



# The **lwarp** package

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

v0.50 — 2018/03/03

© 2016–2018 Brian Dunn  
bd@BDTechConcepts.com

### Abstract

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

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

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

A modular package-loading system uses the lwarp version of a package for HTML when available. More than two 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 a “beta” version of lwarp, and some things may change in response to user feedback and further project development.**

### License:

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

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

T<sub>E</sub>X and related projects:

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

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

Donations may be directed towards individual projects:

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

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

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

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

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

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

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

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

**MetaPost:** Postscript graphics.

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

**PDF Accessibility:** Modern PDF standards.

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

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

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

For users of MiK<sub>T</sub>E<sub>X</sub>: <https://miktex.org/donations.html>

# Contents

<b>1</b>	<b>Supporting TeX development</b>	<b>2</b>
	<b>List of Figures</b>	<b>40</b>
	<b>List of Tables</b>	<b>40</b>
<b>2</b>	<b>Updates</b>	<b>41</b>
<b>3</b>	<b>Introduction</b>	<b>52</b>
3.1	Supported packages and features . . . . .	54
<b>4</b>	<b>Alternatives</b>	<b>58</b>
4.1	Internet class . . . . .	58
4.2	TeX4ht . . . . .	58
4.3	Translators . . . . .	58
4.4	AsciiDoc and AsciiDoctor . . . . .	59
4.4.1	AsciiDoctor-LaTeX . . . . .	59
4.5	Pandoc . . . . .	59
4.6	Word processors . . . . .	59
4.7	Commercial systems . . . . .	59
4.8	Comparisons . . . . .	59
<b>5</b>	<b>Installation</b>	<b>61</b>
5.1	Installing the lwarp package . . . . .	63
5.2	Installing the lwarpmk utility . . . . .	64
5.2.1	Using a local copy of lwarpmk . . . . .	66
5.3	Installing additional utilities . . . . .	66
<b>6</b>	<b>Tutorial</b>	<b>69</b>
6.1	Starting a new project . . . . .	69
6.2	Compiling the print version with lwarpmk . . . . .	73
6.3	Compiling the HTML version with lwarpmk . . . . .	74
6.4	Generating the SVG images . . . . .	75
6.5	Using MATHJAX for math . . . . .	76
6.6	Changing the CSS style . . . . .	77
6.7	Customizing the HTML output . . . . .	77
6.8	Using latexmk . . . . .	78
6.9	Using XeLaTeX or LuaLaTeX . . . . .	79
6.10	Using a glossary . . . . .	80
6.11	Cleaning auxiliary files . . . . .	81
6.12	Cleaning auxiliary and output files . . . . .	81
6.13	Processing multiple projects in the same directory . . . . .	81
6.14	Using the make utility . . . . .	81
6.15	Converting an existing document . . . . .	82

<b>7</b>	<b>Additional details</b>	<b>83</b>
7.1	Font and UTF-8 support	83
7.1.1	Indexes and UTF-8	84
7.2	lwarp package loading and options	85
7.3	Customizing the HTML output	87
7.3.1	Example HTML file naming	91
7.4	Customizing the CSS	92
7.5	Selecting the operating system	93
7.6	Selecting actions for print or HTML output	94
7.7	Commands to be placed into the warpprint environment	95
7.8	Title page	95
7.9	HTML page meta descriptions	96
7.10	HTML page meta title	97
7.11	HTML page meta author	97
7.12	Modifying xindy index processing	97
<b>8</b>	<b>Special cases and limitations</b>	<b>98</b>
8.1	Sectioning	98
8.2	Formatting	98
8.2.1	Text formatting	98
8.2.2	Horizontal space	98
8.2.3	Text alignment	98
8.2.4	Accents	99
8.2.5	Textcomp	99
8.2.6	Superscripts and other non-math uses of math mode	99
8.2.7	Empty <code>\item</code> followed by a new line of text or a nested list:	99
8.2.8	Filenames and URLs in lists or footnotes	99
8.2.9	relsize package	99
8.3	Boxes and minipages	100
8.3.1	Marginpars	100
8.3.2	Save Boxes	100
8.3.3	Minipages	100
8.3.4	Side-by-side minipages	101
8.3.5	Framed minipages and other environments	101
8.3.6	fancybox package	102
8.3.7	mdframed package	104
8.4	Cross-references	104
8.4.1	Page references	104
8.4.2	cleveref and varioref packages	105
8.4.3	Hyperlinks, hyperref, and url	105
8.4.4	Footnotes and page notes	105
8.5	Front and back matter	106
8.5.1	Starred chapters and sections	106
8.5.2	abstract package	106
8.5.3	titling and authblk	107
8.5.4	tocloft package	107

8.5.5	appendix package	107
8.5.6	pagenote package	107
8.5.7	endnotes package	107
8.5.8	glossaries package	107
8.5.9	Index and the tocbibind package	108
8.6	Math	109
8.6.1	Rendering tradeoffs	109
8.6.2	SVG option	110
8.6.3	MATHJAX option	111
8.6.4	ntheorem package	112
8.6.5	siunitx package	112
8.6.6	units and nicefrac packages	113
8.6.7	newtxmath package	113
8.7	Graphics	113
8.7.1	grffile package	114
8.7.2	color package	114
8.7.3	xcolor package	114
8.7.4	epstopdf package	115
8.7.5	overpic package	115
8.8	Tabbing	115
8.9	Tabular	115
8.9.1	longtable package	118
8.9.2	supertabular and xtab packages	119
8.9.3	bigdelim package	119
8.10	Floats	120
8.10.1	float, trivfloat, and/or algorithmicx together	120
8.10.2	caption and subcaption packages	120
8.10.3	subfig package	120
8.10.4	floatrow package	120
8.10.5	keyfloat package	121
8.11	Koma-Script	121
8.12	Memoir	121
8.13	Miscellaneous	122
8.13.1	verse and memoir	122
8.13.2	newclude package	123
8.13.3	babel package	123
8.13.4	todonotes and luatodonotes packages	123
8.13.5	fixme	124
8.13.6	xparse	124
<b>9</b>	<b>EPUB conversion</b>	<b>125</b>
<b>10</b>	<b>Word-processor conversion</b>	<b>127</b>
10.1	Activating word-processor conversion	127
10.2	Additional modifications	129
10.3	Recommendations	130

10.4	Limitations . . . . .	131
<b>11</b>	<b>Modifying lwarp</b>	<b>133</b>
11.1	Modifying a package for lwarp . . . . .	133
11.1.1	Adding a package to the lwarp.dtx file . . . . .	134
11.2	Modifying a class for lwarp . . . . .	134
11.3	Testing lwarp . . . . .	135
11.4	Modifying lwarpmk . . . . .	135
<b>12</b>	<b>Troubleshooting</b>	<b>136</b>
12.1	Using the lwarp.sty package . . . . .	136
12.1.1	Debug tracing output . . . . .	139
12.2	Compiling the lwarp.dtx file . . . . .	139
<b>1</b>	<b>lwarp.sty</b>	<b>139</b>
<b>13</b>	<b>Implementation</b>	<b>139</b>
<b>14</b>	<b>Section depths and HTML headings</b>	<b>140</b>
<b>15</b>	<b>Source Code</b>	<b>142</b>
<b>16</b>	<b>Detecting the TeX Engine — pdflatex, lualatex, xelatex</b>	<b>143</b>
<b>17</b>	<b>MD5 hashing</b>	<b>143</b>
<b>18</b>	<b>pdfLaTeX T1 and UTF8 encoding</b>	<b>144</b>
<b>19</b>	<b>Unicode input characters</b>	<b>144</b>
<b>20</b>	<b>Miscellaneous tools</b>	<b>145</b>
<b>21</b>	<b>Early package requirements</b>	<b>145</b>
<b>22</b>	<b>Operating-System portability</b>	<b>146</b>
22.1	Common portability code . . . . .	146
22.2	Unix, Linux, and Mac OS . . . . .	146
22.3	MS-WINDOWS . . . . .	146
<b>23</b>	<b>Package options</b>	<b>147</b>
23.1	Conditional compilation . . . . .	150
<b>24</b>	<b>Package load order</b>	<b>151</b>
24.1	Tests of package load order . . . . .	151
24.2	Enforcing package loading after lwarp . . . . .	153
<b>25</b>	<b>Required packages</b>	<b>158</b>

<b>26</b>	<b>Loading packages</b>	<b>162</b>
<b>27</b>	<b>Additional required packages</b>	<b>166</b>
<b>28</b>	<b>File handles</b>	<b>166</b>
<b>29</b>	<b>Include a file</b>	<b>167</b>
<b>30</b>	<b>Copying a file</b>	<b>168</b>
<b>31</b>	<b>Debugging messages</b>	<b>169</b>
<b>32</b>	<b>HTML-conversion output modifications</b>	<b>170</b>
32.1	User-level controls . . . . .	170
32.2	Heading adjustments . . . . .	172
<b>33</b>	<b>Remembering original formatting macros</b>	<b>173</b>
<b>34</b>	<b>Accents</b>	<b>176</b>
<b>35</b>	<b>Configuration Files</b>	<b>177</b>
35.1	project_html.tex . . . . .	178
35.2	lwarpmk.conf . . . . .	178
35.3	project.lwarpmkconf . . . . .	179
35.4	lwarp.css . . . . .	180
35.5	lwarp_sagebrush.css . . . . .	208
35.6	lwarp_formal.css . . . . .	213
35.7	sample_project.css . . . . .	217
35.8	lwarp.xdy . . . . .	218
35.9	lwarp_one_limage.cmd . . . . .	219
35.10	lwarp_mathjax.txt . . . . .	219
35.11	lwarpmk option . . . . .	222
<b>36</b>	<b>Stacks</b>	<b>235</b>
36.1	Assigning depths . . . . .	235
36.2	Closing actions . . . . .	236
36.3	Closing depths . . . . .	236
36.4	Pushing and popping the stack . . . . .	237
<b>37</b>	<b>Data arrays</b>	<b>238</b>
<b>38</b>	<b>Sanitizing labels and filenames</b>	<b>239</b>
<b>39</b>	<b>HTML entities</b>	<b>239</b>
<b>40</b>	<b>HTML filename generation</b>	<b>240</b>
<b>41</b>	<b>Homepage link</b>	<b>243</b>

<b>42</b>	<b>\LWRPrintStack diagnostic tool</b>	<b>244</b>
<b>43</b>	<b>Closing stack levels</b>	<b>244</b>
<b>44</b>	<b>PDF pages and styles</b>	<b>245</b>
<b>45</b>	<b>HTML tags, spans, divs, elements</b>	<b>246</b>
45.1	Mapping $\LaTeX$ Sections to HTML Sections . . . . .	247
45.2	Babel-French . . . . .	247
45.3	HTML tags . . . . .	248
45.4	Block tags and comments . . . . .	250
45.5	Div class and element class . . . . .	251
45.6	Single-line elements . . . . .	252
45.7	HTML5 semantic elements . . . . .	253
45.8	High-level block and inline classes . . . . .	253
45.9	Closing HTML tags . . . . .	255
<b>46</b>	<b>Paragraph handling</b>	<b>256</b>
<b>47</b>	<b>Paragraph start/stop handling</b>	<b>259</b>
<b>48</b>	<b>Page headers and footers</b>	<b>262</b>
<b>49</b>	<b>CSS</b>	<b>263</b>
<b>50</b>	<b>Title, HTML meta author, HTML meta description</b>	<b>263</b>
<b>51</b>	<b>Footnotes</b>	<b>264</b>
51.1	Regular page footnotes . . . . .	265
51.2	Minipage footnotes . . . . .	265
51.3	Titlepage thanks . . . . .	265
51.4	Regular page footnote implementation . . . . .	265
51.5	Minipage footnote implementation . . . . .	267
51.6	Printing pending footnotes . . . . .	269
<b>52</b>	<b>Marginpars</b>	<b>270</b>
<b>53</b>	<b>Splitting HTML files</b>	<b>271</b>
<b>54</b>	<b>Sectioning</b>	<b>277</b>
54.1	User-level starred section commands . . . . .	278
54.2	Book class commands . . . . .	278
54.3	Sectioning support macros . . . . .	279
54.4	\section and friends . . . . .	286
<b>55</b>	<b>Starting a new file</b>	<b>288</b>
<b>56</b>	<b>Starting HTML output</b>	<b>291</b>



<b>57</b>	<b>Ending HTML output</b>	<b>294</b>
<b>58</b>	<b>Title page</b>	<b>295</b>
58.1	Setting the title, etc. . . . .	296
58.2	\if@titlepage . . . . .	297
58.3	Changes for \affiliation . . . . .	298
58.4	Printing the thanks . . . . .	299
58.5	Printing the title, etc. in HTML . . . . .	299
58.6	Printing the title, etc. in print form . . . . .	300
58.7	\maketitle for HTML output . . . . .	301
58.8	\published and \subtitle . . . . .	304
<b>59</b>	<b>Abstract</b>	<b>305</b>
<b>60</b>	<b>Quote and verse</b>	<b>306</b>
60.1	Citations and attributions . . . . .	306
60.2	Quotes, quotations . . . . .	307
60.3	Verse . . . . .	307
<b>61</b>	<b>Verbatim and tabbing</b>	<b>308</b>
<b>62</b>	<b>Theorems</b>	<b>311</b>
<b>63</b>	<b>Lists</b>	<b>312</b>
63.1	List environment . . . . .	313
63.2	Itemize . . . . .	316
63.3	Enumerate . . . . .	317
63.4	Description . . . . .	317
63.5	Patching the lists . . . . .	318
<b>64</b>	<b>Tabular</b>	<b>320</b>
64.1	Limitations . . . . .	320
64.2	Token lookahead . . . . .	323
64.3	Tabular variables . . . . .	323
64.4	Handling &, @, !, and bar . . . . .	327
64.4.1	Localizing & catcodes . . . . .	328
64.4.2	Handling & . . . . .	329
64.4.3	Filling an unfinished row . . . . .	330
64.5	Handling \\ . . . . .	332
64.6	Parsing @, >, <, !, bar columns . . . . .	333
64.7	Parsing 'l', 'c', or 'r' columns . . . . .	336
64.8	Parsing 'p', 'm', or 'b' columns . . . . .	336
64.9	Parsing 'D' columns . . . . .	336
64.10	Parsing the column specifications . . . . .	337
64.11	colortbl and xparse tabular color support . . . . .	342
64.12	Starting a new row . . . . .	344
64.13	Printing vertical bar tags . . . . .	345

64.14	Printing at or bang tags	345
64.15	Data opening tag	346
64.16	Midrules	348
64.17	Cell colors	353
64.18	Multicolumns	356
64.18.1	Parsing multicolumns	356
64.18.2	Multicolumn factored code	359
64.18.3	Multicolumn	362
64.18.4	Longtable captions	362
64.18.5	Counting HTML tabular columns	365
64.19	Multicolumnrow	366
64.20	Utility macros inside a table	368
64.21	Special-case tabular markers	368
64.22	Checking for a new table cell	369
64.23	\mrowcell	372
64.24	\mcolrowcell	372
64.25	New tabular environment	373
<b>65</b>	<b>Cross-references</b>	<b>379</b>
65.1	Setup	379
65.2	Zref setup	381
65.3	Labels	382
65.4	References	384
65.5	Hyper-references	386
<b>66</b>	<b>Floats</b>	<b>390</b>
66.1	Float captions	391
66.1.1	Caption inside a float environment	393
66.1.2	Caption and LOF linking and tracking	393
<b>67</b>	<b>Table of Contents, LOF, LOT</b>	<b>397</b>
67.1	Reading and printing the TOC	397
67.2	High-level TOC commands	400
67.3	Side TOC	401
67.4	Low-level TOC line formatting	402
<b>68</b>	<b>Index and glossary</b>	<b>405</b>
<b>69</b>	<b>Bibliography presentation</b>	<b>408</b>
<b>70</b>	<b>Restoring original formatting</b>	<b>409</b>
<b>71</b>	<b>Math</b>	<b>412</b>
71.1	Limitations	412
71.1.1	Rendering tradeoffs	412
71.1.2	SVG option	412
71.1.3	MATHJAX option	413

71.2	Inline and display math . . . . .	414
71.3	MATHJAX support . . . . .	420
71.4	Equation environment . . . . .	422
71.5	AMS Math environments . . . . .	424
71.5.1	Support macros . . . . .	424
71.5.2	Environment patches . . . . .	426
<b>72</b>	<b>Lateximages</b>	<b>433</b>
72.1	Description . . . . .	433
72.2	Support counters and macros . . . . .	434
72.3	Font size . . . . .	434
72.4	Sanitizing math expressions for HTML . . . . .	434
72.5	Equation numbers . . . . .	436
72.6	HTML <alt> tags . . . . .	437
72.7	lateximage . . . . .	438
<b>73</b>	<b>center, flushleft, flushright</b>	<b>443</b>
<b>74</b>	<b>Pre-loaded packages</b>	<b>444</b>
<b>75</b>	<b>Siunitx</b>	<b>444</b>
<b>76</b>	<b>Graphics print-mode modifications</b>	<b>445</b>
76.1	General limitations . . . . .	445
76.2	Print-mode modifications . . . . .	446
<b>77</b>	<b>Xcolor boxes</b>	<b>447</b>
<b>78</b>	<b>Cleveref</b>	<b>451</b>
<b>79</b>	<b>Picture</b>	<b>454</b>
<b>80</b>	<b>Boxes and Minipages</b>	<b>454</b>
80.1	Counters and lengths . . . . .	455
80.2	Footnote handling . . . . .	456
80.3	Minipage handling . . . . .	456
80.4	Parbox, mbox, makebox, framebox, fbox, raisebox . . . . .	459
<b>81</b>	<b>Direct formatting</b>	<b>465</b>
<b>82</b>	<b>Skips, spaces, font sizes</b>	<b>471</b>
<b>83</b>	<b>\phantomsection</b>	<b>479</b>
<b>84</b>	<b>\LaTeX and other logos</b>	<b>479</b>
84.1	HTML logos . . . . .	480
84.2	Print logos . . . . .	482

---

<b>85</b>	<b>\AtBeginDocument, \AtEndDocument</b>	<b>482</b>
<b>86</b>	<b>Koma-script</b>	<b>482</b>
<b>87</b>	<b>Memoir</b>	<b>483</b>
<b>88</b>	<b>Trademarks</b>	<b>484</b>
<b>2</b>	<b>lwarp-a4.sty</b>	<b>485</b>
<b>89</b>	<b>a4</b>	<b>485</b>
<b>3</b>	<b>lwarp-a4wide.sty</b>	<b>485</b>
<b>90</b>	<b>a4wide</b>	<b>485</b>
<b>4</b>	<b>lwarp-a5comb.sty</b>	<b>485</b>
<b>91</b>	<b>a5comb</b>	<b>485</b>
<b>5</b>	<b>lwarp-abstract.sty</b>	<b>485</b>
<b>92</b>	<b>abstract</b>	<b>485</b>
<b>6</b>	<b>lwarp-acro.sty</b>	<b>488</b>
<b>93</b>	<b>acro</b>	<b>488</b>
<b>7</b>	<b>lwarp-acronym.sty</b>	<b>489</b>
<b>94</b>	<b>acronym</b>	<b>489</b>
<b>8</b>	<b>lwarp-adjmulticol.sty</b>	<b>491</b>
<b>95</b>	<b>adjmulticol</b>	<b>491</b>
<b>9</b>	<b>lwarp-addlines.sty</b>	<b>492</b>
<b>96</b>	<b>addlines</b>	<b>492</b>

<b>10</b>	<b>lwarp-afterpage.sty</b>	<b>492</b>
97	afterpage	492
<b>11</b>	<b>lwarp-algorithmicx.sty</b>	<b>492</b>
98	algorithmicx	492
<b>12</b>	<b>lwarp-alltt.sty</b>	<b>493</b>
99	alltt	493
<b>13</b>	<b>lwarp-amsthm.sty</b>	<b>494</b>
100	amsthm	494
<b>14</b>	<b>lwarp-anonchap.sty</b>	<b>498</b>
101	anonchap	498
<b>15</b>	<b>lwarp-anysize.sty</b>	<b>498</b>
102	anysize	498
<b>16</b>	<b>lwarp-appendix.sty</b>	<b>499</b>
103	appendix	499
<b>17</b>	<b>lwarp-arabicfront.sty</b>	<b>499</b>
104	arabicfront	499
<b>18</b>	<b>lwarp-array.sty</b>	<b>500</b>
105	array	500
<b>19</b>	<b>lwarp-atbegshi.sty</b>	<b>500</b>
106	atbegshi	500

---

<b>20</b>	<b>lwarp-authblk.sty</b>	<b>501</b>
107	authblk	501
<b>21</b>	<b>lwarp-axodraw2.sty</b>	<b>502</b>
108	axodraw2	502
<b>22</b>	<b>lwarp-backref.sty</b>	<b>502</b>
109	backref	502
<b>23</b>	<b>lwarp-balance.sty</b>	<b>502</b>
110	balance	502
<b>24</b>	<b>lwarp-bigdelim.sty</b>	<b>503</b>
111	bigdelim	503
<b>25</b>	<b>lwarp-bigstrut.sty</b>	<b>504</b>
112	bigstrut	504
<b>26</b>	<b>lwarp-blowup.sty</b>	<b>505</b>
113	blowup	505
<b>27</b>	<b>lwarp-bookmark.sty</b>	<b>505</b>
114	bookmark	505
<b>28</b>	<b>lwarp-booktabs.sty</b>	<b>505</b>
115	booktabs	505
<b>29</b>	<b>lwarp-boxedminipage.sty</b>	<b>506</b>
116	boxedminipage	506

---

<b>30</b>	<b>lwarp-boxedminipage2e.sty</b>	<b>506</b>
117	boxedminipage2e	506
<b>31</b>	<b>lwarp-breakurl.sty</b>	<b>507</b>
118	breakurl	507
<b>32</b>	<b>lwarp-bytefield.sty</b>	<b>507</b>
119	bytefield	507
<b>33</b>	<b>lwarp-cancel.sty</b>	<b>508</b>
120	cancel	508
<b>34</b>	<b>lwarp-caption.sty</b>	<b>509</b>
121	caption	509
<b>35</b>	<b>lwarp-caption2.sty</b>	<b>512</b>
122	caption2	512
<b>36</b>	<b>lwarp-ccaption.sty</b>	<b>512</b>
123	ccaption	512
<b>37</b>	<b>lwarp-changebar.sty</b>	<b>512</b>
124	changebar	512
<b>38</b>	<b>lwarp-changepage.sty</b>	<b>513</b>
125	changepage	513
<b>39</b>	<b>lwarp-chngpage.sty</b>	<b>513</b>

---

<b>126</b>	<b>chngepage</b>	<b>513</b>
<b>40</b>	<b>lwarp-chappg.sty</b>	<b>514</b>
127	chappg	514
<b>41</b>	<b>lwarp-chapterbib.sty</b>	<b>514</b>
128	chapterbib	514
<b>42</b>	<b>lwarp-cite.sty</b>	<b>514</b>
129	cite	514
<b>43</b>	<b>lwarp-color.sty</b>	<b>515</b>
130	color	515
<b>44</b>	<b>lwarp-colortbl.sty</b>	<b>515</b>
131	colortbl	515
<b>45</b>	<b>lwarp-continue.sty</b>	<b>517</b>
132	continue	517
<b>46</b>	<b>lwarp-crop.sty</b>	<b>517</b>
133	crop	517
<b>47</b>	<b>lwarp-cuted.sty</b>	<b>518</b>
134	cuted	518
<b>48</b>	<b>lwarp-cutwin.sty</b>	<b>518</b>
135	cutwin	518



---

<b>49</b>	<b>lwarp-dblfloatfix.sty</b>	<b>519</b>
136	dblfloatfix	519
<b>50</b>	<b>lwarp-dblfnote.sty</b>	<b>519</b>
137	dblfnote	519
<b>51</b>	<b>lwarp-dcolumn.sty</b>	<b>520</b>
138	dcolumn	520
<b>52</b>	<b>lwarp-diagbox.sty</b>	<b>520</b>
139	diagbox	520
<b>53</b>	<b>lwarp-draftwatermark.sty</b>	<b>522</b>
140	draftwatermark	522
<b>54</b>	<b>lwarp-easy-todo.sty</b>	<b>522</b>
141	easy-todo	522
<b>55</b>	<b>lwarp-ebook.sty</b>	<b>523</b>
142	ebook	523
<b>56</b>	<b>lwarp-ellipsis.sty</b>	<b>523</b>
143	ellipsis	523
<b>57</b>	<b>lwarp-emptypage.sty</b>	<b>524</b>
144	emptypage	524
<b>58</b>	<b>lwarp-endfloat.sty</b>	<b>524</b>
145	endfloat	524

---

<b>59</b>	<b>lwarp-endheads.sty</b>	<b>524</b>
146	endheads	524
<b>60</b>	<b>lwarp-endnotes.sty</b>	<b>525</b>
147	endnotes	525
<b>61</b>	<b>lwarp-enumerate.sty</b>	<b>526</b>
148	enumerate	526
<b>62</b>	<b>lwarp-enumitem.sty</b>	<b>526</b>
149	enumitem	526
<b>63</b>	<b>lwarp-epigraph.sty</b>	<b>527</b>
150	epigraph	527
<b>64</b>	<b>lwarp-epstopdf.sty</b>	<b>528</b>
151	epstopdf	528
<b>65</b>	<b>lwarp-eso-pic.sty</b>	<b>529</b>
152	eso-pic	529
<b>66</b>	<b>lwarp-everypage.sty</b>	<b>529</b>
153	everypage	529
<b>67</b>	<b>lwarp-everyshi.sty</b>	<b>530</b>
154	everyshi	530
<b>68</b>	<b>lwarp-extramarks.sty</b>	<b>530</b>
155	extramarks	530

