . . . .	<i>512</i>
multirow (package) . . . . .	<i>511</i>
multiline (environment) . . . . .	<i>6730</i>
multiline* (environment) . . . . .	<i>6754</i>
N	
\Nameref . . . . .	<i>6033</i>
\nameref . . . . .	<i>6024</i>
nameref (package) . . . . .	<i>514</i>
needspace (package) . . . . .	<i>514</i>
newclude (package) . . . . .	<i>102, 515</i>
newfloat	
with trivfloat, algorithmic . . . . .	<i>588</i>
\newfloat . . . . .	<i>2, 4</i>
newfloat (package) . . . . .	<i>135</i>
\newline . . . . .	<i>7717</i>
\newtheorem . . . . .	<i>271</i>
newtxmath (package) . . . . .	<i>94</i>
newunicodechar (package) . . . . .	<i>72, 515</i>
nextpage (package) . . . . .	<i>515</i>
nicefrac (package) . . . . .	<i>93</i>
\nolinebreak . . . . .	<i>7778</i>
\nolinkurl . . . . .	<i>6057</i>
nonumonpart (package) . . . . .	<i>516</i>
\nopagebreak . . . . .	<i>7782</i>
\nopagecolor . . . . .	<i>44</i>

nopageno (package) . . . . .	516	amsthm . . . . .	421
\normalfont . . . . .	7651	anonchap . . . . .	424
nowidow (package) . . . . .	516	any size . . . . .	424
ntheorem (package) . . . . .	517	appendix . . . . .	424
\numberline . . . . .	6323	arabicfront . . . . .	425
numbers		array . . . . .	323
left margin . . . . .	119	authblk . . . . .	425
numindex (option) [tocbibind] . . . . .	90, 580	babel . . . . .	102
<b>O</b>			
OpenOffice (program) . . . . .	45	balance . . . . .	426
options:		bigdelim . . . . .	426
[lwarpmk]:		bigstrut . . . . .	428
htmlglossary . . . . .	89, 468	bookmark . . . . .	428
printglossary . . . . .	89, 468	booktabs . . . . .	429
[tocbibind]:		boxedminipage . . . . .	429
numindex . . . . .	90, 580	boxedminipage2e . . . . .	429
[tocloft]:		calc . . . . .	135
titles . . . . .	88	caption . . . . .	135
BaseJobname . . . . .	124	caption2 . . . . .	430
HomeHTMLFilename . . . . .	63, 125	ccaption . . . . .	430
HTMLFilename . . . . .	63, 125	change page . . . . .	430
IndexLanguage . . . . .	70, 89, 124, 468, 579	chappg . . . . .	431
latexmk . . . . .	63, 125	chngpage . . . . .	431
lwarpmk . . . . .	124	cleveref . . . . .	384
mathjax . . . . .	63, 73, 124	cmap . . . . .	73
mathsvg . . . . .	63, 73, 124	color . . . . .	431
OSWindows . . . . .	74, 122, 125	comment . . . . .	126
warpHTML . . . . .	73, 123	crop . . . . .	431
warpprint . . . . .	73, 123	cuted . . . . .	432
xdyFilename . . . . .	79, 124	cutwin . . . . .	432
\OSPathSymbol . . . . .	28	dblfnote . . . . .	433
OSWindows (option) . . . . .	74, 122, 125	dcolumn . . . . .	433
overpic (package) . . . . .	528	draft watermark . . . . .	434
<b>P</b>			
packages		ebook . . . . .	434
required . . . . .	132	ellipsis . . . . .	435
packages:		empty page . . . . .	435
a4 . . . . .	416	endnotes . . . . .	435
a4wide . . . . .	416	enumerate . . . . .	436
a5comb . . . . .	416	enumitem . . . . .	436
abstract . . . . .	416	environ . . . . .	135
addlines . . . . .	419	epigraph . . . . .	437
adjmulticol . . . . .	418	eso-pic . . . . .	438
afterpackage . . . . .	134	etoolbox . . . . .	121
afterpage . . . . .	419	everyhook . . . . .	134
algorithmicx . . . . .	420	every page . . . . .	438
alltt . . . . .	421	everyshi . . . . .	439
amsmath . . . . .	136	expl3 . . . . .	134
		extramarks . . . . .	439
		fancybox . . . . .	440
		fancyhdr . . . . .	444
		fancyvrb . . . . .	444