---

<b>69</b>	<b>lwarp-fancybox.sty</b>	<b>531</b>
156	fancybox	531
<b>70</b>	<b>lwarp-fancyheadings.sty</b>	<b>535</b>
157	fancyheadings	535
<b>71</b>	<b>lwarp-fancyhdr.sty</b>	<b>535</b>
158	fancyhdr	535
<b>72</b>	<b>lwarp-fancyref.sty</b>	<b>536</b>
159	fancyref	536
<b>73</b>	<b>lwarp-fancyvrb.sty</b>	<b>537</b>
160	fancyvrb	537
<b>74</b>	<b>lwarp-figcaps.sty</b>	<b>542</b>
161	figcaps	542
<b>75</b>	<b>lwarp-figsize.sty</b>	<b>543</b>
162	figsize	543
<b>76</b>	<b>lwarp-fix2col.sty</b>	<b>543</b>
163	fix2col	543
<b>77</b>	<b>lwarp-fixme.sty</b>	<b>543</b>
164	fixme	543
<b>78</b>	<b>lwarp-fixmetodonotes.sty</b>	<b>545</b>

---

<b>165</b>	<b>fixmetodonotes</b>	<b>545</b>
<b>79</b>	<b>lwarp-flafter.sty</b>	<b>545</b>
166	flafter	545
<b>80</b>	<b>lwarp-float.sty</b>	<b>546</b>
167	float and \newfloat	546
<b>81</b>	<b>lwarp-floatflt.sty</b>	<b>547</b>
168	floatflt	547
<b>82</b>	<b>lwarp-floatpag.sty</b>	<b>548</b>
169	floatpag	548
<b>83</b>	<b>lwarp-floatrow.sty</b>	<b>548</b>
170	floatrow	548
<b>84</b>	<b>lwarp-fltrace.sty</b>	<b>554</b>
171	fltrace	554
<b>85</b>	<b>lwarp-flushend.sty</b>	<b>554</b>
172	flushend	554
<b>86</b>	<b>lwarp-fncychap.sty</b>	<b>555</b>
173	fncychap	555
<b>87</b>	<b>lwarp-fnlineno.sty</b>	<b>555</b>
174	fnlineno	555

---

<b>88</b>	<b>lwarp-fnpos.sty</b>	<b>556</b>
175	fnpos	556
<b>89</b>	<b>lwarp-fontenc.sty</b>	<b>556</b>
176	fontenc	556
<b>90</b>	<b>lwarp-fontspec.sty</b>	<b>556</b>
177	fontspec	556
<b>91</b>	<b>lwarp-footmisc.sty</b>	<b>557</b>
178	footmisc	557
<b>92</b>	<b>lwarp-footnote.sty</b>	<b>558</b>
179	footnote	558
<b>93</b>	<b>lwarp-footnotehyper.sty</b>	<b>559</b>
180	footnotehyper	559
<b>94</b>	<b>lwarp-footnpag.sty</b>	<b>559</b>
181	footnpag	559
<b>95</b>	<b>lwarp-framed.sty</b>	<b>559</b>
182	framed	559
<b>96</b>	<b>lwarp-ftnright.sty</b>	<b>562</b>
183	ftnright	562
<b>97</b>	<b>lwarp-fullpage.sty</b>	<b>562</b>
184	fullpage	562

<b>98</b>	<b>lwarp-fullwidth.sty</b>	<b>562</b>
185	fullwidth	562
<b>99</b>	<b>lwarp-fwlw.sty</b>	<b>563</b>
186	fwlw	563
<b>100</b>	<b>lwarp-geometry.sty</b>	<b>563</b>
187	geometry	563
<b>101</b>	<b>lwarp-glossaries.sty</b>	<b>564</b>
188	glossaries	564
<b>102</b>	<b>lwarp-graphics.sty</b>	<b>565</b>
189	graphics	565
189.1	Graphics extensions . . . . .	565
189.2	Length conversions and graphics options . . . . .	566
189.3	Printing HTML styles . . . . .	568
189.4	\includegraphics . . . . .	569
189.5	Boxes . . . . .	574
<b>103</b>	<b>lwarp-graphicx.sty</b>	<b>576</b>
190	graphicx	576
<b>104</b>	<b>lwarp-grffile.sty</b>	<b>577</b>
191	grffile	577
<b>105</b>	<b>lwarp-hang.sty</b>	<b>577</b>
192	hang	577
<b>106</b>	<b>lwarp-hanging.sty</b>	<b>579</b>

---

<b>193</b>	<b>hanging</b>	<b>579</b>
<b>107</b>	<b>lwarp-hypcap.sty</b>	<b>580</b>
194	hypcap	580
<b>108</b>	<b>lwarp-hypdestopt.sty</b>	<b>580</b>
195	hypdestopt	580
<b>109</b>	<b>lwarp-hypernat.sty</b>	<b>580</b>
196	hypernat	580
<b>110</b>	<b>lwarp-hyperref.sty</b>	<b>580</b>
197	hyperref	580
<b>111</b>	<b>lwarp-hyperxmp.sty</b>	<b>585</b>
198	hyperxmp	585
<b>112</b>	<b>lwarp-hyphenat.sty</b>	<b>585</b>
199	hyphenat	585
<b>113</b>	<b>lwarp-idxlayout.sty</b>	<b>586</b>
200	idxlayout	586
<b>114</b>	<b>lwarp-ifoddpage.sty</b>	<b>587</b>
201	ifoddpage	587
<b>115</b>	<b>lwarp-indentfirst.sty</b>	<b>587</b>
202	indentfirst	587

---

<b>116</b>	<b>lwarp-inputenc.sty</b>	<b>588</b>
203	inputenc	588
<b>117</b>	<b>lwarp-keyfloat.sty</b>	<b>588</b>
204	keyfloat	588
<b>118</b>	<b>lwarp-layout.sty</b>	<b>590</b>
205	layout	590
<b>119</b>	<b>lwarp-letterspace.sty</b>	<b>591</b>
206	letterspace	591
<b>120</b>	<b>lwarp-lettrine.sty</b>	<b>591</b>
207	lettrine	591
<b>121</b>	<b>lwarp-lineno.sty</b>	<b>592</b>
208	lineno	592
<b>122</b>	<b>lwarp-lips.sty</b>	<b>595</b>
209	lips	595
<b>123</b>	<b>lwarp-listings.sty</b>	<b>595</b>
210	listings	595
<b>124</b>	<b>lwarp-longtable.sty</b>	<b>599</b>
211	longtable	599
<b>125</b>	<b>lwarp-lscape.sty</b>	<b>601</b>



<b>212</b>	<b>lscape</b>	<b>601</b>
<b>126</b>	<b>lwarp-ltcaption.sty</b>	<b>601</b>
<b>213</b>	<b>ltcaption</b>	<b>601</b>
<b>127</b>	<b>lwarp-ltxtable.sty</b>	<b>602</b>
<b>214</b>	<b>ltxtable</b>	<b>602</b>
<b>128</b>	<b>lwarp-luacolor.sty</b>	<b>602</b>
<b>215</b>	<b>luacolor</b>	<b>602</b>
<b>129</b>	<b>lwarp-luatodonotes.sty</b>	<b>602</b>
<b>216</b>	<b>luatodonotes</b>	<b>602</b>
<b>130</b>	<b>lwarp-marginfit.sty</b>	<b>604</b>
<b>217</b>	<b>marginfit</b>	<b>604</b>
<b>131</b>	<b>lwarp-marginfix.sty</b>	<b>605</b>
<b>218</b>	<b>marginfix</b>	<b>605</b>
<b>132</b>	<b>lwarp-marginnote.sty</b>	<b>605</b>
<b>219</b>	<b>marginnote</b>	<b>605</b>
<b>133</b>	<b>lwarp-mcaption.sty</b>	<b>606</b>
<b>220</b>	<b>mcaption</b>	<b>606</b>
<b>134</b>	<b>lwarp-mdframed.sty</b>	<b>606</b>
<b>221</b>	<b>mdframed</b>	<b>606</b>
221.1	Limitations . . . . .	606

221.2	Package loading	607
221.3	Patches	607
221.4	Initial setup	608
221.5	Color and length HTML conversion	608
221.6	Environment encapsulation	609
221.7	Mdframed environment	610
221.8	Titles and subtitles	611
221.9	New environments	614
<b>135</b>	<b>lwarp-memhfixc.sty</b>	<b>617</b>
222	memhfixc	617
<b>136</b>	<b>lwarp-metalogo.sty</b>	<b>617</b>
223	metalogo	617
<b>137</b>	<b>lwarp-microtype.sty</b>	<b>617</b>
224	microtype	617
<b>138</b>	<b>lwarp-midfloat.sty</b>	<b>618</b>
225	midfloat	618
<b>139</b>	<b>lwarp-midpage.sty</b>	<b>619</b>
226	midpage	619
<b>140</b>	<b>lwarp-morefloats.sty</b>	<b>619</b>
227	morefloats	619
<b>141</b>	<b>lwarp-moreverb.sty</b>	<b>619</b>
228	moreverb	619
<b>142</b>	<b>lwarp-morewrites.sty</b>	<b>621</b>

---

<b>229</b>	<b>morewrites</b>	<b>621</b>
<b>143</b>	<b>lwarp-mparhack.sty</b>	<b>621</b>
<b>230</b>	<b>mparhack</b>	<b>621</b>
<b>144</b>	<b>lwarp-multicol.sty</b>	<b>621</b>
<b>231</b>	<b>multicol</b>	<b>621</b>
<b>145</b>	<b>lwarp-multirow.sty</b>	<b>623</b>
<b>232</b>	<b>multirow</b>	<b>623</b>
232.1	Multirow . . . . .	623
232.2	Combined multicolumn and multirow . . . . .	625
<b>146</b>	<b>lwarp-multitoc.sty</b>	<b>626</b>
<b>233</b>	<b>multitoc</b>	<b>626</b>
<b>147</b>	<b>lwarp-nameref.sty</b>	<b>626</b>
<b>234</b>	<b>nameref</b>	<b>626</b>
<b>148</b>	<b>lwarp-natbib.sty</b>	<b>626</b>
<b>235</b>	<b>natbib</b>	<b>626</b>
<b>149</b>	<b>lwarp-needspace.sty</b>	<b>627</b>
<b>236</b>	<b>needspace</b>	<b>627</b>
<b>150</b>	<b>lwarp-newclude.sty</b>	<b>628</b>
<b>237</b>	<b>newclude</b>	<b>628</b>
<b>151</b>	<b>lwarp-newunicodechar.sty</b>	<b>628</b>
<b>238</b>	<b>newunicodechar</b>	<b>628</b>

<b>152</b>	<b>lwarp-nextpage.sty</b>	<b>628</b>
239	nextpage	628
<b>153</b>	<b>lwarp-nonfloat.sty</b>	<b>629</b>
240	nonfloat	629
<b>154</b>	<b>lwarp-nonumonpart.sty</b>	<b>629</b>
241	nonumonpart	629
<b>155</b>	<b>lwarp-nopageno.sty</b>	<b>629</b>
242	nopageno	629
<b>156</b>	<b>lwarp-nowidow.sty</b>	<b>629</b>
243	nowidow	629
<b>157</b>	<b>lwarp-ntheorem.sty</b>	<b>630</b>
244	ntheorem	630
244.1	Limitations	630
244.2	Options	631
244.3	Remembering the theorem style	631
244.4	HTML cross-referencing	635
244.5	\newtheoremstyle	635
244.6	Standard styles	635
244.7	Additional objects	637
244.8	Renewed standard configuration	637
244.9	amsthm option	639
244.10	Ending a theorem	641
244.11	\NoEndMark	641
244.12	List-of	641
244.13	Symbols	642
244.14	Cross-referencing	642
<b>158</b>	<b>lwarp-overpic.sty</b>	<b>642</b>
245	overpic	642

---

<b>159</b>	<b>lwarp-pagegrid.sty</b>	<b>643</b>
246	pagegrid	643
<b>160</b>	<b>lwarp-pagenote.sty</b>	<b>643</b>
247	pagenote	643
<b>161</b>	<b>lwarp-pagesel.sty</b>	<b>644</b>
248	pagesel	644
<b>162</b>	<b>lwarp-paralist.sty</b>	<b>644</b>
249	paralist	644
<b>163</b>	<b>lwarp-parskip.sty</b>	<b>645</b>
250	parskip	645
<b>164</b>	<b>lwarp-pbox.sty</b>	<b>645</b>
251	pbox	645
<b>165</b>	<b>lwarp-pdfscape.sty</b>	<b>645</b>
252	pdfscape	645
<b>166</b>	<b>lwarp-pdfrender.sty</b>	<b>646</b>
253	pdfrender	646
<b>167</b>	<b>lwarp-pdfsync.sty</b>	<b>646</b>
254	pdfsync	646
<b>168</b>	<b>lwarp-pfnote.sty</b>	<b>646</b>

---

<b>255</b>	<b>pfnote</b>	<b>646</b>
<b>169</b>	<b>lwarp-phfqit.sty</b>	<b>647</b>
256	phfqit	647
<b>170</b>	<b>lwarp-placeins.sty</b>	<b>647</b>
257	placeins	647
<b>171</b>	<b>lwarp-prelim2e.sty</b>	<b>648</b>
258	prelim2e	648
<b>172</b>	<b>lwarp-prettyref.sty</b>	<b>648</b>
259	prettyref	648
<b>173</b>	<b>lwarp-preview.sty</b>	<b>648</b>
260	preview	648
<b>174</b>	<b>lwarp-quotchap.sty</b>	<b>649</b>
261	quotchap	649
<b>175</b>	<b>lwarp-ragged2e.sty</b>	<b>650</b>
262	ragged2e	650
<b>176</b>	<b>lwarp-realscripts.sty</b>	<b>651</b>
263	realscripts	651
<b>177</b>	<b>lwarp-relsize.sty</b>	<b>652</b>
264	relsize	652

---

<b>178</b>	<b>lwarp-resizegather.sty</b>	<b>653</b>
265	resizegather	653
<b>179</b>	<b>lwarp-romanbar.sty</b>	<b>653</b>
266	romanbar	653
<b>180</b>	<b>lwarp-romanbarpagenumber.sty</b>	<b>654</b>
267	romanbarpagenumber	654
<b>181</b>	<b>lwarp-rotating.sty</b>	<b>654</b>
268	rotating	654
<b>182</b>	<b>lwarp-rotfloat.sty</b>	<b>654</b>
269	rotfloat	654
<b>183</b>	<b>lwarp-savetrees.sty</b>	<b>655</b>
270	savetrees	655
<b>184</b>	<b>lwarp-scalefnt.sty</b>	<b>656</b>
271	scalefnt	656
<b>185</b>	<b>lwarp-schemata.sty</b>	<b>656</b>
272	schemata	656
<b>186</b>	<b>lwarp-scrextend.sty</b>	<b>657</b>
273	scrextend	657
<b>187</b>	<b>lwarp-scrhack.sty</b>	<b>660</b>
274	scrhack	660

<b>188</b>	<b>lwarp-scrlayer.sty</b>	<b>660</b>
275	scrlayer	660
<b>189</b>	<b>lwarp-scrlayer-notecolumn.sty</b>	<b>662</b>
276	scrlayer-notecolumn	662
<b>190</b>	<b>lwarp-scrlayer-scrpage.sty</b>	<b>662</b>
277	scrlayer-scrpage	662
<b>191</b>	<b>lwarp-section.sty</b>	<b>663</b>
278	section	663
<b>192</b>	<b>lwarp-sectionbreak.sty</b>	<b>664</b>
279	sectionbreak	664
<b>193</b>	<b>lwarp-sectsty.sty</b>	<b>665</b>
280	sectsty	665
<b>194</b>	<b>lwarp-setspace.sty</b>	<b>665</b>
281	setspace	665
<b>195</b>	<b>lwarp-shadow.sty</b>	<b>666</b>
282	shadow	666
<b>196</b>	<b>lwarp-showidx.sty</b>	<b>667</b>
283	showidx	667
<b>197</b>	<b>lwarp-showkeys.sty</b>	<b>667</b>
284	showkeys	667



---

<b>198</b>	<b>lwarp-sidecap.sty</b>	<b>667</b>
285	sidecap	667
<b>199</b>	<b>lwarp-sidenotes.sty</b>	<b>668</b>
286	sidenotes	668
<b>200</b>	<b>lwarp-siunitx.sty</b>	<b>670</b>
287	siunitx	670
<b>201</b>	<b>lwarp-soul.sty</b>	<b>675</b>
288	soul	675
<b>202</b>	<b>lwarp-soulpos.sty</b>	<b>677</b>
289	soulpos	677
<b>203</b>	<b>lwarp-soulutf8.sty</b>	<b>678</b>
290	soulutf8	678
<b>204</b>	<b>lwarp-stabular.sty</b>	<b>678</b>
291	stabular	678
<b>205</b>	<b>lwarp-stfloats.sty</b>	<b>679</b>
292	stfloats	679
<b>206</b>	<b>lwarp-subfig.sty</b>	<b>679</b>
293	subfig	679
<b>207</b>	<b>lwarp-subfigure.sty</b>	<b>684</b>
294	subfigure	684

<b>208</b>	<b>lwarp-supertabular.sty</b>	<b>685</b>
295	supertabular	685
<b>209</b>	<b>lwarp-syntonly.sty</b>	<b>686</b>
296	syntonly	686
<b>210</b>	<b>lwarp-tabls.sty</b>	<b>687</b>
297	tabls	687
<b>211</b>	<b>lwarp-tabularx.sty</b>	<b>687</b>
298	tabularx	687
<b>212</b>	<b>lwarp-tabulary.sty</b>	<b>688</b>
299	tabulary	688
<b>213</b>	<b>lwarp-textarea.sty</b>	<b>688</b>
300	textarea	688
<b>214</b>	<b>lwarp-textcomp.sty</b>	<b>689</b>
301	textcomp	689
301.1	Limitations . . . . .	689
301.2	Package loading . . . . .	689
301.3	Remembering original defintions . . . . .	689
301.4	HTML symbols . . . . .	690
301.4.1	pdf $\TeX$ symbols . . . . .	690
301.4.2	$\text{\XeTeX}$ and Lua $\TeX$ symbols . . . . .	690
301.5	HTML dicritics . . . . .	691
301.6	Inside a lateximage . . . . .	692
<b>215</b>	<b>lwarp-textfit.sty</b>	<b>692</b>
302	textfit	692

<b>216</b>	<b>lwarp-textpos.sty</b>	<b>693</b>
303	textpos	693
<b>217</b>	<b>lwarp-theorem.sty</b>	<b>694</b>
304	theorem	694
304.1	Remembering the theorem style . . . . .	694
304.2	CSS patches . . . . .	696
<b>218</b>	<b>lwarp-threeparttable.sty</b>	<b>698</b>
305	threeparttable	698
<b>219</b>	<b>lwarp-tikz.sty</b>	<b>699</b>
306	tikz	699
<b>220</b>	<b>lwarp-titleps.sty</b>	<b>700</b>
307	titleps	700
<b>221</b>	<b>lwarp-titleref.sty</b>	<b>703</b>
308	titleref	703
<b>222</b>	<b>lwarp-titlesec.sty</b>	<b>703</b>
309	titlesec	703
<b>223</b>	<b>lwarp-titletoc.sty</b>	<b>705</b>
310	titletoc	705
<b>224</b>	<b>lwarp-titling.sty</b>	<b>707</b>
311	titling	707
<b>225</b>	<b>lwarp-tocbasic.sty</b>	<b>712</b>

312	<b>tocbasic</b>	712
226	<b>lwarp-tocbibind.sty</b>	712
313	<b>tocbibind</b>	712
227	<b>lwarp-tocloft.sty</b>	715
314	<b>tocloft</b>	715
228	<b>lwarp-tocstyle.sty</b>	721
315	<b>tocstyle</b>	721
229	<b>lwarp-todo.sty</b>	722
316	<b>todo</b>	722
230	<b>lwarp-todonotes.sty</b>	723
317	<b>todonotes</b>	723
231	<b>lwarp-transparent.sty</b>	725
318	<b>transparent</b>	725
232	<b>lwarp-trivfloat.sty</b>	725
319	<b>trivfloat</b>	725
	319.1 Combining \newfloat, \trivfloat, and algorithmicx . . . . .	726
233	<b>lwarp-turnthepage.sty</b>	726
320	<b>turnthepage</b>	726
234	<b>lwarp-typearea.sty</b>	727
321	<b>typearea</b>	727

---

<b>235</b>	<b>lwarp-ulem.sty</b>	<b>727</b>
322	ulem	727
<b>236</b>	<b>lwarp-upref.sty</b>	<b>730</b>
323	upref	730
<b>237</b>	<b>lwarp-verse.sty</b>	<b>730</b>
324	verse	730
<b>238</b>	<b>lwarp-vertbars.sty</b>	<b>732</b>
325	vertbars	732
<b>239</b>	<b>lwarp-vmargin.sty</b>	<b>732</b>
326	vmargin	732
<b>240</b>	<b>lwarp-vwcol.sty</b>	<b>733</b>
327	vwcol	733
<b>241</b>	<b>lwarp-wallpaper.sty</b>	<b>735</b>
328	wallpaper	735
<b>242</b>	<b>lwarp-watermark.sty</b>	<b>736</b>
329	watermark	736
<b>243</b>	<b>lwarp-wrapfig.sty</b>	<b>736</b>
330	wrapfig	736
<b>244</b>	<b>lwarp-xcolor.sty</b>	<b>738</b>

<b>331</b>	<b>xcolor</b>	<b>738</b>
331.1	Limitations	738
331.2	Xcolor definitions: location and timing	738
331.3	Package loading	740
331.4	Remembering and restoring original definitions	741
331.5	HTML color style	741
331.6	HTML border	743
331.7	High-level macros	743
331.8	Row colors	747
<b>245</b>	<b>lwarp-xfrac.sty</b>	<b>748</b>
332	xfrac	748
<b>246</b>	<b>lwarp-xltxtra.sty</b>	<b>751</b>
333	xltxtra	751
<b>247</b>	<b>lwarp-xmpinl.sty</b>	<b>752</b>
334	xmpinl	752
<b>248</b>	<b>lwarp-xtab.sty</b>	<b>752</b>
335	xtab	752
<b>249</b>	<b>lwarp-xurl.sty</b>	<b>754</b>
336	xurl	754
<b>250</b>	<b>lwarp-xy.sty</b>	<b>754</b>
337	xy	754
<b>251</b>	<b>lwarp-zwpage layout.sty</b>	<b>755</b>
338	zwpage layout	755
<b>252</b>	<b>lwarp-patch-komascript.sty</b>	<b>756</b>

<b>339 patch-komascript</b>	<b>756</b>
-----------------------------	------------

<b>253 lwarp-patch-memoir.sty</b>	<b>758</b>
-----------------------------------	------------

<b>340 patch-memoir</b>	<b>758</b>
340.1 Packages . . . . .	759
340.2 Preliminary setup . . . . .	760
340.3 Laying out the page . . . . .	761
340.4 Text and fonts . . . . .	763
340.5 Titles . . . . .	764
340.6 Abstracts . . . . .	764
340.7 Document divisions . . . . .	764
340.8 Pagination and headers . . . . .	767
340.9 Paragraphs and lists . . . . .	769
340.10 Contents lists . . . . .	769
340.11 Floats and captions . . . . .	774
340.12 Page notes . . . . .	779
340.13 Decorative text . . . . .	781
340.14 Poetry . . . . .	781
340.15 Boxes, verbatims and files . . . . .	781
340.16 Cross referencing . . . . .	782
340.17 Back matter . . . . .	783
340.18 Miscellaneous . . . . .	783
340.19 Final patchwork . . . . .	784

<b>Change History and Index</b>	<b>785</b>
---------------------------------	------------

## List of Figures

1	tutorial.tex listing	70
---	----------------------	----

## List of Tables

1	$\text{\LaTeX}$ -HTML generation — lwarp package — Supported features	54
2	Required software programs	62
3	Files created along with the print version	72
4	Package options	85
5	HTML settings	87
6	Section HTML headings for word-processor conversion	131
7	Section depths and HTML headings	141
8	Tabular baseline	337
9	Tabular HTML column conversions	338
10	Cross-referencing data structures	380
11	Float data structures	390
12	AMStm package — CSS styling of theorems and proofs	494
13	Ntheorem package — CSS styling of theorems and proofs	630
14	Theorem package — CSS styling of theorems and proofs	694



## 2 Updates

The following is a summary of updates to lwarp, highlighting new features and any special changes which must be made due to improvements or modifications in lwarp itself.

For a detailed list of changes, see the Change History on page [785](#).

### v0.50:

#### svg math

- `svg math` and other `lateximages` now are converted to `svg` using parallel background tasks, utilizing all available CPU cores.
- Inline `svg math` image file names now are MD5 hashes made from their source  $\TeX$  code. Identical inline math expressions, such as multiple instance of  $\$x\$$ , now share a single image file. This reduces the number of images to store, transmit, process, and display. Each image file is only converted to `svg` a single time, and reused if it already exists. Display math and other forms of `svg` image such as `picture` and `Tikz` still use individual image files which are recreated each time `lwarpmk images` is run.
- Fixes: `SVG math` and/or `\underline` in a sectioning file name.
- Improved `svg display math` and tags.
- Improved `svg math` and `siunitx alt` tags.
- Improved `siunitx` units.
- Fix: `\ensuremath` with `MathJax` now creates a `lateximage`.
- Fix: `\centering`, etc. in `svg math`, `lateximage`, `Tikz`.

#### misc. fixes

- Fix: Made various macros robust, additionally fixing `authblk`.
- Fix: `ntheorem` if neither `standard` nor `amsthm` selected.
- Fix: `listings`: Improved column alignment.
- Fix: Load `fontspec` if necessary.

#### packages

- Added `xy`, `epstopdf`, `diagbox`, `pbox`, `bytefield`, `axodraw2`, `phfquit`, `schemata`, `dblfloatfix`, `nonfloat`, `morefloats`.

### v0.49:

#### tabular

- Added `xcolor \rowcolors`.
- Fix: `\noalign` inside a `tabular`.

#### math

- Fix: `\eqref` in a caption.

#### misc fixes

- Fix: Incorrect PDF font size changes caused occasional HTML corruption.
- Fix: `printlen` changes are now grouped for HTML output.

#### packages

- Added `vwcol`, `vertbars`, `hyphenat`, `lineno`, `fnlineno`, `figsize`, `hypdestopt`, `pagegrid`, `pdfrender`, `luacolor`, `resizegather`.


**v0.48:**

- docs**
  - Added some documentation regarding converting an existing document. See section 6.15.
- cleveref**
  - Updated compatibility for new cleveref v0.21.
- tabular**
  - Fix: Ignores optional tabular column arguments.
- minor updates**
  - Added `\leftline`, `\centerline`, `\rightline`.
  - Lists have improved font control via `\makelabel`.
  - Print-mode `lateximage` now boxed to the natural width of its multiline contents.
  - `abstract` now allows an optional name, as required by some classes.
- math**
  - Fix: Improved spacing, `\mbox`, and font sizes with `svg math`, `Tikz`.
  - `siunitx`: Improved `svg math`, fraction compatibility, color output.
- misc. fixes**
  - Fix: LOF/LOT links.
  - Fix: Virtual page size grouping caused excessive PDF page breaks.
  - Fix: Parsing similar package names in a single `\usepackage`.
  - Fix: Adapts to classes without `\part`.
  - Fix: `\newline` in `\title` was causing `<br>` in window title.
  - Fix: `\maketitle` with `\cr`, `\crrcr`, `\noalign`, for `IEEEtran` class.
  - Fix: `xfrac` neutralized `BlockClass` and others.
  - Fix: `todonotes` and `luatodonotes`: Improved `\todototoc`.
- packages**
  - Added `colortbl`, `chapterbib`, `acro`, `acronym`, `hypernat`, `hypcap`, `stfloats`, `vmargin`, `fancyheadings`.
  - `fancyref`: Now directly supported.

**v0.47:**

- math**
  - Improved `svg math` baseline and sizing.
  - Fixes: `svgmath` in captions, subcaptions, `\nameref`.
  - Fixes: Line wrap at hyphen in HTML output.
- packages**
  - Added `endheads`, `multitoc`, `sectionbreak`, `blowup`, `xurl`.

**v0.46:**

-  **name change**
  - `\PrintStack` changed to `\LWRPrintStack`.
- misc. fixes**
  - Fix: Empty lines between tabular rows.
  - Fix: Stack unnesting.
  - Fix: SVG math and `lateximages` in numerous situations.
  - Fix: Spaces in `\usepackage`.
  - Fix: Now allows `MATHJAX` inside `verse`.

**v0.45:**

- docs
  - Improved MikTeX install instructions.
  - Improved graphics and epstopdf instructions.
  - Updates to the [Introduction](#).
- memoir
  - Added memoir, memhfixc. See section [8.12](#).
- cross-references
  - Fix: Now allows underscores in labels.
  - Fix: `\_` and `\<blank>` in section/file names.
- math
  - Fix: Now allows MATHJAX inside tabbing.
- bibliography
  - Fix: Bibliography `\em` names.
  - Added cite, natbib, backref. (Also works as-is with biblatex.)
- misc. fixes
  - Fix: Empty lines between tabular rows.
  - Fix: “Improper `\prevdepth`” with minipages, lists.
  - Fix: Incorrect svg math and lateximages with subfig.
  - Fix: Lateximages from incorrect pages with Mathjax.
  - Fix: Missing sidetoc if using listings.
  - Fix: Added an array emulation package.
- packages
  - Added subfigure, prettyref, hanging, midpage, flafter, fltrace, changebar, endfloat, continue, fwlw, turnthepage, footnpag, pagesel, textfit, titleref.

**v0.44:**

- koma-script
  - Added koma-script classes (except scrlltr2, scrjura).
  - Added scrextend, sclayer, sclayer-notecolumn, sclayer-scrpage, scrhack, tocstyle, tocbasic.
- HTML title and author
  - Added `\HTMLTitle`. Fixed web page title if `\HTMLTitle` empty and no `\title` given and not using titling package.
  - Fixed web page author if `\HTMLauthor` is empty and `\author` is not given.
- encodings
  - If using pdflatex, automatically loads T1 and UTF8 encodings. (Additional fontenc encodings may be loaded after lwarp.)
- lists
  - Added list and trivlist environments, hang.
- tabular
  - Fix: `\multicolumn` alignment if formatting for a word processor.
  - Added ltxtable.
- math
  - Fix: MATHJAX combined with lateximages.
  - algorithmicx: Improved comment symbol and floating.
- packages
  - Completed todonotes and luatodonotes.
  - Added todo, easy-todo, fixmetodonotes, fixme.
  - Added soulutf8, soulpos, cancel.

- Added section, fancyref, ifoddpage.
- Added preview, atbegshi, watermark.
- Improved tocloft \newlistof and \newlistentry.

**v0.43:**

## footnotes

## sectioning

## tabular

## math

## packages

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

**v0.42:**Support T<sub>E</sub>X!

## word-processor conversion

⚠ name change  
⚠ name change

## tabbing

## overpic

## math

- Added T<sub>E</sub>X development support page, [Supporting T<sub>E</sub>X development](#).
- 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 mini-pages, at the table of contents, at the LOF and LOT, and whether to print math as  $\LaTeX$  source for copy/paste into the LibreOffice Writer TeXMaths extension.
  - Improved formatting for numerous objects. See section 10.
- Add: tabbing environment.
- Add: overpic package. See section 245.
- Fix: Text copy/paste of  $\mathcal{MS}$  math environment numbers and names.
- Improved \ensuremath.

- MATHJAX with siunitx: Updated script and documentation.
- textcomp: Improved `\interrobangdown`.
- realscripts: Fix for subscripts in a `lateximage`.
- morewrites: Enforces loading before `lwarp`.

**v0.41:**

- Added `tabular` vertical rules, subject to some limitations. See the rules section of section 8.9.

- Improved `booktabs`: Width and trim are honored.
- 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 232.2 on section 232.2.
- 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` and `graphicx` have been moved from the `lwarp` core, and are only loaded if requested with `\usepackage`.

- 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 8.7.

- `grffile` is now directly supported instead of emulated.

- Fix for `bigdelim`, and improved documentation. See section 111.

- Improved  $\TeX$  and `textcomp` symbols.

- Fix for  $\TeX$  logos and `\InlineClass`, etc. inside a `lateximage`.

- Fix for `xltxtra` with  $\TeX$ .

- Fixes for `tocbibind` with `\simplechapter`, etc.

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

- Added `\underline`.

- Added `adjmulticol`.

- Added `cuted`, `midfloat`.

- Added `pfnote`, `fnpos`, `dblfnote`.

- Added `stabular`, `tabls`.

- Added `sectsty`, `anonchap`, `quotchap`.

**v0.39:**

title pages

⚠ `\published and`  
`\subtitle`

⚠ load order

tabular

multi column/row cell

⚠ macros inside tabular

⚠ tabular defined inside  
another environment

tabular

margins

page layout

- Improved the titlepage HTML code, `\thanks` notes, and `\maketitle`. titling is no longer required, but is still supported. The `\published` and `\subtitle` fields are no longer provided, but `\AddSubtitlePublished` replicates them using titling. See section 58.8. `authblk` is added, and should be loaded before titling. See section 58.

- `\multirow` now supports the new optional `vpos` argument.
- Added `\multicolumnrow` for combined `\multicolumn` and `\multirow`. See section 232.2.

- Tabular special cases:

- Added `\TabularMacro` to mark custom macros inside tabular data cells, avoiding row corruption. See section 8.9.
- Added `\ResumeTabular` for use when a tabular environment is defined inside another environment. See section 8.9.

- Added `supertabular`, `xtab`, `bigstrut`, `bigdelim`.

- Added `fullwidth`.

- Added `addlines`, `anysize`, `a4`, `a4wide`, `a5comb`, `textarea`, `zwpgelayout`, `typearea`, `ebook`.

**v0.38:**

forced single-pass compile

starred sections

updated tutorial

packages

font size

page numbering

front &amp; back matter

- Added `lwarpmk print1` and `lwarpmk html1` actions to force a compile of the project a single time. Useful when multiple passes are not needed, or changes were not detected.
- Added `\ForceHTMLPage` and `\ForceHTMLTOC` to force a starred sectional unit onto its own HTML page and with its own toc entry. See section 8.5.1.
- Modified the tutorial to use the new `\ForceHTMLPage` and `\ForceHTMLTOC` macros.
- Added `appendix`, `tocbibind`, `fncychap`, `fix2col`.
- Added `relsize`, `scalegnt`.
- Added `realscripts`, `metalogo`, `xltxtra`.
- Added `grffile`, `romanbar`.
- Added `arabicfront`, `chappg`, `nonumonpart`, `nopageno`, `romanbarpagenumber`.
- Docs: Improved description of the use of front/back matter. See section 8.5.
- Fix: color requests `xcolor`.
- Fix: `\part` for article class.

**v0.37:**

- `\include` for HTML
  - `latexmk`
  - accents and symbols
  - `babel-french`
- `\include` now maintains independent .aux files for HTML versions.
  - `comment`, used by lwarp, now maintains independent cut files for print and HTML versions, helping latexmk to better know whether to recompile.
  - Improved support for  $\TeX$  accents, textcomp, siunitx symbols.
  - Improved babel-french handling for load order and ~ tilde.

**v0.36:**

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

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

**v0.34:**

- ⚠ Optional arguments
- BlockClass's optional argument has been moved in front of the mandatory argument:  
`BlockClass[style]{class}` (NEW)  
 instead of:  
`BlockClass{class}[style]` (OLD)  
 This change makes it more consistent with  $\TeX$  standards, and avoids problems with space between arguments.
- ⚠ Optional arguments
- Likewise, `\InlineClass`'s optional argument now comes before the mandatory arguments:  
`\InlineClass[style]{class}{text}`

spans with minipages

framing minipages

lateximage, svg math,  
tabular

eqnarray

verbatim packages

framing packages

list packages

babel-french

- Improved compatibility between spans, minipages, lists, frames, and math. Handles minipages and lists inside an HTML span, such as an `\fbox` containing a minipage, although with minimal HTML formatting. See section 8.3.3. `\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 8.3.5. `\fbox` and minipages now often work in `svg math` and `lateximages`. `MATHJAX` supports `\fbox`, but not `\fboxBlock` nor `fminipage`.
- Improved compatibility between `lateximage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, `booktabs`.
- Improved font control for `lateximages` and `svg math`.
- Added the `eqnarray` environments.
- `fancyvrb` is no longer required (preloaded), but is still supported.
- Added `verbatim` and `moreverb`.
- Added `fancybox`, `boxedminipage2e` and `shadow`.
- `enumitem` is no longer required, but is still supported.
- Added `enumerate` and `paralist`.
- `titleps` is no longer required, but is still supported.
- Added `crop`.
- Added `rotfloat`, `marginfit`, and several minor packages; see the change log.
- Adds fixed-width HTML spaces around punctuation when using `babel-french`. `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 64.4.1. This change is not required for the routine use of tables, but only when a table is defined inside another macro or environment, and while also using the `&` character inside the definition. This may include the use inside conditional expressions.
- Several math environments were incorrectly placed inline. Also, for `amsmath` with `svg math`, the `fleqn` option has been removed, resulting in improved spacing for aligned equations.
- Bug fixes; see the changelog.



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

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

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

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

**v0.30:**

### lwarp-newproject

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

To modify existing documents, remove from the document source:

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

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

### HTML setup changes.

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

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

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

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

- The file `lwarp_html.xdy` has been renamed `lwarp.xdy`. To update each document's project:
  1. Make the changes shown above.
  2. Recompile the document in print mode. This updates the project's configuration files, and also generates the new file `lwarp.xdy`.
  3. The old file `lwarp_html.xdy` may be deleted.
- The new lwarp package option `optnxdyFilename` may be used to tell `lwarpmk` to use a custom `.xdy` file instead of `lwarp.xdy`. See section 7.12.
- Improvements in index processing:
  - xindy's language is now used for index processing as well as glossary.
  - Print mode without `latexmk` now uses xindy instead of `makeindex`.
  - `texindy`/`xindy` usage depends on `pdflatex` vs `xelatex`, `lualatex`.
  - For `pdflatex` and `texindy`, the `-C utf8` option is used. This is supported in modern distributions, but a customized `lwarpmk.lua` may need to be created for use with older distributions.

#### v0.29:

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

#### v0.28:

- `\HTMLAuthor {<name>}` assigns HTML meta author if non-empty. Defaults to `\theauthor`.
- Boolean `HTMLDebugComments` controls whether HTML comments are added for closing `<div>`s, opening and closing sections, etc.
- Boolean `FormatEPUB` changes HTML output for easy EPUB conversion via an external program. Removes per-file headers, footers, and nav. Adds footnotes per chapter/section.
- Boolean `FormatWordProcessor` changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments. Name changed to `FormatWP` as of v0.42.

- Boolean `HTMLMarkFloats` adds text marks around floats only if `FormatWordProcessor`. These make it easier to identify float boundaries, which are to be manually converted to word-processor frames. Name changed to `WPMarkFloats` as of v0.42.
- Updated for the new `MATHJAX` CDN repository.
- Adds tabulary.
- Supports the options syntax for graphics.
- Improved index references, now pointing exactly to their target.
- Adds glossaries. `lwarpmk` is modified to add `printglossary` and `htmlglossary` actions.

### 3 Introduction

The lwarp project aims to allow a rich  $\text{\LaTeX}$  document to be converted to a reasonable HTML5 interpretation, with only minor intervention on the user's part. 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 shows that these new features degrade gracefully.

---

lwarp is a native  $\text{\LaTeX}$  package, and operates by either patching or emulating various functions. Source-level compatibility is a major goal, but occasional user intervention is required in certain cases.

As a package running directly in  $\text{\LaTeX}$ , lwarp has some advantages over other methods of HTML conversion.  $\text{\TeX}$  itself is still used, allowing a wider range of  $\text{\TeX}$  trickery to be understood. Lua expressions are still available with Lua $\text{\TeX}$ . Entire categories of  $\text{\LaTeX}$  packages work as-is when used with lwarp: definitions, file handling, utilities, internal data structures and calculations, specialized math-mode typesetting for various fields of science and engineering, and anything generating plain-text output. Blocks of PDF output may be automatically converted to SVG images while using the same font and spacing as the original print document, directly supporting Tikz and picture. Numerous packages are easily adapted for HTML versions, either by loading and patching the originals, or by creating nullified or emulated replacements, and all without resorting to external programming. As a result, several hundred packages have already been adapted (table 1), and an uncounted number more work as-is.

Packages have been selected according to several criteria: perceived importance, popularity lists, recent CTAN updates, CTAN topics, mention in other packages, support by other HTML conversion methods, and from sample documents taken from public archives<sup>1</sup>.

---

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

---

pdf $\text{\LaTeX}$ , x $\text{\LaTeX}$ , or lua $\text{\LaTeX}$  may be 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 PNG and JPG are used as-is.

---

<sup>1</sup>An amazing number of decades-old packages are still in use today.

SVG images may be used for math, and are also used for `picture`, `Tikz`, and similar environments. The svg format has better browser and e-book support than MathML (as of this writing), while still allowing for high-quality display and printing of images (again, subject to potentially bug-ridden<sup>2</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>3</sup> Custom  $\text{\LaTeX}$  macros may be used as-is in math expressions, since the math is evaluated entirely inside  $\text{\LaTeX}$ . An MD5 hash is used to combine multiple instances of the same inline math expression into a single image file, which then needs to be converted to svg only a single time.

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.<sup>4</sup> lwarp maintains  $\text{\LaTeX}$  control for cross-referencing and equation numbering, and attempts to force MATHJAX to tag equations accordingly.

---

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.

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 standard  $\text{\LaTeX}$  version. These `lwarp-packagename.sty` files contain code used to emulate or replace functions for HTML output.

---

<sup>2</sup>FIREFOX has had an on-again/off-again bug for quite some time regarding printing svgs at high resolution.

<sup>3</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!

<sup>4</sup>One svg image file per math expression, except that duplicate inline math expressions are combined into a single file according to the MD5 hash function of its contents. A common scientific paper can easily include several thousand files, and in one case the MD5 hash cut the number of files in half and the rendering time by 30%.

### 3.1 Supported packages and features

Table 1 lists some of the various  $\text{\LaTeX}$  features which may be used.

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

Category	Status and supported features.
Engines:	pdf $\text{\LaTeX}$ , X $\text{\LaTeX}$ , Lua $\text{\LaTeX}$
Classes:	book, report, article, scrbook, scrreprt, scrartcl, memoir.
Koma-script:	scrxextend, scrlayer, scrhack. Others as listed below.
Memoir:	memhfixc
Page layout:	geometry, fancyhdr, titleps, scrlayer-scrpage, typearea, vmargin, addlines, anysize, a4, a4wide, a5comb, textarea, zwpagelayout, ebook, preview, draftwatermark, watermark, everyshi, atbegshi, continue, fwlw, turnthepage, pagesel, blowup, pagegrid.
Sectioning:	Adds FileDepth for splitting the HTML output. Files may be numbered sequentially or named according to section name. Common short words and punctuation are removed from the filenames. titlesec, fncychap, sectsty, section, anonchap, quotchap, sectionbreak.
Table of contents, figures, tables:	Supported, with hyperlinks. tocbibind, titletoc, tocloft, tocbasic, and tocstyle, multitoc.
Title page:	\maketitle, titlepage, titling, authblk.
Front & back matter:	abstract, appendix.
Indexing:	texindy is used, with hyperlinks. idxlayout.
Glossary:	glossaries and xindy are used.
Bibliography:	cite, natbib, biblatex, chapterbib, backref, hypernat.
Cross-references:	hyperref, cleveref, varioref, fancyref, prettyref, titleref, url, breakurl, xurl, bookmark, hypdestopt.

## lwarp Supported Functions — continued

Category	Status
Languages:	babel. (polyglossia is untested.)
Margin notes:	marginfit, marginfix, scrlayer-notecolumn.
Footnotes:	Adds FootnoteDepth to print footnotes at section breaks. footnote, footmisc, marginnote, sidenote, pagenote, endnotes, endheads, footnpag, nccfoots.
Math:	Converted to SVG images with HTML <alt> tags containing the $\LaTeX$ source for the math expression. MATHJAX supported as an alternative. $\mathcal{AMS}$ environments are supported. User-defined macros are available during conversion, due to native $\LaTeX$ processing.
Theorems:	Native $\LaTeX$ theorems, theorem, amsthm, ntheorem.
Additional math:	Math fonts via SVG images, resizegather. As-is: delarray, bm, braket, etc.
Units and fractions:	siunitx, xfrac, nicefrac, units
Science and engineering:	listings, algorithmicx, bytefield, axodraw2, phfqit. Tested to work with SVG math: physics, slashed, heppennames, hepnice names, simpler-wick, linop, blochsphere, etc. ( <i>Math-mode packages generally work as-is with SVG math.</i> )
Liberal arts:	schemata
Floats:	Appear where declared. float, rotfloat, newfloat, caption and subcaption, subfig, subfigure, capt-of, placeins, trivfloat, floatrow, subfloat, keyfloat, wrapfig, cutwin, floatflt, flafter, fltrace, endfloat, hypcap, stfloats, fix2col, dblfloatfix, nonfloat, morefloats.
Tabular:	tabular environment, array, tabularx, tabulary, threeparttable, multirow, longtable, supertabular, xtab, ltxtable, booktabs, colortbl, bigdelim, diagbox.

## lwarp Supported Functions — continued

Category	Status
Graphics:	graphics and graphicx. <code>\includegraphics</code> 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. <code>\rotatebox</code> and <code>\scalebox</code> are supported as well as HTML can handle. rotating is emulated but all objects are unrotated. <code>picture</code> , <code>tikz</code> , and <code>xy</code> are converted to an SVG image. <code>grffile</code> , <code>epstopdf</code> , <code>overpic</code> , <code>figsize</code> .
xcolor:	Full package color names, any color models, and mixing. <code>\textcolor</code> , <code>\colorbox</code> , <code>\fcolorbox</code> . Enhanced for HTML compatibility.
Lists:	Standard $\text{\LaTeX}$ environments, <code>enumitem</code> , <code>enumerate</code> , <code>paralist</code> , <code>hang</code> .
Environments:	Standard $\text{\LaTeX}$ environments.
<code>minipage</code> , <code>\parbox</code> :	Some HTML5-imposed limitations. Nested minipages are supported. <code>pbox</code> .
Quotations:	<code>verse</code> , <code>csquotes</code> , <code>epigraph</code>
Verbatim:	<code>verbatim</code> , <code>moreverb</code> , <code>fancyvrb</code> (except for verbatim footnotes), <code>shortvrb</code> .
Frames:	<code>framed</code> , <code>fancybox</code> , <code>mdframed</code> , <code>boxedminipage2e</code> , <code>shadow</code> , <code>vertbars</code> .
Multi-columns:	<code>multicol</code> , <code>adjmulticol</code> , <code>vwcol</code> .
Margins:	<code>midpage</code> , <code>hanging</code> , <code>fullwidth</code> .
Line numbering:	<code>lineno</code> , <code>fnlineno</code> .
Acronyms:	<code>acro</code> , <code>acronym</code> .
Todo notes:	<code>todo</code> , <code>todonotes</code> , <code>easy-todo</code> , <code>fixmetodonotes</code> , <code>fixme</code> , <code>changebar</code> .



## lwarp Supported Functions — continued

Category	Status
Direct formatting:	<code>\emph</code> , <code>\textsuperscript</code> , <code>\textbf</code> , etc are supported. <code>\bfseries</code> , etc. are only supported in a few cases. <code>lettrine</code> , <code>ulem</code> , <code>soul</code> , <code>soulutf8</code> , <code>soulpos</code> , <code>cancel</code> , <code>resize</code> , <code>scalefont</code> , <code>textfit</code> , <code>realscripts</code> , <code>hyphenat</code> , <code>pdfrender</code> , <code>luacolor</code> .
Ordinals:	<code>nth</code> , <code>fntcount</code> , <code>engord</code> .
Text ligatures:	Ligatures for symbols are supported. Ligatures for <code>f</code> , <code>q</code> , <code>t</code> are intentionally turned off because many simpler browsers do not display them correctly. Modern full-featured browsers re-create these ligatures on-the-fly.
Horizontal space:	HTML output for <code>thin-unbreakable</code> , <code>unbreakable</code> , <code>\enskip</code> , <code>\quad</code> , <code>\qquad</code> , <code>\hspace</code> .
Rules:	<code>\rule</code> with <code>width</code> , <code>height</code> , <code>raise</code> , <code>text color</code> .
HTML reserved characters:	<code>\&amp;</code> , <code>\textless</code> , and <code>\textgreater</code> are converted to HTML entities.
Fonts:	Used as-is. Appear in SVG math expressions or embedded image environments.
Symbols:	Native $\TeX$ diacriticals, <code>textcomp</code> , <code>textgreek</code> .
Working as-is:	Various utility, calculation, file, and text-only packages, such as <code>calc</code> , <code>fileerr</code> , <code>somedefs</code> , <code>trace</code> , <code>xspace</code> . Also, any math-only packages, including specialized typesetting for various fields of science and engineering.

## 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, lwarp may be a good choice. For new projects with a large number of documents, it may be worth investigating the alternatives before decided which path to take.

### 4.1 Internet class

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

### 4.2 TeX4ht

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

Prog `htlatex` 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 most cases, lwarp provides a better HTML conversion, while supporting more 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` **LaTeXML**: <http://dlmf.nist.gov/LaTeXML/>

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

Prog `LaTeX2HTML` **LaTeX2HTML**: <http://www.latex2html.org/>  
and <http://ctan.org/pkg/latex2html>.

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

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

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

## 4.4 AsciiDoc and AsciiDoctor

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  $\text{\LaTeX}$  and HTML.

Prog AsciiDoc **AsciiDoctor:** <http://asciidoctor.org/> (More active.)  
 Prog AsciiDoctor **AsciiDoc:** <http://asciidoc.org/> (The original project.)

### 4.4.1 AsciiDoctor-LaTeX

The AsciiDoctor-LaTeX project is developing additional  $\text{\LaTeX}$ -related features.

**AsciiDoctor-LateX:**

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

## 4.5 Pandoc

Prog Pandoc A markup system which also reads and writes  $\text{\LaTeX}$  and HTML.

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

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

## 4.6 Word processors

Prog Word It should be noted that the popular word processors have advanced through the years in their abilities to represent math with a  $\text{\LaTeX}$ -ish input syntax, unicode math fonts, and high-quality output, and also generate HTML with varying success. See recent developments in MICROSOFT<sup>®</sup> Word<sup>®</sup> and LIBREOFFICE<sup>™</sup> Writer.

Prog LibreOffice  
 Prog OpenOffice

## 4.7 Commercial systems

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

Prog FrameMaker  
 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 lwarp, CSS is used, and much HTML output is adjusted through the usual  $\text{\LaTeX}$  optional macro parameters, but further customization may require patching  $\text{\LaTeX}$  code.

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

## 5 Installation

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

Detailed installation instructions follow.

Table 2: Required software programs

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

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

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

**The lwarp package:** This package.

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

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

**xindy:** The xindy program is used by lwarp to create indexes. On a MiK $\text{\TeX}$  system this may have to be acquired separately, but it is part of the regular installer as of mid 2015.