figcaps	450	memoir	592
filecontents	134	metalogo	506
fix2col	450	microtype	73, 132, 507
float	450	midfloat	508
floatflt	452	moreverb	508
floatpag	453	morewrites	509
floatrow	453	mparhack	510
flushend	459	multicol	510
fncychap	459	multirow	511
fnpos	460	nameref	514
fontenc	72, 460	needspace	514
fontspec	72, 460	newclude	102, 515
footmisc	461	newfloat	135
footnote	462	newtxmath	94
footnotehyper	463	newunicodechar	72, 515
framed	463	nextpage	515
ftnright	466	nicefrac	93
fullpage	466	nonumonpart	516
fullwidth	466	nopageno	516
geometry	133, 467	nowidow	516
gettitlestring	134	ntheorem	517
glossaries	88, 467	overpic	528
graphics	469	pagenote	529
graphicx	480	paralist	529
grffile	480	parskip	530
hyperref	330, 480	pdflscape	530
hyperxmp	485	pdfsync	530
idxlayout	485	pfnote	531
ifplatform	121	placeins	531
indentfirst	486	prelim2e	531
inputenc	72, 486	quotchap	532
keyfloat	486	ragged2e	533
kvoptions	123	realscripts	534
layout	488	refcount	135
letltxmacro	121	relsize	534
letterspace	488	romanbar	536
lettrine	488	romanbarpagenumber	536
lips	489	rotating	536
listings	490	rotfloat	537
lmodern	72	savetrees	538
longtable	493	scalefnt	538
lscape	495	sectsty	538
ltcaption	495	setspace	539
lwarp	73	shadow	540
makeidx	135	showidx	540
marginfit	496	showkeys	541
marginfix	496	sidecap	541
marginnote	497	sidenotes	542
mcaption	497	siunitx	93, 379, 543
mdframed	497	soul	545

stabular	547	paralist (package)	529
subfig	547	\parbox	7420
supertabular	553	\parsemulticolumnalignment	5505
syntonly	554	parskip (package)	530
tabls	555	\part	4336
tabularx	555	pdfcrop (program) [requirement]	48
tabulary	556	pdfLaTeX (program) [requirement]	48
textarea	556	pdflscape (package)	530
textcomp	73, 557	pdfseparate (program) [requirement]	
textpos	560		48, 52
theorem	561	pdfsync (package)	530
threeparttable	565	pdftocairo (program) [requirement]	48, 52
tikz	566	pdftotext (program) [requirement]	48, 52
titleps	567	Perl	53
titleref	570	perl (program) [requirement]	53
titlesec	570	pfnote (package)	531
titletoc	572	\phantomsection	7837
titling	574	picture (environment)	386, 7293
tocbibind	90, 579	placeins (package)	531
tocloft	581	Plastex (program)	44
transparent	586	\popclose	3296
trivfloat	587	Poppler	48, 52
typearea	588	prelim2e (package)	531
ulem	589	\printauthor	258, 4601, 4620
units	93	\printdate	258, 4612, 4622
upref	591	printglossary (option) [lwarpmk]	89, 468
verse	591, 592	\printindex	6373
wallpaper	593	\PrintStack	3414
wrapfig	593	\printthanks	257, 260
xcolor	381, 595	\printtitle	258, 4585, 4619
xfrac	603	problems	114
xifthen	135	programs	
xltxtra	605	utility	47
xmpincl	606	programs:	
xparse	133	[requirement]:	
xstring	135	LuaTeX	48
xtab	606	MathJax	48
zref	136	pdfcrop	48
zwpagelayout	608	pdfLaTeX	48
page		pdfseparate	48, 52
inaccessible	64	pdftocairo	48, 52
\pagebreak	7779	pdftotext	48, 52
\pagecolor	43	perl	53
\pagernote (package)	529	XeLaTeX	48
\pagenumbering	3446	Adobe	46
\pageref	6019	AsciiDoc	45
\pagerefPageFor	6018	AsciiDoctor	45
\pagestyle	3438	Asciidoctor-LaTeX	45
Pandoc (program)	45	Flare	46
\paragraph	4385	FrameMaker	46