**latexmk:** Optionally used by lwarpmk to compile  $\text{\LaTeX}$  code. On a MiK $\text{\TeX}$  system, Perl may need to be installed first.

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

**POPPLER PDF utilities:**

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

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

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

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

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

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

```
Enter ⇒ brew install poppler
```

For WINDOWS, see:

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

**Perl:**

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

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

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

**MATHJAX:** Optionally used to display math. From: [mathjax.org](http://mathjax.org)

## 5.1 Installing the lwarp package

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

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

```
Enter ⇒ kpsewhich lwarp.sty
```

If a path to `lwarp.sty` is shown, then lwarp is already installed and you may skip to the next section.

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

```
Enter ⇒ tlmgr install lwarp
```

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

1. To install lwarp the first time, use the MikTeX Package Manager (Admin).
2. To update lwarp, use MikTeX Update (Admin).
3. Either way, also update the package `miktex-misc`, which will install and update the `lwarpmk` executable.

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

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

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

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

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

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

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

This should be something like:

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

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

```
Enter ⇒ mktexlsr
```

— or —

```
Enter ⇒ texhash
```

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

**CTAN .dtx and .ins files:** Another form of T<sub>E</sub>X package is the .dtx and .ins source files. These files are used to create the documentation and .sty files.

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

```
Enter ⇒ pdflatex lwarp.dtx
```

 (several times)
6. Create the .sty files:
 

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

```
<texlocal>/tex/latex/local/lwarp
```
8. Copy the documentation lwarp.pdf to a source directory in the local tree, such as:
 

```
<texlocal>/doc/local/lwarp
```
9. Renew the cache:
 

```
Enter ⇒ 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 lwarpmk.
11. Once the local version of lwarpmk.lua is installed, it may be made available system-wide as per section 5.2.

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

Just testing!

## 5.2 Installing the lwarpmk utility

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

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



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

Otherwise, continue with the following:

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

```
Enter ⇒ kpsewhich lwarpmk.lua
```

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

4. Create `lwarpmk`:

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

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

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

This will be something like:

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

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

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

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

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

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

- (c) Make the link executable:

```
Enter ⇒ chmod 0755 lwarpmk
```

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

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

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

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

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

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

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

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

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

### 5.2.1 Using a local copy of lwarpmk

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

1. When compiling the tutorial in section 6, use the lwarpmk option for the lwarp package:

```
\usepackage[lwarpmk]{lwarp}
```

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

**Unix:**

- (a) Make lwarpmk.lua executable:

```
Enter ⇒ chmod 0755 lwarpmk.lua
```

- (b) Compile documents with

```
Enter ⇒ ./lwarpmk.lua html
```

```
Enter ⇒ ./lwarpmk.lua print
```

etc.

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

**WINDOWS:**

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

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

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

etc.

Or:

```
Enter ⇒ lwarpmk html
```

```
Enter ⇒ lwarpmk print
```

etc.

## 5.3 Installing additional utilities

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

Enter the following in a command line. If each programs' version is displayed, then that utility is already installed. See table 2 on page 62.

```
Enter ⇒ luatex --version
```

```
Enter ⇒ xindy --version
```

```
Enter ⇒ latexmk --version
```

```

Enter ⇒ perl --version
Enter ⇒ pdfcrop --version
Enter ⇒ pdftotext -v
Enter ⇒ pdfseparate --version
Enter ⇒ pdftocairo -v

```

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

The T<sub>E</sub>X utilities xindy, latexmk, and pdfcrop may be installed in TexLive with tlmgr, installed by MikTeX, provided by your operating system's package manager, or downloaded from the CTAN archive:

```

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

```

```

Prog  pdftotext
Prog  pdfseparate
Prog  pdftocairo

```

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

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

### To install the POPPLER utilities to a MACOS machine:

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

```
Enter ⇒ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```
2. Install the POPPLER utilities:  

```
Enter ⇒ brew install poppler
```

### To install the POPPLER utilities to a WINDOWS machine:

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

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

Be sure to include \bin.
9. Click "Ok" when done.

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

1. Download and install a version of Perl, such as STRAWBERRY PERL, to a directory without a space in its name, such as C:\Strawberry.
2. Edit the Path as seen above for the POPPLER utilities.
3. Enter the bin directory of the Perl utility, such as:  
C:\Strawberry\perl\bin  
Be sure to include \bin.
4. Click "Ok" when done.

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

## 6 Tutorial

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

### 6.1 Starting a new project

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

File `tutorial.tex`

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

#### Copy from the documentation PDF:

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

#### Copy from the lwarp documentation directory:

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

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

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

File `lwarp_tutorial.txt`

⚠ Note: `.txt` suffix!

⚠ Bad formatting!

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

3. Compile the project:

```
Enter ⇒ pdflatex tutorial.tex
```

(several times)

(xelatex or lualatex may be used as well.)

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

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

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

Figure 1: tutorial.tex listing

Note: There are two pages!

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

\documentclass{book}

\usepackage{iftex}

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

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

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

% --- LOAD PDFLATEX MATH FONTS HERE ---

% --- 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
\HTMLTitle{Webpage Title} % Overrides \title for the web page.
\HTMLAuthor{Some Author} % Sets the HTML meta author tag.
```

```

\HTMLLanguage{en-US}           % Sets the HTML meta language.
\HTMLDescription{A description.}% Sets the HTML meta description.
\HTMLFirstPageTop{Name and \fbox{HOMEPAGE LOGO}}
\HTMLPageTop{\fbox{LOGO}}
\HTMLPageBottom{Contact Information and Copyright}
\CSSFilename{lwarp_sagebrush.css}

\begin{document}

\maketitle                     % Or titlepage/titlingpage environment.

% An article abstract would go here.

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

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

\section{Some math}

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


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

\end{document}

```

Table 3: Files created along with the print version

---

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

---

When the lwarpmk option is given to the lwarp package:

**lwarpmk.lua:** A local copy of the lwarpmk utility.  
 On Unix-related operating systems this file must be made executable:  

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

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



## 6.2 Compiling the print version with lwarpmk

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

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

1. Re-compile the print version:

Enter ⇒ 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>5</sup>

5. Process the index.<sup>67</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>5</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>6</sup>A lwarpmk printglossary command is also available to process a glossary produced with the glossaries package. See section 8.5.8.

<sup>7</sup>Also see section 8.5.9 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>8</sup>, `Some-math.html`, and the document's index in `_Index.html`.<sup>9</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>1011</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>8</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>9</sup>`index.html` is commonly used as a homepage, so the document index is in `_Index.html`.

<sup>10</sup>A `lwarpmk htmlglossary` command is also available to process a glossary produced with the `glossaries` package. See section 8.5.8.

<sup>11</sup>Also see section 8.5.9 for index options.

## 6.4 Generating the SVG images

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


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

1. Create the svg images:


Enter  $\Rightarrow$  `lwarpmk limages`


Enter  $\Rightarrow$  `lwarpmk html`

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

 **math in \section** When using svg math in sectioning commands, use `\(` and `\)` to delimit math instead of `$`:

```
\section{Section name with math \((1+2=3)\)}
```

 **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  $\LaTeX$  would, but has the disadvantage of requiring an individual file for each math expression. For inline math, lwarp uses an MD5 hash on its  $\LaTeX$  source to combine multiple instances of identical inline expressions into a single image file, but display math and other environments such as `picture` and `Tikz` require one image file each. For a document with a large amount of math, see section 6.5 to use `MATHJAX` instead.

 **Lots of files!**

## 6.5 Using MATHJAX for math

**math with MATHJAX** Math may also be represented using the MATHJAX JAVASCRIPT project.

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

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

2. Recompile

```
Enter ⇒ lwarpmk html
```

3. Reload the math page.



### MATHJAX requirements

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

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

## 6.6 Changing the CSS style

For a formal css style, add to the preamble:

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

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

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

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

## 6.7 Customizing the HTML output

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

See section 7.3 for more information.

## 6.8 Using latexmk

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

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

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

2. Recompile the printed version of the document.

```
Enter ⇒ lwarpmk print
```

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

3. Recompile the document.

```
Enter ⇒ lwarpmk print
```

and/or

```
Enter ⇒ lwarpmk html
```

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

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

```
Enter ⇒ lwarpmk print1
```

and/or

```
Enter ⇒ lwarpmk html1
```

## 6.9 Using XeLaTeX or LuaLaTeX

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

1. Remove the auxiliary files for the project:

```
Enter ⇒ lwarpmk cleanall
```

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

```
Enter ⇒ xelatex tutorial.tex
```

-or-

```
Enter ⇒ lualatex tutorial.tex
```

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

3. To recompile the document:

```
Enter ⇒ lwarpmk print
```

-and-

```
Enter ⇒ lwarpmk html
```

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

## 6.10 Using a glossary

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

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

---

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

---

To process the glossary for the print version:

```
Enter ⇒  lwarpmk printglossary
```

To process the glossary for the HTML version:

```
Enter ⇒  lwarpmk htmlglossary
```

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



## 6.11 Cleaning auxiliary files

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

```
Enter ⇒ lwarpmk clean
```

## 6.12 Cleaning auxiliary and output files

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

```
Enter ⇒ lwarpmk cleanall
```

## 6.13 Processing multiple projects in the same directory

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

To create each project:

```
Enter ⇒ pdflatex project_a
```

```
Enter ⇒ pdflatex project_b
```

Each project is given its own configuration file:

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

To compile each project with `lwarkmk`:

```
Enter ⇒ lwarpmk print project_a
```

```
Enter ⇒ lwarpmk html project_b
```

## 6.14 Using the make utility

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

```
lwarpmk pdftohtml [project]
```

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

## 6.15 Converting an existing document

To convert an existing document for use with lwarp:

1. Arrange the document in the following order:
  - (a) Declare the `\documentclass`.
  - (b) Load text fonts.
  - (c) Load `inputenc`, `fontenc`, and/or `fontspec`.
  - (d) Load lwarp.
  - (e) Load remaining packages.
2. Also modify the document:
  - (a) Change `\includegraphics` PDF filenames from `filename.pdf` to `filename` without a suffix. Other image formats may be left unmodified, or may be loaded without a file suffix.
  - (b) Possible changes to tabular environments: `* columns`, `multirow`, `longtable`, `supertabular`, `xtab`, `bigdelim`. See section 8.9.
  - (c) Possible option clashes with `memoir`. See section 8.12.
  - (d) Other changes as per section 8.
3. Create an SVG version of any PDF image.
4. Manually compile print version with `pdflatex`, `lualatex`, or `xelatex`.
5. `lwarpmk print` to finish the print version.
6. `lwarpmk html` to create the HTML version.
7. `lwarpmk limages` to create the svg images of any `svg math`, `lateximage`, `Tikz`, etc.

## 7 Additional details

### 7.1 Font and UTF-8 support

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

**pdflatex, T1, UTF8** While using pdflatex, fontenc is automatically loaded with T1 encoding, and inputenc is automatically loaded with UTF8 encoding, each of which is required for the conversion process. fontenc may be loaded with an additional encoding after lwarp.

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



```
usepackage{lmodern}
```

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

```
\usepackage{dejavu}
```

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



X<sub>Y</sub>TeX and LuaTeX users must use the fontspec package. Do NOT use fontenc!

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

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

- For X<sub>Y</sub>TeX or LuaTeX:

- fontspec and font choices

lwarp sets the following to turn off TeX ligatures during the generation of HTML tags, and turn off common ligatures in regular text, since older browsers may not display them correctly and newer browsers can automatically re-create them.

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

- For pdflatex:

- lmodern or other font-related packages

Pkg fontspec

ligatures

Pkg lmodern

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

3. `\usepackage{lwarp}` (section 7.2) goes after any of the above, followed by:
4. `\usepackage{newtxmath}` or other math-related font packages. Many of these load `amsmath`, which must be loaded after `lwarp`, so they must also be loaded after `lwarp`.
5. ... the rest of the preamble and the main document.

### 7.1.1 Indexes and UTF-8

`lwarp` uses the `xindy` program to process indexes.

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

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

Table 4: Package options

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

## 7.2 lwarp package loading and options

lwarp supports book, report, and article classes, as well as the equivalent Koma-script classes and memoir.

Pkg	<code>lwarp</code>	Load the lwarp package immediately after the font and UTF-8 setup commands.
Opt	<code>warpprint</code>	Usually controlled by <code>lwarpmk</code> , and not set in the document. Select the <code>warpprint</code> option to generate print output (default), or the <code>warpHTML</code> option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual <code>pdflatex</code> , etc. When lwarp is loaded in print mode, it creates <code>&lt;project&gt;_html.tex</code> , which sets the <code>warpHTML</code> option before calling the user's source code <code>&lt;project&gt;.tex</code> . In this way, <code>&lt;project&gt;.tex</code> can <code>\usepackage{lwarp}</code> without any options to create a printed version, while <code>&lt;project&gt;_html.tex</code> will create an HTML version.
Opt	<code>warpHTML</code>	
Opt	<code>mathsvg</code>	For math display, select <code>mathsvg</code> (default), or <code>mathjax</code> . For more information about the math options, see section 8.6.
Opt	<code>mathjax</code>	


---


Opt	OSWindows	See section 7.5 if using WINDOWS.
Opt	BaseJobname	Not intended for the user. Used internally by lwarp when creating the *_html.tex file used to compile the HTML version. See section 23.
Opt	HomeHTMLFilename	See section 7.3.
Opt	HTMLFilename	See section 7.3.
Opt	IndexLanguage	If using an index or glossary, see section 23.
	Opt latexmk	Has lwarpmk use latexmk to recompile the document several times if necessary. Otherwise, lwarpmk attempts to determine for itself whether to recompile. See section 7.3.
	Opt lwarpmk	If you wish to have lwarp generate a local copy of lwarpmk.lua for archival or local-installation purposes, compile the print version with the lwarpmk option set. See section 23.
Opt	xdyFilename	The default xindy filename is lwarp.xdy. If you wish to use a custom .xdy file for index generation, see section 23.

Table 5: HTML settings

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

### 7.3 Customizing the HTML output

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

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

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

```
Enter ⇒ lwarpmk clearall
```

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

```
Enter ⇒ lwarpmk print
```

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

```
Enter ⇒ lwarpmk html
```

**Options for the lwarp package:**

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

Opt	HomeHTMLFilename Default: <code>\BaseJobname</code>	<b>HomeHTMLFilename:</b> Filename of the homepage, without the “.html” suffix. Defaults to the <code>\BaseJobname</code> . A common setting is:  <code>HomeHTMLFilename=index</code>  causing the homepage to be the file <code>index.html</code> . Underscores are allowed in <code>HomeHTMLFilename</code> and <code>HTMLFilename</code> options, but may need to be escaped elsewhere, such as when appearing in a list:  <code>\item [\href{file\_name.pdf}{text}] \</code>  See section 7.3.1 for examples of naming and numbering HTML files.
Opt	HTMLFilename Default: <code>&lt;empty&gt;</code>	<b>HTMLFilename:</b> A filename prefix for the rest of the HTML web pages. Useful for numbered web pages with a common prefix. May be empty. See section 7.3.1 for examples of naming and numbering HTML files.
Opt	latexmk Default: <code>false</code>	<b>latexmk:</b> Controls whether lwarp uses latexmk to compile the document. This setting is written to lwarpmk’s configuration files.
Opt	mathsvg Default: <code>true</code>	<b>mathsvg:</b> Selects SVG display for math output. (The default.)
Opt	mathjax Default: <code>false</code>	<b>mathjax:</b> Selects MATHJAX for math output.

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

Ctrl	tocdepth	<b>tocdepth:</b> Sectioning depth of the table of contents. See section 14 for a list of L <sup>A</sup> T <sub>E</sub> X stack depths.
Ctrl	SideTOCDepth Default: <code>1</code>  <b>sideroc</b>	<b>SideTOCDepth:</b> Sectioning depth of the sideroc. Defaults to 1, causing the sideroc to show sections but not subsections.  Each subpage of the website has its own small table of contents on the side (the “sideroc”). Its depth is set by <code>SideTOCDepth</code> . This sideroc is only shown if the web page is wide enough. When using a narrow web browser window, “responsive web design” is used to show the sideroc at the top of the page and a link back to “Home” at the bottom.

It is recommended to set:

`SideTOCDepth=FileDepth`

or

`SideTOCDepth=FileDepth+1`

 **inaccessible pages**

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



Ctrl **FileDepth**  
Default: -5

**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**  
Default: true

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

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

⚠ Inaccessible pages!

⚠ Lost in an old page!

When making changes to the file structure, it is possible to end up with the web browser pointing to an old file which is no longer in use. When this occurs, changes to the web site will not appear in the browser, even if reloading the page, because that page is no longer in use. It is best to return to the home page, clean the files (`lwarpmk cleanall`), change FileDepth and/or CombineHigherDepths, then finally recompile and renavigate to the desired page using the new file structure.

Bool **FileSectionNames**  
Default: true

**FileSectionNames:** If true, web page filenames are derived from a sanitized version of the section names. If false, web pages are numbered. Either way, the HTMLFilename option is used as a prefix. See section 7.3.1 for examples of naming and numbering HTML files. The user must ensure that filenames are unique after begin sanitized. For example, math in the section name is removed before creating the filename, so the rest of the filename must be sufficiently unique to avoid name collisions.

⚠ Unique filename!

Ctrl **FootnoteDepth**  
Default: 5

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

**\abstractname**  
Default: Abstract

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

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

`\CSSFilename`  
Default: `lwarp.css`

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

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

`\HTMLLanguage`  
Default: `en-US`

**\HTMLLanguage:** `{\langauge}` The HTML file's `html lang` meta tag. Defaults to `en-US`.

`\HTMLTitle`  
Default: `\thetitle`

**\HTMLTitle:** `{\title}` Overrides `\title` for the HTML header's meta title. Defaults to `\thetitle`, which is set by `\title`, or empty otherwise. Unlike the author, `\thetitle` is set by `\title` even if not using the titling package.

`\HTMLAuthor`  
Default: `\theauthor`

**\HTMLAuthor:** `{\author}` The HTML header's meta author. Defaults to `\theauthor`, which is set by `\author` if using the titling package, but is empty otherwise. There are several ways to represent the author and affiliations, especially if using the `authblk` package, most of which do not result in a sensible `\theauthor`, so `\HTMLAuthor` is useful to create a list of authors without their affiliations.

`\HTMLDescription`  
Default: `<empty>`

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

`\HTMLFirstPageTop`  
Default: `<empty>`

**\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`  
Default: `<empty>`

**\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`  
Default: `<empty>`

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

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

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

### 7.3.1 Example HTML file naming

Examples of ways to name or number HTML files:

**Numbered HTML nodes:**

Example: Homepage `index.html`, and `node-1`, `node-2`.<sup>12</sup>

---

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

---

<sup>12</sup>See `\SetHTMLFileNumber` to number in groups by chapter, for example.

**Named HTML sections, no prefix:**

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

---

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

---

**Named HTML sections, with prefix:**

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

---

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

---

**7.4 Customizing the CSS**

`\CSSFilename`  
Default: `lwarp.css`

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

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

The styles provided by lwarp include:

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

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

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

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

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



#### Rename it!

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

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

File `lwarp.css`  
File `project.css`  
File `sample_project.css`

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

---

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

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

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

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

---

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

## 7.5 Selecting the operating system

Prog	Unix	lwarp tries to detect which operating system is being used. UNIX / MAC OS / LINUX is the default (collectively referred to as "UNIX" in the configuration files), and MS-WINDOWS is supported as well.
Prog	Mac OS	
Prog	Linux	
Prog	MS-Windows	If MS-WINDOWS is not correctly detected, use the <code>lwarp</code> option <code>OSWindows</code> .
Prog	Windows	
Opt	OSWindows	

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

## 7.6 Selecting actions for print or HTML output

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

For most of built-in  $\text{\TeX}$  and many additional packages there is user-level source code support or emulation, so no special handling will be required. For those cases which lwarp does not handle by itself, the following environments and macros may be used to isolate sections of code for print-only or HTML-only.

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

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

---

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

---

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

---

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

---

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

---

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

---

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

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

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

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

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

## 7.7 Commands to be placed into the warpprint environment

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

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

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

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

## 7.8 Title page

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

---

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

---

The title is used in the meta tags in the HTML files, unless overridden by `\HTMLTitle`, and the rest are used in `\maketitle`. To use a `\subtitle` or `\published` field, see section 58.8.

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

Env `titlepage` The `titlepage` environment may be used to hold a custom title page. The `titlepage`

will be set in a `<div>` class `titlepage`, and `\printtitle`, etc. may be used inside this environment.

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

`\title` `{\langle title \rangle}`



Avoid newlines in the `\title`; these will interfere with the file break and CSS detection. Use a `\subtitle` command instead (section 58.8). The title will appear in the document `\maketitle` as a heading `<h1>`. The HTML meta title tag will also have this title, unless `\HTMLTitle` is used to set the meta title to something else instead.

`\author` `{\langle author \rangle}`



In `\author`, use `\protect` before formatting commands such as `\textsc`. In HTML, the author will appear in a `<div>` of class `author` in the `\maketitle`. If the titling package is used, the author will also appear in a HTML meta tag, but `\HTMLAuthor` may be necessary to create a plain list of names if `\author` had affiliations added. `\affiliation` is a new addition to `lwarp`.

`\date` `{\langle date \rangle}`

`\date` works as expected. In HTML, this will appear in a `<div>` class `datedate`.

`\thanks` `{\langle text \rangle}`

`\thanks` are allowed in the titlepage fields, and will be rendered as HTML notes at the bottom of the title page.

## 7.9 HTML page meta descriptions

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

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

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

**disabling** To disable the generation of HTML description meta tags, use:



`\HTMLDescription{}`

## 7.10 HTML page meta title

`\HTMLTitle {<title>}` Sets the contents of the web page `<meta name="title">` element. Defaults to `\HTMLtitle{\thetitle}`. May be set empty to cancel the meta title tag.

## 7.11 HTML page meta author

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

`\author` may be used to create a list of authors and their affiliations, in several formats if using `authblk`, and these may not successfully parse properly into a sensible list for `\theauthor`. `\HTMLAuthor` may be used to set the meta tag to a simple list of names.

## 7.12 Modifying xindy index processing

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

To customize index processing:

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

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

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

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

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


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

## 8 Special cases and limitations

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

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

### 8.1 Sectioning


 **math in `\section`** When using SVG math in sectioning commands, use `\(` and `\)` to delimit math instead of `$`:

```
\section{Section name with math \((1+2=3)\)}
```

### 8.2 Formatting

#### 8.2.1 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.

#### 8.2.2 Horizontal space

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

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

`\kern` `\kern` and `\hspace` are entered into the HTML PDF output as-is, then interpreted by `pdftotext`, and thus usually appear as a single space.

#### 8.2.3 Text alignment


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

### 8.2.4 Accents

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

### 8.2.5 Textcomp

`Pkg 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.

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

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

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

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

### 8.2.8 Filenames and URLs in lists or footnotes

`filename underscore` Escape underscores in the filenames:


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

### 8.2.9 relsize package

`Pkg relsize` 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.

## 8.3 Boxes and minipages


### 8.3.1 Marginpars

- `\marginpar` [*left*] {*right*} `\marginpar` may contains paragraphs, but in order to remain inline with the surrounding text `lwarp` nullifies block-related macros inside the `\marginpar`. Paragraph breaks are converted to `<br />` tags.
- `\marginparBlock` [*left*] {*right*} To include block-related macros, use `\marginparBlock`, which takes the same arguments but creates a `<div>` instead of a `<span>`. A line break will occur in the text where the `\marginBlock` occurs.


### 8.3.2 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.

### 8.3.3 Minipages

-  **inline** A line of text with an inline minipage or parbox will have the minipage or parbox placed onto its own line, because a paragraph is a block element and cannot be made `inline-block`.
- placement** Minipages and parboxes will be placed side-by-side in HTML unless you place a `\newline` between them.
- side-by-side** Side-by-side minipages may be separated by `\quad`, `\qquad`, `\enskip`, `\hspace`, `\hfill`, or a `\rule`. When inside a `center` environment, the result is similar in print and HTML. Paragraph tags are suppressed between side-by-side minipages and these spacing commands, but not at the start or end of the paragraph.
- in a span** There is limited support for minipages inside an HTML `<span>`. An HTML `<div>` cannot appear inside a `<span>`. While in a `<span>`, minipages, and parboxes, and any enclosed lists have limited HTML tags, resulting in an “inline” format, without markup except for HTML breaks. Use `\newline` or `\par` for an HTML break.
- size** When using `\linewidth`, `\textwidth`, and `\textheight`, widths and heights are scaled proportionally to a 6×9 inch text area.
- no-width minipages** A minipage of width exactly `\linewidth` is automatically given no HTML width.

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

 **text alignment** Nested minipages adopt their parent's text alignment in HTML, whereas in regular  $\text{\LaTeX}$  PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

### 8.3.4 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.

### 8.3.5 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`  
`\fboxBlock`  
 Env `fminipage`

For a framed object, options include:

**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.

For inline text:

**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.

For inline minipage and lists:

**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  $\text{\LaTeX}$  code, the result in HTML is problematic since a table is a `<div>` not a `<span>`, so use

`\fboxBlock` around a `tabular`, or else place the `tabular` inside a `minipage`, or use `fminipage`, described next. Also see below regarding the “Misplaced alignment tab character &” error.

For display `tabular`,  
minipages, and lists:

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

colored boxes and frames:

**To create colored frames and boxes:** See section 331 for `xcolor`’s `\colorbox` and `\fcolorbox`, and `lwarp`’s additional `\colorboxBlock` and `\fcolorboxBlock`.

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

**To frame equations:** See section 156 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}
```

---

### 8.3.6 fancybox package

Pkg `fancybox`  
framed equation example

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

```

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

```

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

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

```

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

```

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

```

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


```

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

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

### 8.3.7 mdframed package

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

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

**font** For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```


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


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

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


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

## 8.4 Cross-references

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

 **underscores** `\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.

### 8.4.1 Page references

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

Ex:



`\ref{sec:name}` on page `\pageref{sec:name}`  
in HTML becomes:

“Sec. 1.23 on page (see sec. 1.23)”.