GELLMU	44	pdfcrop (program)	48
GladTeX	45	pdfLaTeX (program)	48
Hevea	44	pdfseparate (program)	48, 52
InDesign	46	pdftocairo (program)	48, 52
LaTeX2HTML	44	pdftotext (program)	48, 52
LaTeXXML	44	perl (program)	53
LibreOffice	45	XeLaTeX (program)	48
Linux	74, 122	\RequirePackage	405
lwarpmk	112, 190	\resizebox	302
Mac OS	74, 122	\restylefloat	17
Madcap	46	\ResumeTabular	5680
MathJax	91, 92, 352	\rmfamily	7645
MS-Windows	74, 122	romanbar (package)	536
OpenOffice	45	romanbarpagenumber (package)	536
Pandoc	45	\rotatebox	263
Plastex	44	rotating (package)	536
TeX2page	44	rotfloat (package)	537
TeX4ht	44	\rule	7834
TeXMaths	108		
TtH	44		
Unix	74, 122	<b>S</b>	
Windows	74, 122	sample_project.css (file)	78, 187
Word	45	savetrees (package)	538
xindy	78	\sb	7653
project.css (file)	78	\scalebox	282
project.lwarpmkconf (file)	150	scalefnt (package)	538
project_html.tex (file)	149	\scshape	7650
published	265	section	
\pushclose	3269	depths	118
		heading, word processor	108
		missing	64
		\section	4360
<b>Q</b>		sectsty (package)	538
\qqquad	405, 7735	\SetHTMLFileName	3352
\quad	405, 7730	setspace (package)	539
quotchap (package)	532	settings	
quote (environment)	4753	css project-specific	78
		CSS selection	78
		HTML conversion	63
		lwarpmk package options	73
		selecting output	75
		title page	76
<b>R</b>		\sffamily	7646
ragged2e (package)	533	\sfrac	603
\raggedbottom	3442	shadow (package)	540
\raggedleft	254	showidx (package)	540
\raggedright	254	showkeys (package)	541
\raisebox	7505	sidecap (package)	541
realscripts (package)	534	sidenotes (package)	542
\ref	6012	SideTOCDepth (counter)	64, 343
refcount (package)	135	\sidetocname	6302
\reflectbox	297		
relsize (package)	534		
[requirement]:			
LuaLaTeX (program)	48		
MathJax (program)	48		

\simplechapterdelim	4196	TeXMaths (program)	108
sunitx		textarea (package)	556
with TeXMaths	108	\textbf	7554
siunitx (package)	93, 379, 543	\textcolor	33
\sloppy	3444	\textcomp	73, 557
soul (package)	545	\textgreater	3342
\sp	7652	\textit	7579
space		\textless	3342
horizontal	79	\textmd	7548
between minipages	405	\textnormal	7596
split		\textpos	560
miss-numbered	93, 517	\textrm	7557
stabular (package)	547	\textsc	7582
stack depths	118	\textsf	7564
\StartDefiningTabulars	4992	\textsl	7589
subequations		\textsubscript	7656
MathJax	92, 353	\textsuperscript	7654
subfig (package)	547	\texttt	7569
\subparagraph	4393	\textup	7572
\subsection	4368	\tfl@chapter@fix	587
\subsubsection	4376	\thanks	77
subtitle	265	\thanksmarkseries	84
supertabular (package)	553	\theauthor	258
SVG		\thedate	258
images showing as HTML	60, 115	theindex (environment)	6380
math summary	91, 352	theorem (package)	561
mathsvg option	124	\thetitle	258
syntonly (package)	554	\thispagestyle	3439
<b>T</b>			
tabbing (environment)	4820	threeparttable (package)	565
\tableofcontents	66, 6252	tikz	
tabs (package)	555	catcodes	566
tabular		dollar redefined	566
baseline	292	tikz (package)	566
HTML columnn conversion	292	\tikzpicture (environment)	8
macros inside	97, 277	\title	76
misplaced \noalign	98, 278	titlepage	
misplaced alignment tab character &		subtitle and published	265
	96, 276	titlepage (environment)	76, 4569
multicolumn with multirow	513	titleps (package)	567
row corruption	97, 277	titleref (package)	570
\TabularMacro	5678	titles (option) [tocloft]	88
tabularx (package)	555	titlesec (package)	570
tabulary (package)	556	titletoc (package)	572
tagged equations		titling (package)	574
Mathjax	92, 352	titlingpage (environment)	14, 76
\TeX	7842	[tocbibind]:	
\TeX2page (program)	44	numindex (option)	90, 580
\TeX4ht (program)	44	tocbibind (package)	90, 579
		tocdepth (counter)	63