`\pagerefPageFor` may be redefined to “page for”, empty, etc. See section 65.4.

### 8.4.2 cleveref and varioref packages

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

⚠ **cleveref page numbers**

Ex:

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

See `\cpagerefFor` at section 78 to redefine the message which is printed for page number references.

### 8.4.3 Hyperlinks, hyperref, and url

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

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

⚠ **backref** When generating HTML, lwarp’s emulation of `hyperref` does not automatically load `backref`, so `backref` must be loaded explicitly.

### 8.4.4 Footnotes and page notes

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

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

## 8.5 Front and back matter

### 8.5.1 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 `lwarp`.

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

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

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

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

⚠ inaccessible HTML page

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

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

### 8.5.2 abstract package

`Pkg abstract` 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.

⚠ missing TOC

### 8.5.3 titling and authblk

Pkg titling

Pkg authblk

package support

⚠ load order

\published and \subtitle

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

If using the `titling` package, additional titlepage fields for `\published` and `\subtitle` may be added by using `\AddSubtitlePublished` in the preamble. See section 58.8.

### 8.5.4 tocloft package

Opt tocloft titles

Pkg tocloft

⚠ tocloft & other packages

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

### 8.5.5 appendix package

Pkg appendix

⚠ incorrect TOC link

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

### 8.5.6 pagenote package

Pkg pagenote

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

### 8.5.7 endnotes package

Pkg endnotes

table of contents

To place the endnotes in the TOC, use:

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

HTML page

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

```
\ForceHTMLPage
\theendnotes
```

### 8.5.8 glossaries package

Pkg glossaries

`xindy` is required for `glossaries`.

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

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

#### placement and toc options

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

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

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

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

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

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


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

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

Opt	<code>lwarpmk</code>	<code>printglossary</code>	The <code>lwarpmk</code> has the commands <code>lwarpmk printglossary</code> and <code>lwarpmk htmlglossary</code> to process the glossaries created by <code>glossaries</code> using <code>xindy</code> .
Opt	<code>lwarpmk</code>	<code>htmlglossary</code>	

### 8.5.9 Index and the `tocbibind` package

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

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

#### placement and toc options

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

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

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

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

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

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

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

Pkg tocibind

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

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

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

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

Opt tocibind numindex  
numbered index section

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

See section 68 for lwarp's core index and glossary code, and section 313 for tocibind.

## 8.6 Math

### 8.6.1 Rendering tradeoffs

Math rendering

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

- SVG files** Rendering math as images creates a new svg file for each expression, except that an MD5 hash is used to combine identical duplicates of the same inline math expression into a single file, which must be converted to svg only once. Display math is still handled as individual files, since it may contain labels or references which are likely to change.
- SVG inline** The svg images are currently stored separately, but they could be encoded in-line directly into the HTML document. 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  $\text{\LaTeX}$ -to-HTML 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.


### 8.6.2 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:  

$$\text{\textbackslash renewcommand{\LateximageFontSizeName}\{large\}}$$

- SVG math copy/paste** For svg math, text copy/paste from the HTML `<alt>` tags lists the equation number or tag for single equations, along with the  $\text{\LaTeX}$  code for the math expression. For  $\mathcal{AMS}$  environments with multiple numbers in the same environment, only the first and last is copy/pasted, as a range. No tags are listed inside a starred  $\mathcal{AMS}$  environment, although the `\tag` macro will still appear inside the  $\text{\LaTeX}$  math expression.

-  **SVG math in  $\text{\TeX}$  boxes** SVG math does not work inside  $\text{\TeX}$  boxes, since a `\newpage` is required before and after each image.

<sup>13</sup>See section 332 regarding fonts and fractions.

### 8.6.3 MATHJAX option

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

Prog MathJax

When MATHJAX is enabled, math is rendered twice:

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

**MATHJAX limitations** Limitations when using MATHJAX include:

Prog MathJax

**chapter numbers**

- In document classes which have chapters, `\tagged` equations have the chapter number prepended in HTML output, unlike  $\text{\LaTeX}$ . `\tag*` equations (correctly) do not. This may be improved with future versions of the MATHJAX support script.

<https://groups.google.com/forum/#!topic/mathjax-users/jUteWUcE2bY>

**subequations**

- MATHJAX itself does not support subequations. This may be improved by parsing the  $\text{\LaTeX}$  math expression to manually insert tags, but this has not yet been done.

**footnotes in math**

- Footnotes inside equations are not yet supported while using MATHJAX.

**lateximage**

- Math appearing inside a `lateximage`, and therefore also inside a `Tikz` or `picture` environment, is rendered as SVG math even if MATHJAX is used in the rest of the document.

**siunitx**

- Usage of `siunitx` inside a math equation is supported via a third-party MATHJAX extension. While inside a math expression, do not use `\SI` or `\si` inside `\text`, where it will be rendered as normal text.

<https://github.com/burnpanck/MathJax-siunitx>

Also see section 8.6.5.

**tabbing**

- A tabbing environment is emulated using an HTML `<pre>`. While MATHJAX is enabled inside tabbing, the browser may not correctly render the horizontal alignment of the math and text following after on the same line.

⚠ **siunitx inside an equation**

### ⚠ other macros and packages

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

### custom MathJax macros

- MATHJAX does not automatically support custom  $\TeX$  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}
```

## 8.6.4 ntheorem package

Pkg `ntheorem`

This conversion is not total. Font control is via CSS, and the custom  $\TeX$  font settings are ignored.

### ⚠ Font control

### ⚠ 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.

## 8.6.5 siunitx package

Pkg `siunitx`

Due to `pdftolatex` limitations, fraction output is replaced by symbol output for `per-mode` and `quotient-mode`.

### fractions

### ⚠ 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.



### 8.6.6 units and nicefrac packages

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

### 8.6.7 newtxmath package

Pkg newtxmath The proper load order is:  
 ⚠ loading sequence ...  
                   `\usepackage{lwarp}`  
                   ...  
                   `\usepackage{amsthm}`  
                   `\usepackage{newtxmath}`  
                   ...

## 8.7 Graphics

Pkg graphics For `\includegraphics` with .pdf files, the user should provide a .pdf image file,  
 Pkg graphicx and also a .svg, .png, or .jpg version of the same image. **These should be referred**  
 ⚠ .pdf image files **to without a file extension:**  
 ⚠ no file extension `\includegraphics{filename} % print:.pdf, HTML:.svg or other`

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

Prog pdftocairo To convert a PDF image to SVG, use the utility `pdftocairo`:

Enter ⇒ `pdftocairo -svg filename.pdf`


If a .pdf file is 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`

Pkg epstopdf For .eps files, use `epstopdf` to provide a PDF version, and also provide a SVG version as well.

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`.


 **viewports** A single optional parameter is interpreted as the newer graphicx syntax. Note that viewports are not supported by warp; the entire image will be shown.

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


**options** `\includegraphics` accepts width and height, origin, rotate and scale, plus a new class key.

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

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


 **browser support** `\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The css3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike  $\TeX$ , so expect some ugly results for scaling and rotating.

### 8.7.1 grffile package

 **matching PDF and SVG** 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.

### 8.7.2 color package

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

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

### 8.7.3 xcolor package

Pkg xcolor `\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** `\rowcolors` is supported, except that the optional argument is ignored so far.

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


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

### 8.7.4 epstopdf package

**Pkg** `epstopdf` When using `epstopdf` to convert images to PDF, use the `pdftocairo` utility to also provide an SVG version as well. In the document, refer to the image filename without a suffix. The PDF version will be used in print output, and the SVG version will be used for HTML.

### 8.7.5 overpic package

**Pkg** `overpic` 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.

 **scaling**

## 8.8 Tabbing

The `tabbing` environment works, except that `svg math` and `lateximages` do not yet work inside the environment.

## 8.9 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 `xtable`.

**Defining environments:**

⚠ misplaced alignment  
alignment tab character &

⚠ floatrow

⚠ tabular inside another  
environment

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

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

This includes before and after defining any macro which used \ttabbox from floatrow.

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

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

**Cell contents:**

⚠ paragraphs

⚠ \multirow

vposn

- 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 & ...
```

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

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

⚠ `\multicolumn & \multirow`

⚠ skipped cells

⚠ empty cells

`vposn`

⚠ macro in a table  
custom macros

- See section 232.2 for `\multicolumnrow`.  
lwarp does not support directly combining `\multicolumn` and `\multirow`. Use `\multicolumnrow` instead. To create a 2 column, 3 row cell:

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

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

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

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

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

`@` and `!`

`\multirow`

⚠ `\newcolumntype`

- \* in a column specification is not used (so far). Repeat the column type the correct number of times.
- Only one each of `@` and `!` is used at each column, and they are used in that order.
- In `\multirow` cells, the print version may have extra instances of `<`, `>`, `@`, and `!` cells on the second and later rows in the `\multirow` which do not appear in the HTML version.
- `\newcolumntype` is ignored; unknown column types are set to 1.

### Rules:

vertical rules

width and trim

full-width rules

- Vertical rules next to either side of an `@` or `!` column are displayed on both sides of the column.
- Width options are honored. Trim options are converted to rounded top corners. Trim corners are not rounded with `@` or `!` columns, and full-width rules ignore trim.
- `\toprule`, `\midrule`, `\bottomrule`, and `\hline` ignore trim. When given an optional width, each cell is styled to create the custom border. Without an optional width, the entire row is given a class to assign the standard border.

**combined rules**

⚠ `\warpprintonly`  
 misplaced `\noalign`

- 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.

- For `\toprule` and `\bottomrule`, when combined with a `warpprint` or `warppHTML` environment, if a “misplaced `\noalign`” error occurs, change

```
This & That \endhead
```

to

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

and likewise with the other `\end` headings. Keep the `\endfirsthead` row unchanged, as it is still relevant to HTML output.

**colortbl:**

⚠ **row/cell color**

Only use `\rowcolor` and `\cellcolor` at the start of a row, in that order.

`colortbl` ignores the overhang arguments.

**Other:****longtable headings**

⚠ **S columns**

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

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

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

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

**8.9.1 longtable package**

Pkg `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`.

⚠ **lateximage** `longtable` is not supported inside a `lateximage`.

### 8.9.2 supertabular and xtab packages

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

Pkg `xtab` `\StartDefiningTabulars`  
 ⚠ misplaced alignment `\tablefirsthead`  
 alignment tab character & `...`  
`\EndDefiningTabulars`

See section 8.9.

⚠ **lateximage** `supertabular` and `xtab` are not supported inside a `lateximage`.

### 8.9.3 bigdelim package

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

---

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

---

```

      a   b
left { c   d
      e   f
      g   h

```

---

## 8.10 Floats

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

Pkg float  
Pkg trivfloat  
Pkg algorithmicx

If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 319.1.

### ⚠ package conflicts

### 8.10.2 caption and subcaption packages

Pkg caption  
Pkg subcaption

To pass options to `caption`, select the options before loading `lwarp`:

### ⚠ options

```
\documentclass{article}
...
\PassOptionsToPackage{options_list}{caption}
...
\usepackage{lwarp}
...
\usepackage{caption}
```

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

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

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

### 8.10.3 subfig package

Pkg subfig

### ⚠ 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.

### 8.10.4 floatrow package

Pkg floatrow

Use `\StartDefiningTabulars` and `\EndDefiningTabulars` before and after defining macros using `\ttabbox` with a tabular inside. See section 8.9.

### ⚠ misplaced alignment tab character & subfig package

When combined with the `subfig` package, while inside a `subfloatrow` `\ffigbox` and



`\ttabbox` must have the caption in the first of the two of the mandatory arguments.

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

⚠ `\FBwidth, \FBheight`

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

### 8.10.5 keyfloat package

Pkg `keyfloat`

⚠ `keywrap`

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

## 8.11 Koma-Script

Cls `komascript`

Many features are ignored during the HTML conversion. The goal is source-level compatibility.

`\titlehead`, `\subject`, `\captionformat`, `\figureformat`, and `\tableformat` are not yet emulated.

⚠ Not fully tested! [Please send bug reports!](#)

Some features have not yet been tested. Please contact the author with any bug reports.

## 8.12 Memoir

Cls `memoir`

⚠ options clash

While emulating `memoir`, `lwarp` pre-loads a number of packages (section 340.1). This can cause an options clash when the user's document later loads the same packages with options. To fix this problem, specify the options before loading `lwarp`:

```

\documentclass{memoir}
...
\PassOptionsToPackage{options_list}{package_name}
...
\usepackage{lwarp}
...
\usepackage{package_name}

```

`\verbfootnote` is not supported.

`\newfootnoteseries`, etc. are not supported.

`lwarp` loads `pagenote` to perform `memoir`'s `pagenote` functions, but there are minor differences in `\pagenotesubhead` and related macros.

Poem numbering is not supported.

The `verbatim` environment does not yet support the `memoir` enhancements. It is currently recommended to load and use `fancyvrb` instead.

The `memoir` glossary system is not yet supported by `lwarpmk`. The `glossaries` package may be used instead, but does require the glossary entries be changed from the `memoir` syntax to the `glossaries` syntax.

## 8.13 Miscellaneous

### 8.13.1 verse and memoir

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

---

```

\begin{warpHTML}

\let\attrib\attribution


\end{warpHTML}

```

---

<p>Len <code>\leftskip</code></p> <p>Len <code>\leftmargini</code></p> <p>Len <code>\TMLvleftskip</code></p> <p>Len <code>\TMLleftmargini</code></p>	<p>These lengths are used by <code>verse</code> and <code>memoir</code> to control the left margin, and they may already be set by the user for print output. New lengths <code>\HTMLvleftskip</code> and <code>\HTMLleftmargini</code> are provided to control the margins in HTML output. These new lengths may be set by the user before any <code>verse</code> environment, and persist until they are manually changed again. One reason to change <code>\HTMLleftmargini</code> is if there</p>
--	---

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.

 **spacing** 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.

### 8.13.2 newclude package

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

 **loading**

---

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

---


### 8.13.3 babel package

Pkg `babel`

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

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

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

 **customized spacing** the user’s changes do not affect the HTML output.

### 8.13.4 todonotes and luatodonotes packages

Pkg `todonotes` The documentation for `todonotes` and `luatodonotes` have an example with a `todo` inside a caption. If this example does not work it will be necessary to move the `todo` outside of the caption.

Pkg `luatodonotes`

### 8.13.5 fixme

Pkg `fixme` External layouts (`\fxloadlayouts`) are not supported.



#### external layouts

User control is provided for setting the HTML styling of the “faces”. The defaults are as follows, and may be changed in the preamble after `fixme` is loaded:

```
\def\FXFaceInlineHTMLStyle{font-weight:bold}
\def\FXFaceEnvHTMLStyle{font-weight:bold}
\def\FXFaceSignatureHTMLStyle{font-style:italic}
\def\FXFaceTargetHTMLStyle{font-style:italic}
```

### 8.13.6 xparse

Pkg `xparse` To remove from the log any warnings about redeclaring objects, place the following before `lwarp` is loaded:

```
\usepackage[log-declarations=false]{xparse}
```

## 9 EPUB conversion

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

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

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

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

To help convert lwarp HTML output to EPUB, add

```
\booltrue{FormatEPUB}
```

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

```
Enter ⇒ lwarpmk cleanall
```

```
Enter ⇒ lwarpmk html
```

```
Enter ⇒ lwarpmk limages
```

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

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

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

 **search order**

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

Preferences → Plugins → File type plugins → HTML to Zip

Check the box Add linked files in breadth first order.

### ⚠ section breaks

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

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

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

Match HTML tags with tag name:

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

//h:h4      — or —      //h:h3

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

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

### *MATHJAX support*

---

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

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

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

---

## 10 Word-processor conversion

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

### 10.1 Activating word-processor conversion

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

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

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

```
\booltrue{FormatWP}
```

formatting adjustments

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

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

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

- Honors several booleans:

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

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

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

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

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

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

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



## 10.2 Additional modifications

---

### *WPMarkFloats*

---

Adds

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

or

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

Bool WPMarkFloats  
Default: false

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

---



---

### *WPMarkMinipages*

---

Adds

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

Bool WPMarkMinipages  
Default: false

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

---



---

### *WPMarkTOC*

---

While formatting for word processors, adds

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

Bool WPMarkTOC  
Default: true

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

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

---

---

*WPMarkLOFT*

---

While formatting for word processors, adds

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

Bool WPMarkLOFT  
Default: false

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

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

---



---

*WPMarkMath*

---

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

Bool WPMarkMath  
Default: false  
Prog TeXMaths  
siunitx

When using the siunitx package, enter

```
\usepackage{siunitx}
```

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

---



---

*WPTitleHeading*

---

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

Bool WPTitleHeading  
Default: false  
section headings

See table 6 on table 6.

---

### 10.3 Recommendations

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

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

Table 6: Section HTML headings for word-processor conversion

Section	HTML headings*			
	With \chapter		Without \chapter	
	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 7 on table 7.

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

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

**siunitx** For siunitx, remember to adjust the preamble as mentioned above.

**LO view border options** LIBREOFFICE has options in the **View** menu to turn on/off the display of thin borders around table cells and text objects.

## 10.4 Limitations

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

Cross references are not actual LIBREOFFICE linked cross references.

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

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

As of when last checked, LIBREOFFICE ignores the following:

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

Libreoffice also has limitations with frames and backgrounds:

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

## 11 Modifying lwarp

To quickly find the source for a package in `lwarp.dtx`, search for `*packagename`, such as `*siunitx`.


Likewise, to quickly find the source for a file in `lwarp.dtx`, search for `*filename`, such as `*lwarp.css`.

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

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

An environment may be converted to a `lateximage` then displayed with an image of the resulting  $\TeX$  output. See section 79 for an example of the `picture` environment.

To create a custom HTML block or inline CSS class, see section 45.8.

  **$\TeX$  boxes** Any  $\TeX$  boxes must be undone, as SVG math or `lateximages` require `\newpage`, which will not work in a  $\TeX$  box.

### 11.1 Modifying a package for lwarp

If a class loads additional packages, it will be required to modify the class for lwarp, since lwarp must be loaded before most other packages.

To work with lwarp, a class must first set up anything which replicates the functions of the basic  $\TeX$  classes, load any required fonts, then load lwarp, then finally load and adjust any other required packages.

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

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

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

Each `lwarp-` package should first call either

```
\LWR@ProvidesPackageDrop
```

or

```
\LWR@ProvidesPackagePass
```

If “Drop”ped, 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 “Pass”ed, the original package is loaded first, with the user-supplied options, then the `lwarp-` version continues loading as well. See section 244 ([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.

### 11.1.1 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 24.1. This causes `lwarp` to stop with an error if `packagename` is loaded before `lwarp`.

## 11.2 Modifying a class for `lwarp`

If a class loads additional packages, it will be required to modify the class for `lwarp`, since `lwarp` must be loaded before most other packages.

To work with `lwarp`, a class must first set up anything which replicates the functions of the basic  $\text{\LaTeX}$  classes, load any required fonts, then load `lwarp`, then finally load and adjust any other required packages.

### 11.3 Testing lwarp

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

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

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

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

### 11.4 Modifying lwarpmk

Prog lwarpmk  
File lwarpmk.lua

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

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

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

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

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

## 12 Troubleshooting

### 12.1 Using the lwarp.sty package

Also see:

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

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

#### Text is not converting:

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

#### Undefined HTML settings:

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

**Tabular problems:** See section 8.9.

#### Obscure error messages:

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

**Options clash:** If using memoir, see section 8.12.

**“Missing \$ inserted.”:** If using a filename or URL in a footnote or `\item`, escape underscores with `\_.`

**“Label(s) may have changed. Rerun to get cross-references right.”:**

This warning may repeat endlessly if a math expression is used in a caption. Simple math expressions such as  $X=1$  may be replaced with

```
\textit{X}\,=\,1
```

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

**“Improper \prevdepth”:** lateximages and svg math require `\newpage`, which cannot work inside  $\text{\TeX}$  boxes or `\ensuremath`. Anything using `\newsavebox`, `\newbox`, `lrbox`, `\savebox`, `\hbox`, `\vbox`, `\usebox`, `\sbox`, etc., must be modified to work without box commands.

If you can locate what used `\ensuremath`, have it temporarily set:

```
\LetLtxMacro\@ensuredmath\LWR@origensuredmath
```

inside a group first.



**Missing sections:**

- See section 7.3 regarding the FileDepth and SideTOCDepth counters, and the use of \tableofcontents in the home page.

**Missing HTML files:**

- See the warning regarding changes to the HTML settings at section 7.3.
- Ensure that the filenames are unique after math and short words are removed. See FileSectionNames at section 7.3.

**Missing / incorrect cross-references:**

- Use lwarpmk again followed by lwarpmk html or lwarpmk print to compile the document one more time.
- Labels with special characters may be a problem. It is best to stick with alpha-numeric, hyphen, underscore, and perhaps the colon (if not French).  
`\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 section 78 to redefine the message which is printed for page number references.

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

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

**Floats out of sequence:**

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

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

**Print document contains HTML tags:**

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

**Images are appearing in strange places:**

labels

⚠ underscores

`\nameref`

⚠ empty link

⚠ cleveref page numbers

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

### SVG images:

#### ⚠ math in `\section`

When using `svg` math in sectioning commands, use `\(` and `\)` to delimit math instead of `$`:

```
\section{Section name with math \((1+2=3)\)}
```

#### ⚠ Adding/removing

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

#### ⚠ HTML instead of images

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

#### ⚠ Lots of files!

Expressing math as `svg` images has the advantage of representing the math exactly as  $\TeX$  would, but has the disadvantage of requiring an individual file for each math expression. For inline math, `lwarp` uses an MD5 hash on its  $\TeX$  source to combine multiple instances of identical inline expressions into a single image file, but display math and other environments such as `picture` and `Tikz` require one image file each. For a document with a large amount of math, see section 6.5 to use `MATHJAX` instead.

### Plain-looking document:

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

### Broken fragments of HTML:

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

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

- Be sure to `lwarpmk clean`, recompile, then start by reloading the home page. You may have been looking at an older version of the document. If you changed a section name, you may have been looking at the file for the old name.
- See the warning regarding changes to the HTML settings at section 7.3.
- Verify that the proper CSS is actually being used.
- The browser may compensate for some subtle changes, such as automatically generating ligatures, reflowing text, etc.

### Un-matched conditional compiles:

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

### 12.1.1 Debug tracing output

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

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

## 12.2 Compiling the `lwarp.dtx` file

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

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

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

### Un-nested environments:

Be sure to properly nest:

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

File 1 **lwarp.sty**

## 13 Implementation

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

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

Packages were patched for re-use where they provided significant functionality. Examples include `xcolor` with its color models and conversion to HTML color output,

and siunitx which provides many number and unit-formatting options, almost all of which are available in pure-text form, and thus easily used by `pdftotext`.

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

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

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

## 14 Section depths and HTML headings

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

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

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

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

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

Table 7: Section depths and HTML headings

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

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

## 15 Source Code

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


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

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

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

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

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

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

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

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

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

Detects X<sub>e</sub>T<sub>E</sub>X and LuaT<sub>E</sub>X:

```
1 \RequirePackage{iftex}
2 \newif\ifxetexorluatex
3 \ifXeTeX
4   \xetexorluatextrue
5 \else
6   \ifLuaTeX
7     \xetexorluatextrue
8   \else
9     \xetexorluatexfalse
10  \fi
11 \fi
12
13 \ifLuaTeX
14 \RequirePackage{luatex85}% until the geometry package is updated
15 \fi
```

## 17 MD5 hashing

The MD5 hash is used for lateximage filenames for svg math.

```
16 \newcommand{\LWR@mdfive}[1]{%
17 \PackageError{lwarp}
18 {No MD5 macro was found.}
19 {Lwarp must find the macros pdfmdfivesum or mdfivesum.}
20 }
21
22 \ifPDFTeX
23 \let\LWR@mdfive\pdfmdfivesum
24 \fi
25
26 \ifLuaTeX
27 \RequirePackage{pdftexcmds}
28 \let\LWR@mdfive\pdf@mdfivesum
29 \fi
30
31 \ifXeTeX
32 \@ifundefined{pdf@mdfivesum}{%
33   {\let\LWR@mdfive\pdfmdfivesum}
```

```

34 \@ifundefined{mdfivesum}{}
35     {\let\LWR@mdfive\mdfivesum}
36 \fi

```

## 18 pdfLaTeX T1 and UTF8 encoding

When using pdfLaTeX, lwarp required T1 and UTF8 encoding.

X<sub>Y</sub>LaTeX and Lua<sub>X</sub>LaTeX are both UTF8 by nature.

```

37 \ifPDFTeX
38 \RequirePackage[T1]{fontenc}
39 \RequirePackage[utf8]{inputenc}
40 \fi

```

## 19 Unicode input characters

**for HTML & PRINT:**

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

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

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

```

41
42 \RequirePackage{newunicodechar}
43
44 \newunicodechar{×}{\texttimes}
45
46 \ifPDFTeX
47 \newunicodechar{ff}{ff}% the first arguments are ligatures
48 \newunicodechar{fi}{fi}
49 \newunicodechar{fl}{fl}
50 \newunicodechar{ffi}{ffi}
51 \newunicodechar{ffl}{ffl}
52 \newunicodechar{--}{--}
53 \newunicodechar{-}{-}

```

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

```

54 \RequirePackage{upquote}
55 \else

```



56 \fi

## 20 Miscellaneous tools

`\LWR@providelength` `{\langle lengthname \rangle}` Provides the length if it isn't defined yet.

Used to provide source compatibility for lengths which will be ignored, but might or might not be already provided by other packages.

```
57 \newcommand*\LWR@providelength[1]{%
58   \ifdeflength{#1}{\newlength{#1}}%
59 }
```

Prints a length in the given units, without printing the unit itself.

`\LWR@convertto` `{\langle dest unit \rangle}{\langle length \rangle}`

```
60 \newcommand*\LWR@convertto[2]{\strip@pt\dimexpr #2*65536/\number\dimexpr 1#1}
```

## 21 Early package requirements

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

Pkg `xpatch` Patches macros with optional arguments.

```
61 \RequirePackage{etoolbox}[2011/01/03]% v2.6 for \BeforeBeginEnvironment, etc.
62 \RequirePackage{xpach}
```

Pkg `ifplatform` Provides `\ifwindows` to try to automatically detect WINDOWS OS.

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

Pkg `letltxmacro` Used to redefine `\textbf` and friends.

```
64 \RequirePackage{letltxmacro}
```

## 22 Operating-System portability

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

### 22.1 Common portability code

Bool usingOSWindows Set if the OSWindows option is used.

```
65 \newbool{usingOSWindows}
66 \boolfalse{usingOSWindows}
```

### 22.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

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

### 22.3 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.

```
68 \newcommand*{\LWR@setOSWindows}
69 {
70 \booltrue{usingOSWindows}
71 \renewcommand*{\OSPathSymbol}{\@backslashchar}
72 }
```

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

```
73 \ifwindows
74 \LWR@setOSWindows
75 \fi
```

## 23 Package options

Pkg `kvoptions` Allows key/value package options.

```
76 \RequirePackage{kvoptions}
77 \SetupKeyvalOptions{family=LWR,prefix=LWR@}
```

Bool `warpingprint`

Bool `warpingHTML`

Bool `mathjax`

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

```
78 \newbool{warpingprint}
79 \newbool{warpingHTML}
80 \newbool{mathjax}
```

**defaults** The default is print output, and svg math if the user chose HTML output.

```
81 \booltrue{warpingprint}%
82 \boolfalse{warpingHTML}%
83 \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.

```
84 \DeclareVoidOption{warpprint}{%
85 \PackageInfo{lwarp}{Using option 'warpprint'}
86 \booltrue{warpingprint}%
87 \boolfalse{warpingHTML}%
88 }
```

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.

```
89 \DeclareVoidOption{warpHTML}{%}
```

```

90 \PackageInfo{lwarp}{Using option 'warpHTML'}%
91 \booltrue{warpingHTML}%
92 \boolfalse{warpingprint}%
93 }

```

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.

```

94 \DeclareVoidOption{mathsvg}{%
95 \PackageInfo{lwarp}{Using option 'mathsvg'}
96 \boolfalse{mathjax}%
97 }

```

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

```

98 \DeclareVoidOption{mathjax}{%
99 \PackageInfo{lwarp}{Using option 'mathjax'}
100 \booltrue{mathjax}%
101 }

```

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.

```

102 \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.

```

103 \DeclareStringOption[english]{IndexLanguage}

```

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

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

```

104 \DeclareStringOption[lwarp.xdy]{xdyFilename}

```

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

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

```

105 \newbool{LWR@creatinglwarpmk}
106 \boolfalse{LWR@creatinglwarpmk}
107
108 \DeclareVoidOption{lwarpmk}{
109 \PackageInfo{lwarp}{Using option 'lwarpmk'}}
110 \booltrue{LWR@creatinglwarpmk}
111 }

```

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.

```

112 \DeclareVoidOption{OSWindows}{
113 \PackageInfo{lwarp}{Using option 'OSWindows'}}
114 \LWR@setOSWindows
115 }

```

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

```

116 \DeclareStringOption[] {HomeHTMLFilename}

```

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

```

117 \DeclareStringOption[] {HTMLFilename}

```

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

```

118 \DeclareBoolOption[false]{latexmk}

```

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

```

119 \ProcessKeyvalOptions*\relax

```

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

```

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

```

Defaults unless already over-ridden by the user:

```

121 \ifcempty{\LWR@HomeHTMLFilename}{
122 \newcommand*{\HomeHTMLFilename}{\BaseJobname}

```

```

123 }{
124 \csedef{HomeHTMLFilename}{\LWR@HomeHTMLFilename}
125 }
126
127 \csedef{HTMLFilename}{\LWR@HTMLFilename}

```

## 23.1 Conditional compilation

`\warpprintonly` `{\<contents>}`

Only process the contents if producing printed output.

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

`\warpHTMLonly` `{\<contents>}`

Only process the contents if producing HTML output.

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

`Pkg`   `comment`   Provides conditional code blocks.

```
130 \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.

```

131 \ifbool{warpingHTML}{
132 \def\DefaultCutFileName{\def\CommentCutFile{comment_html.cut}}
133 }{}
134
135 \ifbool{warpingprint}{
136 \def\DefaultCutFileName{\def\CommentCutFile{comment_print.cut}}
137 }{}

```

```
138 \excludecomment{testing}
```

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

```
139 \includecomment{warpall}
```

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

`Env`   `warpHTML`

For HTML output:

```

140 \ifbool{warpingHTML}{%
141 \includecomment{warpHTML}
142 }
143 {\excludecomment{warpHTML}}%

144 \ifbool{warpingprint}
145 {\includecomment{warpprint}}
146 {\excludecomment{warpprint}}

```

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

```

147 \ifbool{LWR@creatinglwarpmk}
148 {\includecomment{LWR@createlwarpmk}}
149 {\excludecomment{LWR@createlwarpmk}}

```

## 24 Package load order

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

Packages which should only be loaded before lwarp have their own

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

which use `\LWR@loadbefore` to trigger an error if they are loaded after lwarp. Examples include fontspec, inputenc, fontenc, and newunicodechar.

Most packages should be loaded after lwarp. This is enforced by a large number of `\LWR@loadafter` statements, below.

Some packages are emulated by memoir, and so these are tested by `\LWR@notmemoirloadafter`, which does not cause an error if memoir is used.

### 24.1 Tests of package load order

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

```

150 \newcommand*{\LWR@loadafter}[1]{%
151 \@ifpackageloaded{#1}
152 {
153 \PackageError{lwarp}

```

```

154 {Package #1, or one which uses #1, must be loaded after lwarp}
155 {Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{lwarp}.
156 Package #1 may also be loaded by something else, which must also be moved
157 after lwarp.}
158 }
159 {}
160 }

```

`\LWR@notmemoirloadafter` `{\langle packagename \rangle}` Error if not memoir class and this package was loaded before lwarp.

memoir emulates many packages, and pretends that they have already been loaded.

```

161 \@ifclassloaded{memoir}
162 {\newcommand*\LWR@notmemoirloadafter}[1]{} }
163 {\LetLtxMacro\LWR@notmemoirloadafter\LWR@loadafter}

```

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

```

164 \newcommand*\LWR@loadbefore}[1]{%
165 \@ifpackageloaded{#1}
166 {}
167 {
168 \PackageError{lwarp}
169 {Package #1 must be loaded before lwarp}
170 {Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{lwarp}.}
171 }
172 }

```

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

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

```

173 \newcommand*\LWR@loadnever}[2]{%
174 \PackageError{lwarp}
175 {Package #1 is not supported by lwarp's HTML conversion.
176 Package(s) #2 may be useful instead}
177 {Package #1 might conflict with lwarp in some way,
178 or is superceded by another package.
179 For a possible alternative, see package(s) #2.}
180 }

```



## 24.2 Enforcing package loading after lwarp

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:

```

181 \LWR@loadafter{a4}
182 \LWR@loadafter{a4wide}
183 \LWR@loadafter{a5comb}
184 \LWR@notmemoirloadafter{abstract}
185 \LWR@loadafter{acro}
186 \LWR@loadafter{acronym}
187 \LWR@loadafter{adjmulticol}
188 \LWR@loadafter{addlines}
189 \LWR@loadafter{afterpage}
190 \LWR@loadafter{algorithmicx}
191 \LWR@loadafter{alltt}
192 \LWR@loadafter{amsmath}
193 \LWR@loadafter{amsthm}
194 \LWR@loadafter{anonchap}
195 \LWR@loadafter{anysize}
196 \LWR@notmemoirloadafter{appendix}
197 \LWR@loadafter{arabicfront}
198 \LWR@notmemoirloadafter{array}
199 % \LWR@loadafter{atbegshi}% used by morewrites
200 \LWR@loadafter{authblk}
201 \LWR@loadafter{axodraw2}
202 \LWR@loadafter{backref}
203 \LWR@loadafter{balance}
204 \LWR@loadafter{bigdelim}
205 \LWR@loadafter{bigstrut}
206 \LWR@loadafter{blowup}
207 \LWR@loadafter{bookmark}
208 \LWR@notmemoirloadafter{booktabs}
209 \LWR@loadafter{boxedminipage}
210 \LWR@loadafter{boxedminipage2e}
211 \LWR@loadafter{breakurl}
212 \LWR@loadafter{bytefield}
213 \LWR@loadafter{cancel}
214 \LWR@loadafter{caption}
215 \LWR@notmemoirloadafter{ccaption}
216 \LWR@loadafter{changebar}
217 \LWR@notmemoirloadafter{changepage}
218 \LWR@notmemoirloadafter{chngepage}
219 \LWR@loadafter{chappg}
220 \LWR@loadafter{chapterbib}
221 \LWR@loadafter{cite}
222 \LWR@loadafter{color}

```

```
223 \LWR@loadafter{colortbl}
224 \LWR@loadafter{continue}
225 \LWR@notmemoirloadafter{crop}
226 \LWR@loadafter{cuted}
227 \LWR@loadafter{cutwin}
228 \LWR@loadafter{dblfloatfix}
229 \LWR@loadafter{dblfnote}
230 \LWR@notmemoirloadafter{dcolumn}
231 \LWR@loadafter{diagbox}
232 \LWR@loadafter{draftwatermark}
233 \LWR@loadafter{easy-todo}
234 \LWR@loadafter{ebook}
235 \LWR@loadafter{ellipsis}
236 \LWR@loadafter{emptypage}
237 \LWR@loadafter{endfloat}
238 \LWR@loadafter{endheads}
239 \LWR@loadafter{endnotes}
240 \LWR@notmemoirloadafter{enumerate}
241 \LWR@loadafter{enumitem}
242 \LWR@notmemoirloadafter{epigraph}
243 \LWR@loadafter{epstopdf}
244 \LWR@loadafter{eso-pic}
245 \LWR@loadafter{everypage}
246 \LWR@loadafter{everyshi}
247 \LWR@loadafter{extramarks}
248 \LWR@loadafter{fancybox}
249 \LWR@loadafter{fancyhdr}
250 \LWR@loadafter{fancyref}
251 \LWR@loadafter{fancyvrb}
252 \LWR@loadafter{figcaps}
253 \LWR@loadafter{figsize}
254 \LWR@loadafter{fix2col}
255 \LWR@loadafter{fixme}
256 \LWR@loadafter{fixmetodonotes}
257 \LWR@loadafter{flafter}
258 \LWR@loadafter{float}
259 \LWR@loadafter{floatflt}
260 \LWR@loadafter{floatpag}
261 \LWR@loadafter{floatrow}
262 \LWR@loadafter{fltrace}
263 \LWR@loadafter{flushend}
264 \LWR@loadafter{fncychap}
265 \LWR@loadafter{fnlineno}
266 \LWR@loadafter{fnpos}
267 % fontenc must be loaded before lwarp
268 % fontspec must be loaded before lwarp
269 \LWR@loadafter{footmisc}
270 \LWR@loadafter{footnote}
271 \LWR@loadafter{footnotehyper}
272 \LWR@loadafter{footnpag}
```

```
273 \LWR@loadafter{framed}
274 \LWR@loadafter{ftnright}
275 \LWR@loadafter{fullpage}
276 \LWR@loadafter{fullwidth}
277 \LWR@loadafter{fwlw}
278 \LWR@loadafter{geometry}
279 \LWR@loadafter{glossaries}
280 % \LWR@loadafter{graphics}% pre-loaded by xunicode
281 % \LWR@loadafter{graphicx}% pre-loaded by xunicode
282 \LWR@loadafter{grffile}
283 \LWR@loadafter{hang}
284 \LWR@loadafter{hanging}
285 \LWR@loadafter{hypcap}
286 \LWR@loadafter{hypdestopt}
287 \LWR@loadafter{hypernat}
288 \LWR@loadafter{hyperref}
289 \LWR@loadafter{hyperxmp}
290 \LWR@loadafter{hyphenat}
291 \LWR@loadafter{idxlayout}
292 \LWR@loadafter{ifoddpage}
293 \LWR@loadafter{indentfirst}
294 % inputenc must be loaded before lwarp
295 \LWR@loadafter{keyfloat}
296 \LWR@loadafter{layout}
297 \LWR@loadafter{letterspace}
298 \LWR@loadafter{letrine}
299 \LWR@loadafter{lineno}
300 \LWR@loadafter{lips}
301 \LWR@loadafter{listings}
302 \LWR@loadafter{longtable}
303 \LWR@loadafter{lscap}
304 \LWR@loadafter{ltcaption}
305 \LWR@loadafter{ltxttable}
306 \LWR@loadafter{luacolor}
307 \LWR@loadafter{luatodonotes}
308 \LWR@loadafter{marginfit}
309 \LWR@loadafter{marginfix}
310 \LWR@loadafter{marginnote}
311 \LWR@loadafter{mcaption}
312 \LWR@loadafter{mdframed}
313 \LWR@loadafter{memhfixc}
314 \LWR@loadafter{metalogo}
315 \LWR@loadafter{microtype}
316 \LWR@loadafter{midfloat}
317 \LWR@loadafter{midpage}
318 \LWR@loadafter{morefloats}
319 \LWR@notmemoirloadafter{moreverb}
320 % morewrites must be loaded before lwarp
321 \LWR@notmemoirloadafter{mparhack}
322 % \LWR@loadafter{multicol}% loaded by ltxdoc
```

```
323 \LWR@loadafter{multirow}
324 \LWR@loadafter{multitoc}
325 \LWR@loadafter{nameref}
326 \LWR@loadafter{natbib}
327 \LWR@notmemoirloadafter{needspace}
328 % newclude must be loaded before lwarp
329 \LWR@loadafter{newtxmath}
330 % newunicodechar must be loaded before lwarp
331 \LWR@notmemoirloadafter{nextpage}
332 \LWR@loadafter{nonfloat}
333 \LWR@loadafter{nonumonpart}
334 \LWR@loadafter{nopageno}
335 \LWR@loadafter{nowidow}
336 \LWR@loadafter{nththeorem}
337 \LWR@loadafter{overpic}
338 \LWR@loadafter{pagegrid}
339 \LWR@notmemoirloadafter{pagenote}
340 \LWR@loadafter{pagesel}
341 \LWR@loadafter{paralist}
342 \LWR@notmemoirloadafter{parskip}
343 \LWR@loadafter{pbox}
344 \LWR@loadafter{pdfrender}
345 \LWR@loadafter{pdfscape}
346 \LWR@loadafter{pdfsync}
347 \LWR@loadafter{pfnote}
348 \LWR@loadafter{phfqit}
349 \LWR@loadafter{placeins}
350 \LWR@loadafter{prelim2e}
351 \LWR@loadafter{prettyref}
352 \LWR@loadafter{preview}
353 \LWR@loadafter{quotchap}
354 \LWR@loadafter{ragged2e}
355 \LWR@loadafter{realscripts}
356 \LWR@loadafter{relsize}
357 \LWR@loadafter{resizgather}
358 \LWR@loadafter{romanbar}
359 \LWR@loadafter{romanbarpagenumber}
360 \LWR@loadafter{rotating}
361 \LWR@loadafter{rotfloat}
362 \LWR@loadafter{savetrees}
363 % \LWR@loadafter{scalefnt}% loaded by babel-french
364 \LWR@loadafter{schemata}
365 \LWR@loadafter{scrextend}
366 \LWR@loadafter{scrhack}
367 \LWR@loadafter{scrlayer}
368 \LWR@loadafter{scrlayer-notecolumn}
369 \LWR@loadafter{scrlayer-scrpage}
370 \LWR@loadafter{section}
371 \LWR@loadafter{sectionbreak}
372 \LWR@loadafter{sectsty}
```

```
373 \LWR@notmemoirloadafter{setspace}
374 \LWR@loadafter{shadow}
375 \LWR@notmemoirloadafter{showidx}
376 \LWR@loadafter{showkeys}
377 \LWR@loadafter{sidecap}
378 \LWR@loadafter{sidenotes}
379 \LWR@loadafter{siunitx}
380 \LWR@loadafter{soul}
381 \LWR@loadafter{soulpos}
382 \LWR@loadafter{soulutf8}
383 \LWR@loadafter{stabular}
384 \LWR@loadafter{stfloats}
385 \LWR@loadafter{subfig}
386 \LWR@loadafter{subfigure}
387 \LWR@loadafter{supertabular}
388 \LWR@loadafter{tabls}
389 \LWR@notmemoirloadafter{tabularx}
390 \LWR@loadafter{tabulary}
391 \LWR@loadafter{textarea}
392 % \LWR@loadafter{textcomp}% maybe before lwarp with font packages
393 \LWR@loadafter{textfit}
394 \LWR@loadafter{textpos}
395 \LWR@loadafter{theorem}
396 \LWR@loadafter{threeparttable}
397 \LWR@loadafter{tikz}
398 \LWR@loadafter{titleps}
399 \LWR@loadafter{titlesec}
400 \LWR@loadafter{titletoc}
401 \LWR@notmemoirloadafter{titling}
402 % \LWR@loadafter{tocbasic}% preloaded by koma-script classes
403 \LWR@notmemoirloadafter{tocbibind}
404 \LWR@notmemoirloadafter{tocloft}
405 \LWR@loadafter{tocstyle}
406 \LWR@loadafter{todo}
407 \LWR@loadafter{todonotes}
408 \LWR@loadafter{transparent}
409 \LWR@loadafter{trivfloat}
410 \LWR@loadafter{turnthepage}

411 % \LWR@loadafter{typearea}% preloaded by koma-script classes
412 \LWR@loadafter{ulem}
413 \LWR@loadafter{upref}
414 \LWR@loadafter{varioref}% no lwarp package provided
415 \LWR@notmemoirloadafter{verse}
416 \LWR@loadafter{vertbars}
417 \LWR@loadafter{vmargin}
418 \LWR@loadafter{vwcol}
419 \LWR@loadafter{wallpaper}
420 \LWR@loadafter{watermark}
```

```

421 \LWR@loadafter{wrapfig}
422 \LWR@loadafter{xcolor}
423 \LWR@loadafter{xfrac}
424 \LWR@loadafter{xltextra}
425 \LWR@loadafter{xmpincl}
426 \LWR@loadafter{xtab}
427 \LWR@loadafter{xurl}
428 \LWR@loadafter{xy}
429 \LWR@loadafter{zwpgelayout}

```

## 25 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.

**for HTML output:** 430 \begin{warpHTML}

Load fontspec if necessary:

```

431 \ifxetexorluatex
432 \@ifpackageloaded{fontspec}{\fontspec}{}
433 \usepackage[no-math]{fontspec}
434 }

```

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

```

435 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
436 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
437 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
438 \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  $\TeX$  dash and quote ligatures, which may fail on older browsers but at least won't corrupt written words.

```

439 \RequirePackage {microtype}
440
441 \microtypesetup{
442   protrusion=false,
443   expansion=false,

```

```

444     tracking=false,
445     kerning=false,
446     spacing=false}
447
448 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}
449 \fi
450 \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:**

```

451 \begin{warpHTML}
452 \RequirePackage[paperheight=190in,paperwidth=20in,%
453 left=2in,right=12in,%
454 top=1in,bottom=1in,%
455 ]{geometry}
456 \@twosidefalse
457 \@mparswitchfalse
458 \end{warpHTML}

```

**for HTML & PRINT:**

```

459 \begin{warpall}

```

Pkg `xparse`

$\LaTeX$ 3 command argument parsing

```

460 \RequirePackage{xparse}

```

Pkg `afterpackage` Used to patch titling to add `\AddSubtitlePublished`. Provided by `scrfile` or `afterpackage`.

```

461 \@ifundefined{AfterPackage}%
462 {
463 \RequirePackage{afterpackage}
464 }{}
465 \end{warpall}

```

**for HTML output:** 466 \begin{warpHTML}

Pkg expl3

LaTeX3 programming

467 \RequirePackage{expl3}

Pkg gettitlestring

Used to emulate \nameref.

468 \RequirePackage{gettitlestring}

Pkg everyhook

everyhook is used to patch paragraph handling.

469 \RequirePackage{everyhook}

470 \end{warpHTML}

**for HTML & PRINT:** 471 \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>

472 \RequirePackage{filecontents}

473

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

475 {}

476 {

477 \newwrite\fcwrite

478 \let\LWR@origfilec@ntents\filec@ntents

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

480 }

481 \end{warpall}

**for HTML output:** 482 \begin{warpHTML}

Pkg xifthen

483 \RequirePackage{xifthen}



Pkg xstring

484 \RequirePackage{xstring}

Pkg xstring

485 \RequirePackage{verbatim}

Pkg makeidx

486 \RequirePackage{makeidx}

487 \makeindex

Pkg calc

488 \RequirePackage{calc}

Pkg refcount

Provides \setcounterref, \setcounterpageref, etc.

489 \RequirePackage{refcount}

Pkg newfloat

490 \RequirePackage{newfloat}

491 \end{warpHTML}

**for HTML & PRINT:** 492 \begin{warpall}

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

493 \RequirePackage{environ}

494 \end{warpall}

**for HTML output:** 495 \begin{warpHTML}

Pkg zref Used for cross-references.

496 \RequirePackage{zref}

Pkg amsmath Preloaded to avoid options clash and to add patches.

Equation numbers are placed to the left for HTML.

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

```
497 % \PassOptionsToPackage{leqno}{amsmath}% disabled to test centered display math
498 \RequirePackage{amsmath}
```

Patches to allow \eqref inside a caption:

```
499 \def\maketag@@@#1{\text{#1}}
500 \def\tagform@#1{\maketag@@{(\ignorespaces#1\unskip)}}
```

Pkg printlen Used to convert lengths for image width/height options.

```
501 \RequirePackage{printlen}
```

\LWR@printlength  $\{\langle length \rangle\}$

Prints a length using a locally-controlled unit and space. Rounding is used unless the length is small.

```
502 \newrobustcmd*\LWR@printlength}[1]{%
503 \begingroup%
504 \uselengthunit{PT}%
505 \renewcommand*\unitspace}{}%
506 \ifdimless{#1}{10pt}{%
507   \printlength{#1}%
508 }{%
509   \rndprintlength{#1}%
510 }%
511 \endgroup%
512 }
```

```
513 \end{warpHTML}
```

for PRINT output: 514 \begin{warpprint}

Pkg varwidth Used for print-mode lateximage:

```
515 \RequirePackage{varwidth}
```

```
516 \end{warpprint}
```

## 26 Loading packages

for HTML output: 517 \begin{warpHTML}

Remember the original \RequirePackage:

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

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

```
519 \newcommand*{\LWR@requirepackagenames}{}%
```

\LWR@parsedrequirepackagenames Stores the parsed list of required package names after spaces are removed and lwarp- is prepended.

```
520 \newcommand*{\LWR@parsedrequirepackagenames}{}%
```

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

Note that argument 4 is passed directly to \StrBetween.

```
521 \newcommand*\LWR@findword[3][,]{%
522   \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
523 }
```

\LWR@lookforpackagename {*index*}

If this is a package name, re-direct it to the lwarp version by renaming it lwarp- followed by the original name.

Looks index deep into the list of package names, \LWR@requirepackagenames, and builds \LWR@parsedrequirepackagenames which is the modified list of names.

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

Find the index'th package name from the list:

```
525 \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.

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

See if the package name was found:

```
527 \IfStrEq{\LWR@strresulttwo}{}%
528 {%
529 }% no filename
530 {% yes filename
```

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

```

531 \IfFileExists{lwarp-\LWR@strresulttwo.sty}%
532 {% lwarp-* file found
533 \ifdefvoid{\LWR@parsedrequirepackagenames}{%
534 \edef\LWR@parsedrequirepackagenames{lwarp-\LWR@strresulttwo}%
535 }{%
536 \edef\LWR@parsedrequirepackagenames{%
537 \LWR@parsedrequirepackagenames,lwarp-\LWR@strresulttwo%
538 }%
539 }%
540 }%
541 {%

542 \ifdefvoid{\LWR@parsedrequirepackagenames}{%
543 \edef\LWR@parsedrequirepackagenames{\LWR@strresulttwo}%
544 }{%
545 \edef\LWR@parsedrequirepackagenames{%
546 \LWR@parsedrequirepackagenames,\LWR@strresulttwo%
547 }%
548 }%
549 }% no lwarp-* file
550 }% yes filename
551 }

```

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

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

```

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

```

Redirect up to nine names:

```

553 \renewcommand*\LWR@requirepackagenames{#2}%
554 \renewcommand*\LWR@parsedrequirepackagenames{}%
555 \LWR@lookforpackagename{1}%
556 \LWR@lookforpackagename{2}%
557 \LWR@lookforpackagename{3}%
558 \LWR@lookforpackagename{4}%
559 \LWR@lookforpackagename{5}%
560 \LWR@lookforpackagename{6}%
561 \LWR@lookforpackagename{7}%
562 \LWR@lookforpackagename{8}%
563 \LWR@lookforpackagename{9}%

```

`\RequirePackage` depending on the options and version:

```

564 \IfValueTF{#1}%
565 {% options given
566   \IfValueTF{#3}% version given?
567   {\LWR@origRequirePackage[#1]{\LWR@parsedrequirepackagenames}[#3]}%
568   {\LWR@origRequirePackage[#1]{\LWR@parsedrequirepackagenames}}%
569 }%
570 {% no options given
571   \IfValueTF{#3}% version given?
572   {\LWR@origRequirePackage{\LWR@parsedrequirepackagenames}[#3]}%
573   {\LWR@origRequirePackage{\LWR@parsedrequirepackagenames}}%
574 }%
575 }
576 \LetLtxMacro{\usepackage}{\RequirePackage}

```

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

Uses the original package, including options.

```

577 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
578 \PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
579 \IfValueTF{#2}
580 {\ProvidesPackage{lwarp-#1}[#2]}
581 {\ProvidesPackage{lwarp-#1}}
582 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
583 \ProcessOptions\relax
584 \IfValueTF{#2}
585 {\LWR@origRequirePackage{#1}[#2]}
586 {\LWR@origRequirePackage{#1}}
587 }

```

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

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

```

588 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
589 \PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
590 \IfValueTF{#2}
591 {\ProvidesPackage{lwarp-#1}[#2]}
592 {\ProvidesPackage{lwarp-#1}}

```

Ignore all options.

```

593 \DeclareOption*{}

```

Nullifies then processes the options. Seems to be required when options contain curly braces, which were causing “Missing `\begin{document}`”.

```

594 % \ProcessOptions\relax% original LaTeX code
595 \let\ds@\@empty%          from the original \ProcessOptions
596 \edef\@curroptions{}%      lwarp modification to \ProcessOptions
597 \@processoptions\relax%    from the original \ProcessOptions
598 }

599 \end{warpHTML}

```

## 27 Additional required packages

**for HTML output:** 600 \begin{warpHTML}

Pkg caption

```

601 \RequirePackage{caption}%
602 \end{warpHTML}

```

## 28 File handles

Defines file handles for writes.