[tocloft]:	
titles (option) . . . . .	88
tocloft (package) . . . . .	581
\tracingl warp . . . . .	116, 512
transparent (package) . . . . .	586
trivfloat	
with newfloat, algorithmic . . . . .	588
trivfloat (package) . . . . .	587
troubleshooting . . . . .	114
HTML debug comments . . . . .	142
\ttfamily . . . . .	7647
TtH (program) . . . . .	44
tutorial.tex (file) . . . . .	54
typearea (package) . . . . .	588
<b>U</b>	
ulem (package) . . . . .	589
\underline . . . . .	7664
underscore	
filename . . . . .	63, 80
Unicode	
enhanced coverage . . . . .	72
file & section names . . . . .	240
input characters . . . . .	120
selection . . . . .	72
units (package) . . . . .	93
Unix (program) . . . . .	74, 122
\up . . . . .	7662
upref (package) . . . . .	591
\upshape . . . . .	7648
\url . . . . .	6064
usingOSWindows (boolean) . . . . .	122
UTF-8	
enhanced coverage . . . . .	72
file & section names . . . . .	240
index . . . . .	73
selection . . . . .	72
utility	
programs . . . . .	47
<b>V</b>	
Verbatim (environment) . . . . .	13
verbatim (environment) . . . . .	4810
\VerbatimHTMLWidth (length) . . . . .	269
\verbatiminput . . . . .	4802
verse (environment) . . . . .	2
verse (package) . . . . .	591, 592
viewport	
HTML meta tag . . . . .	251
vleftmargini (length) . . . . .	101, 268, 592
vleftskip (length) . . . . .	101, 268, 592
\vspace . . . . .	7776
<b>W</b>	
wallpaper (package) . . . . .	593
warning icon . . . . .	119
warpall (environment) . . . . .	75, 100
warpHTML (environment) . . . . .	50, 67, 75, 101
warpHTML (option) . . . . .	73, 123
\warpHTMLonly . . . . .	67, 75, 90
warpingHTML (boolean) . . . . .	123
warpingprint (boolean) . . . . .	123
warpprint (environment) . . . . .	66, 75, 101
warpprint (option) . . . . .	73, 123
\warpprintonly . . . . .	67, 75, 89
Windows (program) . . . . .	74, 122
Word (program) . . . . .	45
word processor	
conversion recommendations . . . . .	108
HTML conversion settings . . . . .	105, 142, 143
section headings . . . . .	108
WPMarksFloats (boolean) . . . . .	107, 142
WPMarksLOFT (boolean) . . . . .	108, 143
WPMarksMath (boolean) . . . . .	108, 144
WPMarksMinipages (boolean) . . . . .	107, 143
WPMarksTOC (boolean) . . . . .	107, 143
WPTitleHeading (boolean) . . . . .	108, 144
wrapfig (package) . . . . .	593
<b>X</b>	
xcolor (package) . . . . .	381, 595
xdyFilename (option) . . . . .	79, 124
XeLaTeX	
detection . . . . .	132
file & section names . . . . .	240
\XeLaTeX . . . . .	7874
XeLaTeX (program) [requirement] . . . . .	48
\XeTeX . . . . .	7874
xfrac (package) . . . . .	603
\xfracHTMLfontsize . . . . .	3
xifthen (package) . . . . .	135
xindy	
customizing . . . . .	78
xindy (program) . . . . .	78
\xltextra (package) . . . . .	605
xmpincl (package) . . . . .	606
xparse (package) . . . . .	133
xstring (package) . . . . .	135
xtab (package) . . . . .	606

<b>Z</b>	
<b>zwpagelayout</b> (package) . . . . .	<b>608</b>
<b>zref</b> (package) . . . . .	<b>136</b>