**for HTML & PRINT:** 603 \begin{warpall}

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

```

604 \newwrite\LWR@quickfile%
605 \end{warpall}

```

**for HTML output:** 606 \begin{warpHTML}

\LWR@lateximagesfile For lateximages.txt.

```

607 \newwrite\LWR@lateximagesfile
608 \end{warpHTML}

```

## 29 Include a file

During HTML output, `\include{<filename>}` causes the following to occur:

1. lwarp creates `<filename>_html_inc.tex` whose contents are:  

```
\input <filename>.tex
```
2. `<filename>_html_inc.tex` is then `\included` instead of `<filename>.tex`.
3. `<filename>_html_inc.aux` is automatically generated and used by  $\LaTeX$ .

for HTML output: 609 `\begin{warpHTML}`

`\include {<filename>}`

`\@include {<filename>}` Modified to load `_html_inc` files.

```

610 \def\@include#1 {%
611 \immediate\openout\LWR@quickfile #1_html_inc.tex% lwarp
612 \immediate\write\LWR@quickfile{\string\input{#1.tex}}% lwarp
613 \immediate\closeout\LWR@quickfile% lwarp
614 \LWR@origclearpage% \changed
615 \if@filesw
616   \immediate\write\@mainaux{\string\@input{#1_html_inc.aux}}% changed
617 \fi
618 \@tempswatrue
619 \if@partsw
620   \@tempswafalse
621   \edef\reserved@a{#1}%
622   \@for\reserved@a:=\@partlist\do
623   {\ifx\reserved@a\reserved@b\@tempswatrue\fi}%
624 \fi
625 \if@tempswa
626   \let\@auxout\@partaux
627   \if@filesw
628     \immediate\openout\@partaux #1_html_inc.aux % changed
629     \immediate\write\@partaux{\relax}%
630   \fi
631   \@input{#1_html_inc.tex}% changed
632   \LWR@origclearpage% changed
633   \@writeckpt{#1}%
634   \if@filesw
635     \immediate\closeout\@partaux
636   \fi
637 \else
638   \deadcycles\z@

```

```

639 \@nameuse{cp@#1}%
640 \fi
641 \let\@auxout\@mainaux%
642 }

643 \end{warpHTML}

```

## 30 Copying a file

for HTML output: 644 \begin{warpHTML}

\LWR@copyfile {<source filename>} {<destination filename>}

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

```

645 \newwrite\LWR@copyoutfile % open the file to write to
646 \newread\LWR@copyinfile % open the file to read from
647
648 \newcommand*{\LWR@copyfile}[2]{%
649 \LWR@traceinfo{\LWR@copyfile: copying #1 to #2}
650
651 \immediate\openout\LWR@copyoutfile=#2
652 \openin\LWR@copyinfile=#1
653 \begingroup\endlinechar=-1
654 \makeatletter
655
656 \LWR@traceinfo{\LWR@copyfile: about to loop}
657
658 \loop\unless\ifeof\LWR@copyinfile
659 \LWR@traceinfo{\LWR@copyfile: one line}
660 \read\LWR@copyinfile to\LWR@fileline % Read one line and store it into \LWR@fileline
661 % \LWR@fileline\par % print the content into the pdf
662 % print the content:
663 \immediate\write\LWR@copyoutfile{\unexpanded\expandafter{\LWR@fileline}}%
664 \repeat
665 \immediate\closeout\LWR@copyoutfile
666 \LWR@traceinfo{\LWR@copyfile: done}
667 \endgroup
668 }

669 \end{warpHTML}

```



## 31 Debugging messages

**for HTML & PRINT:** 670 \begin{warpall}

Bool LWR@tracinglwarp True if tracing is turned on.

671 \newbool{LWR@tracinglwarp}

\tracinglwarp Turns on the debug tracing messages.

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

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

```
673 \newcommand{\LWR@traceinfo}[1]{%
674 \ifbool{LWR@tracinglwarp}%
675 {%
676     \typeout{*** lwarp: #1}%
677     % \PackageInfo{lwarp}{#1 : }%
678 }%
679 {}%
680 }
```

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

Default: false

```
681 \newbool{HTMLDebugComments}
682 \boolfalse{HTMLDebugComments}
```

If \tracinglwarp, show where preamble hooks occur:

```
683 \AfterEndPreamble{
684 \LWR@traceinfo{AfterEndPreamble}
685 }
686
687 \AtBeginDocument{
688 \LWR@traceinfo{AtBeginDocument}
689 }
```

690 \end{warpall}

## 32 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:** 691 \begin{warpall}

### 32.1 User-level controls

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

```
692 \newbool{FormatEPUB}
693 \boolfalse{FormatEPUB}
```

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

```
694 \newbool{FormatWP}
695 \boolfalse{FormatWP}
```

**Bool** **WPMarkFloats** Adds  
**Default: false**

```

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

or

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

around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames and captions.<sup>14</sup>

```
696 \newbool{WPMarkFloats}
697 \boolfalse{WPMarkFloats}
```

**Bool** **WPMarkMinipages** Adds  
**Default: false**

```

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

<sup>14</sup>Perhaps some day word processors will have HTML import options for identifying <figure> and <figcaption> tags for figures and tables.

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

```
698 \newbool{WPMarkMinipages}
699 \boolfalse{WPMarkMinipages}
```

Bool WPMarkTOC While formatting for word processors, adds  
Default: true  
=== table of contents ===

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

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

```
700 \newbool{WPMarkTOC}
701 \booltrue{WPMarkTOC}
```

Bool WPMarkLOFT While formatting for word processors, adds  
Default: false  
=== list of figures === and/or  
=== list of tables ===

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

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

```
702 \newbool{WPMarkLOFT}
703 \boolfalse{WPMarkLOFT}
```

Bool WPMarkMath While formatting for word processors, prints math as  $\LaTeX$  code instead of creating SVG images or MATHJAX. This is useful for cut/paste into the LibreOffice Writer TeXMaths extension.

```
704 \newbool{WPMarkMath}
705 \boolfalse{WPMarkMath}
```

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

See table 6 on table 6.

```
706 \newbool{WPTitleHeading}
707 \boolfalse{WPTitleHeading}
```

```
708 \end{warpall}
```

## 32.2 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: 709 \begin{warpHTML}

710 \AtBeginDocument{
711 \ifbool{FormatWP}{
712 \@ifundefined{chapter}{
713 \ifbool{WPTitleHeading}{% part and section starting at h2
714 \renewcommand*\LWR@tagtitle}{h1}
715 \renewcommand*\LWR@tagtitleend}{/h1}
716 \renewcommand*\LWR@tagpart}{h2}
717 \renewcommand*\LWR@tagpartend}{/h2}
718 \renewcommand*\LWR@tagsection}{h3}
719 \renewcommand*\LWR@tagsectionend}{/h3}
720 \renewcommand*\LWR@tagsubsection}{h4}
721 \renewcommand*\LWR@tagsubsectionend}{/h4}
722 \renewcommand*\LWR@tagsubsubsection}{h5}
723 \renewcommand*\LWR@tagsubsubsectionend}{/h5}
724 \renewcommand*\LWR@tagparagraph}{h6}
725 \renewcommand*\LWR@tagparagraphend}{/h6}
726 \renewcommand*\LWR@tagsubparagraph}{span class="subparagraph"}
727 \renewcommand*\LWR@tagsubparagraphend}{/span}
728 }% WPTitleHeading
729 {% not WPTitleHeading, part and section starting at h1
730 \renewcommand*\LWR@tagtitle}{div class="title"}
731 \renewcommand*\LWR@tagtitleend}{/div}
732 \renewcommand*\LWR@tagpart}{h1}
733 \renewcommand*\LWR@tagpartend}{/h1}
734 \renewcommand*\LWR@tagsection}{h2}
735 \renewcommand*\LWR@tagsectionend}{/h2}
736 \renewcommand*\LWR@tagsubsection}{h3}
737 \renewcommand*\LWR@tagsubsectionend}{/h3}
738 \renewcommand*\LWR@tagsubsubsection}{h4}
739 \renewcommand*\LWR@tagsubsubsectionend}{/h4}
740 \renewcommand*\LWR@tagparagraph}{h5}
741 \renewcommand*\LWR@tagparagraphend}{/h5}
742 \renewcommand*\LWR@tagsubparagraph}{h6}
743 \renewcommand*\LWR@tagsubparagraphend}{/h6}
744 }% not WPTitleHeading
745 }% chapter undefined
```

```

746 {% chapter defined
747 \ifbool{WPTitleHeading}{%
748 {% not WPTitleHeading, part and chapter starting at h1
749 \renewcommand*{\LWR@tagtitle}{div class="title"}
750 \renewcommand*{\LWR@tagtitleend}{/div}
751 \renewcommand*{\LWR@tagpart}{h1}
752 \renewcommand*{\LWR@tagpartend}{/h1}
753 \renewcommand*{\LWR@tagchapter}{h2}
754 \renewcommand*{\LWR@tagchapterend}{/h2}
755 \renewcommand*{\LWR@tagsection}{h3}
756 \renewcommand*{\LWR@tagsectionend}{/h3}
757 \renewcommand*{\LWR@tagsubsection}{h4}
758 \renewcommand*{\LWR@tagsubsectionend}{/h4}
759 \renewcommand*{\LWR@tagsubsubsection}{h5}
760 \renewcommand*{\LWR@tagsubsubsectionend}{/h5}
761 \renewcommand*{\LWR@tagparagraph}{h6}
762 \renewcommand*{\LWR@tagparagraphend}{/h6}
763 \renewcommand*{\LWR@tagsubparagraph}{span class="subparagraph"}
764 \renewcommand*{\LWR@tagsubparagraphend}{/span}
765 }% not WPTitleHeading
766 }% chapter defined
767 }{}% FormatWP
768 }% AtBeginDocument

769 \end{warpHTML}

```

### 33 Remembering original formatting macros

for HTML output: 770 \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.

```

771 \LetLtxMacro\LWR@origmbox\mbox
772 \LetLtxMacro\LWR@origmakebox\makebox
773
774 \let\LWR@origcomma\,
775 \let\LWR@origtilde~
776 \let\LWR@origenskip\enskip
777 \let\LWR@origquad\quad
778 \let\LWR@origqqquad\qqquad
779
780 \let\LWR@orighspace\hspace
781 \let\LWR@orighfill\hfill
782 \let\LWR@origfil\hfil

```

```
783 \let\LWR@origvspace\vspace
784 \let\LWR@orighss\hss
785 \let\LWR@origllap\llap
786 \let\LWR@origrlap\rlap
787 \let\LWR@origfilneg\hfilneg
788
789 \let\LWR@origrule\rule
790 \let\LWR@orighrulefill\hrulefill
791 \let\LWR@origdotfill\dotfill
792
793 \let\LWR@origmedskip\medskip
794 \let\LWR@origbigskip\bigskip
795
796 \let\LWR@origtextellipsis\textellipsis
797
798 \let\LWR@orignormalsize\normalsize
799 \let\LWR@origsmall\small
800 \let\LWR@origfootnotesize\footnotesize
801 \let\LWR@origscriptsize\scriptsize
802 \let\LWR@origtiny\tiny
803 \let\LWR@origlarge\large
804 \let\LWR@origLarge\Large
805 \let\LWR@origLARGE\LARGE
806 \let\LWR@orighuge\huge
807 \let\LWR@origHuge\Huge
808
809 \LetLtxMacro{\LWR@origtextrm}{\textrm}
810 \LetLtxMacro{\LWR@origtextsf}{\textsf}
811 \LetLtxMacro{\LWR@origtexttt}{\texttt}
812 \LetLtxMacro{\LWR@origtextnormal}{\textnormal}
813 \LetLtxMacro{\LWR@origtextbf}{\textbf}
814 \LetLtxMacro{\LWR@origtextmd}{\textmd}
815 \LetLtxMacro{\LWR@origtextit}{\textit}
816 \LetLtxMacro{\LWR@origtextsl}{\textsl}
817 \LetLtxMacro{\LWR@origtextsc}{\textsc}
818 \LetLtxMacro{\LWR@origtextup}{\textup}
819 \LetLtxMacro{\LWR@origemph}{\emph}
820
821 \LetLtxMacro{\LWR@origrmfamily}{\rmfamily}
822 \LetLtxMacro{\LWR@origsffamily}{\sffamily}
823 \LetLtxMacro{\LWR@origttfamily}{\ttfamily}
824 \LetLtxMacro{\LWR@origbfseries}{\bfseries}
825 \LetLtxMacro{\LWR@origmdseries}{\mdseries}
826 \LetLtxMacro{\LWR@origupshape}{\upshape}
827 \LetLtxMacro{\LWR@origslshape}{\slshape}
828 \LetLtxMacro{\LWR@origscshape}{\scshape}
829 \LetLtxMacro{\LWR@origitshape}{\itshape}
830 \LetLtxMacro{\LWR@origem}{\em}
831 \LetLtxMacro{\LWR@orignormalfont}{\normalfont}
832
```

```
833 \let\LWR@origraggedright\raggedright
834 \let\LWR@origonecolumn\onecolumn
835
836 \let\LWR@origsp\sp
837 \let\LWR@origsb\sb
838 \LetLtxMacro\LWR@origtextsuperscript\textsuperscript
839 \LetLtxMacro\LWR@orig@textsuperscript\@textsuperscript
840
841 \AtBeginDocument{
842 \LetLtxMacro\LWR@origtextsubscript\textsubscript
843 \LetLtxMacro\LWR@orig@textsubscript\@textsubscript
844 }
845
846 \LetLtxMacro\LWR@origunderline\underline


847 \let\LWR@origraggedright\raggedright
848 \let\LWR@origraggedleft\raggedleft
849 \let\LWR@origcentering\centering


850 \let\LWR@orignewpage\newpage
851
852 \let\LWR@origpagestyle\pagestyle
853 \let\LWR@origthispagestyle\thispagestyle
854 \LetLtxMacro\LWR@origpagenumbering\pagenumbering
855
856 \LetLtxMacro{\LWR@origminipage}{\minipage}
857 \let\LWR@origendminipage\endminipage
858 \LetLtxMacro{\LWR@origparbox}{\parbox}
859
860 \let\LWR@orignewline\newline
861
862
863 \AtBeginDocument{% in case packages change definition
864 \let\LWR@orig@trivlist\@trivlist
865 \let\LWR@origtrivlist\trivlist
866 \let\LWR@origendtrivlist\endtrivlist
867 \LetLtxMacro\LWR@origitem\item
868 \LetLtxMacro\LWR@origitemize\itemize
869 \LetLtxMacro\LWR@endorigitemize\enditemize
870 \LetLtxMacro\LWR@origenumerate\enumerate
871 \LetLtxMacro\LWR@endorigenumerate\endenumerate
872 \LetLtxMacro\LWR@origdescription\description
873 \LetLtxMacro\LWR@endorigdescription\enddescription
874 \let\LWR@orig@mklab@mklab
875 \let\LWR@origmakelabel\makelabel
876 \let\LWR@orig@donoparitem\@donoparitem
877 \LetLtxMacro\LWR@orig@item\@item
878 \let\LWR@orig@nbitem\@nbitem
879 }
```

```

880
881 \let\LWR@origpar\par
882
883 \LetLtxMacro{\LWR@origfootnote}{\footnote}
884 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
885
886 \let\LWR@origclearpage\clearpage
887
888
889 \AtBeginDocument{% in case packages change definition
890 \LetLtxMacro\LWR@orighline\hline%
891 \LetLtxMacro\LWR@origcline\cline%
892 }

893 \end{warpHTML}

```

## 34 Accents

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

for HTML output: 894 \begin{warpHTML}

Without `\AtBeginDocument`, `\t` was being re-defined somewhere.

```
895 \AtBeginDocument{
```

The following are restored for print when inside a `lateximage`.

For Unicode engines, only `\t` needs to be redefined:

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

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

```

897 \ifPDFTeX
898 \LetLtxMacro{\LWR@origequalaccent}{\=}
899 \LetLtxMacro{\LWR@origdotaccent}{\cdot}
900 \LetLtxMacro{\LWR@origu}{\u}
901 \LetLtxMacro{\LWR@origv}{\v}
902 \LetLtxMacro{\LWR@origc}{\c}
903 \LetLtxMacro{\LWR@origd}{\d}
904 \LetLtxMacro{\LWR@origb}{\b}

```

The HTML redefinitions follow.



For pdf<sub>La</sub>TeX, Unicode diacritical marks are used:

```

905 \renewcommand*{\=} [1] {\#1\HTMLUnicode{0305}}
906 \renewcommand*{\.} [1] {\#1\HTMLUnicode{0307}}
907 \renewcommand*{\u} [1] {\#1\HTMLUnicode{0306}}
908 \renewcommand*{\v} [1] {\#1\HTMLUnicode{030C}}
909 \renewcommand*{\c} [1] {\#1\HTMLUnicode{0327}}
910 \renewcommand*{\d} [1] {\#1\HTMLUnicode{0323}}
911 \renewcommand*{\b} [1] {\#1\HTMLUnicode{0331}}
912 \fi

```

For all engines, a Unicode diacritical tie is used:

```

913 \def\LWR@t#1#2{\#1\HTMLUnicode{0361}#2}
914 \renewcommand*{\t} [1] {\LWR@t#1}

```

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

```

915 \ifPDFTeX
916 \newcommand*{\LWR@restoreorigaccents}{%
917 \LetLtxMacro{\=}{\LWR@origequalaccent}%
918 \LetLtxMacro{\.}{\LWR@origdotaccent}%
919 \LetLtxMacro{\u}{\LWR@origu}%
920 \LetLtxMacro{\v}{\LWR@origv}%
921 \LetLtxMacro{\t}{\LWR@origt}%
922 \LetLtxMacro{\c}{\LWR@origc}%
923 \LetLtxMacro{\d}{\LWR@origd}%
924 \LetLtxMacro{\b}{\LWR@origb}%
925 }
926 \else% XeLaTeX, LuaLaTeX:
927 \newcommand*{\LWR@restoreorigaccents}{%
928 \LetLtxMacro{\t}{\LWR@origt}%
929 }
930 \fi
931 }% AtBeginDocument

932 \end{warpHTML}

```

## 35 Configuration Files

```

933 \begin{warpprint}
934 \typeout{lwarp: generating configuration files}
935 \end{warpprint}

```

### 35.1 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.

```

936 \begin{warpprint}
937 \immediate\openout\LWR@quickfile=\jobname_html.tex
938 \immediate\write\LWR@quickfile{%
939 \detokenize{\PassOptionsToPackage}%
940 {warpHTML,BaseJobname=\jobname}{lwarp}%
941 }
942 \immediate\write\LWR@quickfile{%
943 \detokenize{\input}\string{\jobname.tex\string }%
944 }
945 \immediate\closeout\LWR@quickfile
946 \end{warpprint}

```

### 35.2 lwarpmk.conf

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

An example `lwarpmk.conf`:

---

```

opssystem = "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:**

```

947 \begin{warpprint}
948 \ifcsdef\LWR@quickfile\{}{\newwrite\LWR@quickfile}
949 \immediate\openout\LWR@quickfile=lwarpmk.conf
950 \ifbool{usingOSWindows}{
951 \immediate\write\LWR@quickfile{opssystem = "Windows"}
952 }{
953 \immediate\write\LWR@quickfile{opssystem = "Unix"}
954 }
955 \ifPDFTeX
956 \immediate\write\LWR@quickfile{latexname = "pdflatex"}
957 \fi

```

```

958 \ifXeTeX
959 \immediate\write\LWR@quickfile{latexname = "xelatex"}
960 \fi
961 \ifLuaTeX
962 \immediate\write\LWR@quickfile{latexname = "lualatex"}
963 \fi
964 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
965 \immediate\write\LWR@quickfile{%
966 homehtmlfilename = "\HomeHTMLFilename"%
967 }
968 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
969 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
970 \immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
971 \immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
972 \immediate\closeout\LWR@quickfile
973 \end{warpprint}

```

### 35.3 project.lwarpmkconf

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

```

974 \begin{warpprint}
975 \ifcsdef{LWR@quickfile}{}{\newwrite{LWR@quickfile}}
976 \immediate\openout\LWR@quickfile=\jobname.lwarpmkconf
977 \ifbool{usingOSWindows}{
978 \immediate\write\LWR@quickfile{opsystem = "Windows"}
979 }{
980 \immediate\write\LWR@quickfile{opsystem = "Unix"}
981 }
982 \ifPDFTeX
983 \immediate\write\LWR@quickfile{latexname = "pdflatex"}
984 \fi
985 \ifXeTeX
986 \immediate\write\LWR@quickfile{latexname = "xelatex"}
987 \fi
988 \ifLuaTeX
989 \immediate\write\LWR@quickfile{latexname = "lualatex"}
990 \fi
991 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
992 \immediate\write\LWR@quickfile{%
993 homehtmlfilename = "\HomeHTMLFilename"%
994 }
995 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
996 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
997 \immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
998 \immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
999 \immediate\closeout\LWR@quickfile

```

```
1000 \end{warpprint}
```

## 35.4 lwarp.css

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

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

```
1001 \begin{warpprint}
1002 \begin{filecontents*}{lwarp.css}
1003 /*
1004  CSS stylesheet for the LaTeX lwarp package
1005  Copyright 2016-2018 Brian Dunn -- BD Tech Concepts LLC
1006 */
1007
1008
1009 /* a fix for older browsers: */
1010 header, section, footer, aside, nav, main,
1011     article, figure { display: block; }
1012
1013
1014 A:link {color:#000080 ; text-decoration: none ; }
1015 A:visited {color:#800000 ; }
1016 A:hover {color:#000080 ; text-decoration: underline ;}
1017 A:active {color:#800000 ; }
1018
1019 a.tocpart {display: inline-block ; margin-left: 0em ;
1020     font-weight: bold ;}
1021 a.tocchapter {display: inline-block ; margin-left: 0em ;
1022     font-weight: bold ;}
1023 a.tocsection {display: inline-block ; margin-left: 1em ;
1024     text-indent: -.5em ; font-weight: bold ; }
1025 a.tocsubsection {display: inline-block ; margin-left: 2em ;
1026     text-indent: -.5em ; }
1027 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
1028     text-indent: -.5em ; }
1029 a.tocparagraph {display: inline-block ; margin-left: 4em ;
1030     text-indent: -.5em ; }
1031 a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
1032     text-indent: -.5em ; }
1033 a.tocfigure {margin-left: 0em}
1034 a.tocsubfigure {margin-left: 2em}
1035 a.tocatable {margin-left: 0em}
1036 a.tocsubtable {margin-left: 2em}
1037 a.toctheorem {margin-left: 0em}
1038 a.toclstlisting {margin-left: 0em}
```

```
1039
1040 body {
1041     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1042         "Lucida Bright", Georgia, serif;
1043     background: #FAF7F4 ;
1044     color: black ;
1045     margin: 0em ;
1046     padding: 0em ;
1047     font-size: 100% ;
1048     line-height: 1.2 ;
1049 }
1050
1051 p {margin: 1.5ex 0em 1.5ex 0em ;}
1052 table p {margin: .5ex 0em .5ex 0em ;}
1053
1054 /* Holds a section number to add space between it and the name */
1055 span.sectionnumber { margin-right: 0em }
1056
1057 /* Inserted in front of index lines */
1058 span.indexitem {margin-left: 0em}
1059 span.indexsubitem {margin-left: 2em}
1060 span.indexsubsubitem {margin-left: 4em}
1061
1062 div.hidden, span.hidden { display: none ; }
1063
1064 kbd {
1065     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1066         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1067         "Courier New", monospace;
1068     font-size: 100% ;
1069 }
1070
1071 pre { padding: 3pt ; }
1072
1073 span.strong { font-weight: bold; }
1074
1075 span.textmd { font-weight: normal; }
1076
1077 span.textsc { font-variant: small-caps; }
1078
1079 span.textsl { font-style: oblique; }
1080
1081 span.textup { font-variant: normal; }
1082
1083 span.textrm {
1084     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1085         "Lucida Bright", Georgia, serif;
1086 }
1087
1088 span.textsf {
```

```
1089     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1090         Geneva, Verdana, sans-serif ;
1091 }
1092
1093 span.textcircled { border: 1px solid black ; border-radius: 1ex ; }
1094
1095 span.underline {
1096     text-decoration: underline ;
1097     text-decoration-skip ;
1098 }
1099
1100 span.overline {
1101     text-decoration: overline ;
1102     text-decoration-skip ;
1103 }
1104
1105 /* for diagbox */
1106 div.diagboxtitleN { border-bottom: 1px solid gray }
1107 div.diagboxtitleS { border-top: 1px solid gray }
1108
1109 div.diagboxE {
1110     padding-left: 2em ;
1111     text-align: right ;
1112 }
1113
1114 div.diagboxW {
1115     padding-right: 2em ;
1116     text-align: left ;
1117 }
1118
1119
1120
1121 /* For realscripts */
1122 .supsubscript {
1123     display: inline-block;
1124     text-align:left ;
1125 }
1126
1127 .supsubscript sup,
1128 .supsubscript sub {
1129     position: relative;
1130     display: block;
1131     font-size: .5em;
1132     line-height: 1;
1133 }
1134
1135 .supsubscript sup {
1136     top: .5em;
1137 }
1138
```

```
1139 .supsubscript sub {
1140     top: .5em;
1141 }
1142
1143 span.attribution {
1144     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
1145 }
1146
1147 span.citetitle {
1148     margin-left: 1em ; font-size: 80% ; font-style: oblique;
1149 }
1150
1151 span.poemtitle {
1152     font-size: 120% ; font-weight: bold;
1153 }
1154
1155 pre.tabbing {
1156     font-family: "Linux Libertine Mono O", "Lucida Console",
1157                 "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1158                 "Liberation Mono", "FreeMono", "Andale Mono",
1159                 "Nimbus Mono L", "Courier New", monospace;
1160 }
1161
1162 blockquote {
1163     margin-left: 0px ;
1164     margin-right: 0px ;
1165 }
1166
1167 /* quotchap is for the quotchap package */
1168 div.quotchap {
1169     font-style: oblique ;
1170     overflow-x: auto ;
1171     margin-left: 2em ;
1172     margin-right: 2em ;
1173 }
1174
1175 blockquote p, div.quotchap p {
1176     line-height: 1.5;
1177     text-align: left ;
1178     font-size: .85em ;
1179     margin-left: 3em ;
1180     margin-right: 3em ;
1181 }
1182
1183 /* qauthor is for the quotchap package */
1184 div.qauthor {
1185     display: block ;
1186     text-align: right ;
1187     margin-left: auto ;
1188     margin-right: 2em ;
```

```
1189 font-size: 80% ;
1190 font-variant: small-caps;
1191 }
1192
1193 div.qauthor p {
1194     text-align: right ;
1195 }
1196
1197 blockquotation {
1198     margin-left: 0px ;
1199     margin-right: 0px ;
1200 }
1201
1202 blockquotation p {
1203     line-height: 1.5;
1204     text-align: left ;
1205     font-size: .85em ;
1206     margin-left: 3em ;
1207     margin-right: 3em ;
1208 }
1209
1210 div.epigraph, div.dictum {
1211     line-height: 1.2;
1212     text-align: left ;
1213     padding: 3ex 1em 0ex 1em ;
1214     /* margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
1215     margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
1216     /* margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
1217     font-size: .85em ;
1218     max-width: 27em ;
1219 }
1220
1221
1222
1223 div.epigraphsource, div.dictumauthor {
1224     text-align:right ;
1225     margin-left:auto ;
1226     /* max-width: 50% ; */
1227     border-top: 1px solid #A0A0A0 ;
1228     padding-bottom: 3ex ;
1229     line-height: 1.2;
1230 }
1231
1232 div.epigraph p, div.dictum p { padding: .5ex ; margin: 0ex ;}
1233 div.epigraphsource p, div.dictumauthor p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
1234 div.dictumauthor { font-style:italic }
1235
1236
1237 /* lettrine package: */
1238 span.lettrine { font-size: 3ex ; float: left ; }
```



```
1239 span.lettrinetext { font-variant: small-caps ; }
1240
1241 /* ulem and soul packages: */
1242 span.uline {
1243     text-decoration: underline ;
1244     text-decoration-skip ;
1245 }
1246
1247 span.uuline {
1248     text-decoration: underline ;
1249     text-decoration-skip ;
1250     text-decoration-style: double ;
1251 }
1252
1253 span.uwave {
1254     text-decoration: underline ;
1255     text-decoration-skip ;
1256     text-decoration-style: wavy ;
1257 }
1258
1259 span.sout {
1260     text-decoration: line-through ;
1261 }
1262
1263 span.xout {
1264     text-decoration: line-through ;
1265 }
1266
1267 span.dashuline {
1268     text-decoration: underline ;
1269     text-decoration-skip ;
1270     text-decoration-style: dashed ;
1271 }
1272
1273 span.dotuline {
1274     text-decoration: underline ;
1275     text-decoration-skip ;
1276     text-decoration-style: dotted ;
1277 }
1278
1279 span.letterspacing { letter-spacing: .2ex ; }
1280
1281 span.capsspacing {
1282     font-variant: small-caps ;
1283     letter-spacing: .1ex ;
1284 }
1285
1286 span.highlight { background: #F8E800 ; }
1287
1288
```

```
1289
1290
1291html body {
1292  margin: 0 ;
1293  line-height: 1.2;
1294}
1295
1296
1297body div {
1298  margin: 0ex;
1299}
1300
1301
1302h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1303{
1304  font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1305             "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1306             "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1307             "Times New Roman", serif;
1308  font-style: normal ;
1309  font-weight: bold ;
1310  text-align: left ;
1311}
1312
1313h1 { /* title of the entire website, used on each page */
1314  text-align: center ;
1315  font-size: 2.5em ;
1316  padding: .4ex 0em 0ex 0em ;
1317}
1318h2 { font-size: 2.25em }
1319h3 { font-size: 2em }
1320h4 { font-size: 1.75em }
1321h5 { font-size: 1.5em }
1322h6 { font-size: 1.25em }
1323span.paragraph {font-size: 1em ; font-variant: normal ;
1324  margin-right: 1em ; }
1325span.subparagraph {font-size: 1em ; font-variant: normal ;
1326  margin-right: 1em ; }
1327
1328div.minisec {
1329  font-family: "DejaVu Sans", "Bitstream Vera Sans",
1330             Geneva, Verdana, sans-serif ;
1331  font-style: normal ;
1332  font-weight: bold ;
1333  text-align: left ;
1334}
1335
1336/* Title of the file */
1337h1 {
1338  margin: 0ex 0em 0ex 0em ;
```

```
1339 line-height: 1.3;
1340 text-align: center ;
1341 }
1342
1343 /* Part */
1344 h2 {
1345     margin: 1ex 0em 1ex 0em ;
1346     line-height: 1.3;
1347     text-align: center ;
1348 }
1349
1350 /* Chapter */
1351 h3 {
1352     margin: 3ex 0em 1ex 0em ;
1353     line-height: 1.3;
1354 }
1355
1356 /* Section */
1357 h4 {
1358     margin: 3ex 0em 1ex 0em ;
1359     line-height: 1.3;
1360 }
1361
1362 /* Sub-Section */
1363 h5 {
1364     margin: 3ex 0em 1ex 0em ;
1365     line-height: 1.3;
1366 }
1367
1368 /* Sub-Sub-Section */
1369 h6 {
1370     margin: 3ex 0em 1ex 0em ;
1371     line-height: 1.3;
1372 }
1373
1374
1375 div.titlepage {
1376     text-align: center ;
1377 }
1378
1379 .footnotes {
1380     font-size: .85em ;
1381     margin: 3ex 1em 0ex 1em ;
1382     padding-bottom: 1ex ;
1383     border-top: 1px solid silver ;
1384 }
1385
1386 .marginpar, .marginparblock {
1387     max-width: 50%;
1388     float: right;
```

```
1389     text-align:left;
1390     margin: 1ex 0.5em 1ex 1em ;
1391     padding: 1ex 0.5em 1ex 0.5em ;
1392     font-size: 85% ;
1393     border-top: 1px solid silver ;
1394     border-bottom: 1px solid silver ;
1395     overflow-x: auto;
1396 }
1397
1398 .marginpar br { margin-bottom: 2ex ; }
1399
1400 div.marginblock, div.marginparblock {
1401     max-width:50%;
1402     float:right;
1403     text-align:left;
1404     margin: 1ex 0.5em 1ex 1em ;
1405     padding: 1ex 0.5em 1ex 0.5em ;
1406     overflow-x: auto;
1407 }
1408
1409 div.marginblock div.minipage,
1410 div.marginparblock div.minipage {
1411     display: block ;
1412     margin: 0pt auto 0pt auto ;
1413 }
1414
1415 div.marginblock div.minipage p ,
1416 div.marginparblock div.minipage p
1417     { font-size: 85%}
1418
1419 div.marginblock br ,
1420 div.marginparblock br
1421     { margin-bottom: 2ex ; }
1422
1423
1424 section.textbody div.footnotes{
1425     margin: 3ex 0em 0ex 0em ;
1426     border-bottom: 2px solid silver ;
1427 }
1428
1429 .footnoteheader {
1430     border-top: 2px solid silver ;
1431     margin-top: 3ex ;
1432     padding-top: 1ex ;
1433     font-weight: bold ;
1434 }
1435
1436 .mpfootnotes {
1437     text-align: left ;
1438     font-size: .85em ;
```

```
1439     margin-left: 1em ;
1440     border-top: 1px solid silver ;
1441 }
1442
1443 /* Remove footnote top border in the title page. */
1444 div.titlepage div.mpfootnotes {
1445     border-top: none ;
1446 }
1447
1448
1449
1450 ol {
1451     margin: 1ex 1em 1ex 0em;
1452     line-height: 1.2;
1453 }
1454
1455 ul, body dir, body menu {
1456     margin: 3ex 1em 3ex 0em;
1457     line-height: 1.2;
1458 }
1459
1460 li { margin: 0ex 0em 1ex 0em; }
1461
1462 html {
1463     margin: 0;
1464     padding: 0;
1465 }
1466
1467 .programlisting {
1468     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1469                 "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1470                 "Courier New", monospace;
1471     margin: 1ex 0ex 1ex 0ex ;
1472     padding: .5ex 0pt .5ex 0pt ;
1473     overflow-x: auto;
1474 }
1475
1476 section.textbody>pre.programlisting {
1477 border-top: 1px solid silver ;
1478 border-bottom: 1px solid silver ;
1479 }
1480
1481
1482 div.displaymath {
1483     text-align: center ;
1484 }
1485
1486 div.displaymathnumbered {
1487     text-align: right ;
1488     margin-left: 5% ;
```

```
1489     margin-right: 5% ;
1490     min-width: 2.5in ;
1491 }
1492
1493 @media all and (min-width: 400px) {
1494     div.displaymathnumbered {
1495         margin-left: 10% ;
1496         margin-right: 10% ;
1497     }
1498 }
1499
1500 @media all and (min-width: 800px) {
1501     div.displaymathnumbered {
1502         margin-right: 20% ;
1503     }
1504 }
1505
1506 @media all and (min-width: 1200px) {
1507     div.displaymathnumbered {
1508         margin-right: 30% ;
1509     }
1510 }
1511
1512
1513 .inlineprogramlisting {
1514     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1515                 "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1516                 "Courier New", monospace;
1517     overflow-x: auto;
1518 }
1519
1520 span.listinglabel {
1521     display: inline-block ;
1522     font-size: 70% ;
1523     width: 4em ;
1524     text-align: right ;
1525     margin-right: 2em ;
1526 }
1527
1528 div.abstract {
1529     margin: 2em 5% 2em 5% ;
1530     padding: 1ex 1em 1ex 1em ;
1531     /* font-weight: bold ; */
1532     font-size: 90% ;
1533     text-align: left ;
1534 }
1535
1536 div.abstract dl {line-height:1.5;}
1537 div.abstract dt {color:#304070;}
1538
```

```
1539 div.abstracttitle{
1540     font-family: "URW Classico", Optima, "Linux Biolinum O",
1541         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1542         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1543     font-weight:bold;
1544     font-size:1.25em;
1545     text-align: center ;
1546 }
1547
1548 span.abstracrunintitle{
1549     font-family: "URW Classico", Optima, "Linux Biolinum O",
1550         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1551         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1552     font-weight:bold;
1553 }
1554
1555
1556 .verbatim {
1557     overflow-x: auto ;
1558 }
1559
1560 .alltt {
1561     overflow-x: auto ;
1562 }
1563
1564
1565 .bverbatim {
1566     margin: 1ex 0pt 1ex 0pt ;
1567     padding: .5ex 0pt .5ex 0pt ;
1568     overflow-x: auto ;
1569 }
1570
1571 .lverbatim {
1572     margin: 1ex 0pt 1ex 0pt ;
1573     padding: .5ex 0pt .5ex 0pt ;
1574     overflow-x: auto ;
1575 }
1576
1577 .fancyvrb {
1578     font-size:.85em ;
1579     margin: 3ex 0pt 3ex 0pt
1580 }
1581
1582 .fancyvrblabel {
1583     font-weight:bold;
1584     text-align: center ;
1585 }
1586
1587
1588 .verse {
```

```
1589     font-family: "Linux Libertine Mono O", "Lucida Console",
1590                 "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1591                 "Liberation Mono", "FreeMono", "Andale Mono",
1592                 "Nimbus Mono L", "Courier New", monospace;
1593     margin-left: 1em ;
1594 }
1595
1596
1597 div.singlespace { line-height: 1.2 ; }
1598 div.onehalfspace { line-height: 1.5 ; }
1599 div.doublespace { line-height: 2 ; }
1600
1601
1602 /* Word processor format output: */
1603 div.wpfigure { border: 1px solid red ; margin: .5ex ; padding: .5ex ; }
1604 div.wptable { border: 1px solid blue ; margin: .5ex ; padding: .5ex ; }
1605 div.wpminipage { border: 1px solid green ; margin: .5ex ; padding: .5ex ;}
1606
1607
1608
1609
1610 /* Minipage environments, vertically aligned to top, center, bottom: */
1611 .minipage, .fminipage, .fcolorminipage {
1612     /* display: inline-block ; */
1613     /* Mini pages which follow each other will be tiled. */
1614     margin: .25em .25em .25em .25em;
1615     padding: .25em .25em .25em .25em;
1616     display: inline-flex;
1617     flex-direction: column ;
1618     overflow: auto;
1619 }
1620
1621 /* Paragraphs in the flexbox did not collapse their margins. */
1622 /* Have not yet researched this. */
1623 .minipage p {margin: .75ex 0em .75ex 0em ;}
1624
1625 .fbboxBlock .minipage, .colorbox .minipage, .colorboxBlock .minipage,
1626 .fcolorbox .minipage, .fcolorboxBlock .minipage
1627     {border: none ; background: none;}
1628
1629 .fbbox, .fbboxBlock { border: 1px solid black ; }
1630
1631 .fbbox, .fbboxBlock, .fcolorbox, .fcolorboxBlock, .colorbox, .colorboxBlock,
1632 .fminipage, .fcolorminipage
1633     {display: inline-block}
1634
1635 .shadowbox, .shabox {
1636     border: 1px solid black;
1637     box-shadow: 3px 3px 3px #808080 ;
1638     border-radius: 0px ;
```



```
1639     padding: .4ex .3em .4ex .3em ;
1640     margin: 0pt .3ex 0pt .3ex ;
1641     display: inline-block ;
1642 }
1643
1644 .doublebox {
1645     border: 3px double black;
1646     border-radius: 0px ;
1647     padding: .4ex .3em .4ex .3em ;
1648     margin: 0pt .3ex 0pt .3ex ;
1649     display: inline-block ;
1650 }
1651
1652 .ovalbox, .Ovalbox {
1653     border: 1px solid black;
1654     border-radius: 1ex ;
1655     padding: .4ex .3em .4ex .3em ;
1656     margin: 0pt .3ex 0pt .3ex ;
1657     display: inline-block ;
1658 }
1659
1660 .Ovalbox { border-width: 2px ; }
1661
1662 .framebox {
1663     border: 1px solid black;
1664     border-radius: 0px ;
1665     padding: .3ex .2em 0ex .2em ;
1666     margin: 0pt .1ex 0pt .1ex ;
1667     display: inline-block ;
1668 }
1669
1670
1671 .mdframed {
1672 /*     padding: 0ex ; */
1673 /*     border: 1px solid black; */
1674 /*     border-radius: 0px ; */
1675     padding: 0ex ;
1676     margin: 3ex 5% 3ex 5% ;
1677 /*     display: inline-block ; */
1678 }
1679
1680 .mdframed p { padding: 0ex .5em 0ex .5em ; }
1681
1682 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
1683
1684 .mdframedtitle {
1685     padding: .5em ;
1686     display: block ;
1687     font-size: 130% ;
1688     margin-bottom: 1ex ;
```

```
1689 }
1690
1691 .mdframedsubtitle {
1692     padding: 0ex .5em 0ex .5em ;
1693     display: block ;
1694     font-size: 115% ;
1695 }
1696
1697 .mdframedsubsubtitle {
1698     padding: 0ex .5em 0ex .5em ;
1699     display: block ;
1700 }
1701
1702 .mdtheorem {
1703     padding: 0ex .5em 0ex .5em ;
1704     margin: 3ex 5% 3ex 5% ;
1705 /*     display: inline-block ; */
1706 }
1707
1708
1709 /* framed package */
1710 .framed, pre.boxedverbatim, fcolorbox {
1711     margin: 3ex 0em 3ex 0em ;
1712     border: 1px solid black;
1713     border-radius: 0px ;
1714     padding: .3ex 1em 0ex 1em ;
1715     display: block ;
1716 }
1717
1718 .shaded {
1719     margin: 3ex 0em 3ex 0em ;
1720     padding: .3ex 1em .3ex 1em ;
1721     display: block ;
1722 }
1723
1724 .snugframed {
1725     margin: 3ex 0em 3ex 0em ;
1726     border: 1px solid black;
1727     border-radius: 0px ;
1728     display: block ;
1729 }
1730
1731 .framedleftbar {
1732     margin: 3ex 0em 3ex 0em ;
1733     border-left: 3pt solid black;
1734     border-radius: 0px ;
1735     padding: .3ex .2em .3ex 1em ;
1736     display: block ;
1737 }
1738
```

```
1739 .framedtitle {
1740     margin: 0em ;
1741     padding: 0em ;
1742     font-size: 130%
1743 }
1744
1745 .framedtitle p { padding: .3em }
1746
1747
1748
1749 dl {
1750     margin: 1ex 2em 1ex 0em;
1751     line-height: 1.3;
1752 }
1753
1754 dl dt {
1755     margin-top: 1ex;
1756     margin-left: 1em ;
1757     font-weight: bold;
1758 }
1759
1760 dl dd p { margin-top: 0em; }
1761
1762
1763 nav {
1764     font-family: "URW Classico", Optima, "Linux Biolinum O",
1765         "DejaVu Sans", "Bitstream Vera Sans",
1766         Geneva, Verdana, sans-serif ;
1767     margin-bottom: 4ex ;
1768 }
1769
1770 nav p {
1771     line-height: 1.2 ;
1772     margin-top: .5ex ;
1773     margin-bottom: .5ex;
1774     font-size: .9em ;
1775 }
1776
1777
1778
1779 img, img.hyperimage, img.borderimage {
1780     max-width: 600px;
1781     border: 1px solid silver;
1782     box-shadow: 3px 3px 3px #808080 ;
1783     padding: .5% ;
1784     margin: .5% ;
1785     background: none ;
1786 }
1787
1788 img.inlineimage{
```

```
1789     padding: 0px ;
1790     box-shadow: none ;
1791     border: none ;
1792     background: none ;
1793     margin: 0px ;
1794     display: inline-block ;
1795     border-radius: 0px ;
1796 }
1797
1798 img.logoimage{
1799     max-width: 300px ;
1800     box-shadow: 3px 3px 3px #808080 ;
1801     border: 1px solid black ;
1802     background:none ;
1803     padding:0 ;
1804     margin:.5ex ;
1805     border-radius: 10px ;
1806 }
1807
1808
1809 .section {
1810 /*
1811     To have each section float relative to each other:
1812 */
1813 /*
1814     display: block ;
1815     float: left ;
1816     position: relative ;
1817     background: white ;
1818     border: 1px solid silver ;
1819     padding: .5em ;
1820 */
1821     margin: 0ex .5em 0ex .5em ;
1822     padding: 0 ;
1823 }
1824
1825
1826 figure {
1827     margin: 3ex auto 3ex auto ;
1828     padding: 1ex 1em 1ex 1em ;
1829     overflow-x: auto ;
1830 }
1831
1832
1833 /* To automatically center images in figures: */
1834 /*
1835 figure img.inlineimage {
1836     margin: 0ex auto 0ex auto ;
1837     display: block ;
1838 }
```

```
1839 */
1840
1841 /* To automatically center minipages in figures: */
1842 /*
1843 figure div.minipage, figure div.minipage div.minipage {
1844     margin: 1ex auto 1ex auto ;
1845     display: block ;
1846 }
1847 */
1848
1849 figure div.minipage p { font-size: 85% ; }
1850
1851 figure.subfigure, figure.subtable {
1852     display: inline-block ; margin: 3ex 1em 3ex 1em ;
1853 }
1854
1855 figcaption .minipage { margin:0 ; padding: 0 }
1856
1857 div.minipage figure { border: none ; box-shadow: none ; }
1858
1859 div.floatrow { text-align: center; }
1860
1861 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
1862
1863 div.floatfoot { font-size: .85em ;
1864     border-top: 1px solid silver ; line-height: 1.2 ; }
1865
1866 figcaption , .lstlistingtitle {
1867     font-size: .85em ;
1868     text-align: center ;
1869     font-weight: bold ;
1870     margin-top: 1ex ;
1871     margin-bottom: 1ex ;
1872 }
1873
1874 figure.subfigure figcaption, figure.subtable figcaption {
1875     border-bottom: none ; background: none ;
1876 }
1877
1878 div.nonfloatcaption {
1879     margin: 1ex auto 1ex auto ;
1880     font-size: .85em ;
1881     text-align: center ;
1882     font-weight: bold ;
1883 }
1884
1885 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
1886 figure div.floatrow div.minipage figcaption {
1887     border: none ;
1888     background: none ;
```

```
1889 }
1890
1891
1892 table {
1893     margin: 1ex auto 1ex auto ;
1894     border-collapse: separate ;
1895     border-spacing: 0px ;
1896     line-height: 1.3 ;
1897 }
1898
1899 tr.hline td {border-top: 1px solid #808080 ; margin-top: 0ex ;
1900     margin-bottom: 0ex ; } /* for \hline */
1901
1902 tr.tbrule td {border-top: 1px solid black ; margin-top: 0ex ;
1903     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
1904
1905 td {padding: .5ex .5em .5ex .5em ;}
1906
1907 table td.tdl { text-align: left ; vertical-align: middle ; }
1908 table td.tdc { text-align: center ; vertical-align: middle ; }
1909 table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
1910 table td.tdbang { text-align: center ; vertical-align: middle ; }
1911 table td.tdr { text-align: right ; vertical-align: middle ; }
1912 table td.tdp { text-align: left ; vertical-align: bottom ; }
1913 table td.tdm { text-align: left ; vertical-align: middle ; }
1914 table td.tdb { text-align: left ; vertical-align: top ; }
1915 table td.tdP { text-align: center ; vertical-align: bottom ; }
1916 table td.tdM { text-align: center ; vertical-align: middle ; }
1917 table td.tdB { text-align: center ; vertical-align: top ; }
1918
1919 table td.tvertbarl { border-left: 1px solid black }
1920 table td.tvertbarr { border-right: 1px solid black }
1921
1922
1923 /* for cmidrules: */
1924 table td.tdrule {
1925     border-top: 1px solid #A0A0A0 ;
1926 }
1927
1928 table td.tdrulel {
1929     border-top-left-radius:.5em ;
1930     border-top: 1px solid #A0A0A0 ;
1931 }
1932
1933 table td.tdruler {
1934     border-top-right-radius:.5em ;
1935     border-top: 1px solid #A0A0A0 ;
1936 }
1937
1938 table td.tdrulelr {
```

```
1939     border-top-left-radius:.5em ;
1940     border-top-right-radius:.5em ;
1941     border-top: 1px solid #A0A0A0 ;
1942 }
1943
1944
1945 /* Margins of paragraphs inside table cells: */
1946 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
1947     padding-bottom: 1ex ; margin: 0ex ; }
1948 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
1949     padding-bottom: 1ex ; margin: 0ex ; }
1950 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
1951     padding-bottom: 1ex ; margin: 0ex ; }
1952
1953 td.tdp , td.tdprule , td.tdP , td.tdPrule
1954     { padding: 0ex .5em 0ex .5em ; }
1955 td.tdm , td.tdmrule , td.tdM , td.tdMrule
1956     { padding: 0ex .5em 0ex .5em ; }
1957 td.tdb , td.tdbrule , td.tdB , td.tdBrule
1958     { padding: 0ex .5em 0ex .5em ; }
1959
1960
1961 /* table notes: */
1962 .tnotes {
1963     margin: 0ex 5% 1ex 5% ;
1964     padding: 0.5ex 1em 0.5ex 1em;
1965     font-size:.85em;
1966     text-align: left ;
1967 }
1968
1969 .tnotes dl dt p {margin-bottom:0px;}
1970
1971 .tnoteitemheader {margin-right: 1em;}
1972
1973
1974 /* for colortbl and cell color */
1975 div.cellcolor {
1976     width: 100% ;
1977     padding: .5ex .5em .5ex .5em ;
1978     margin: -.5ex -.5em -.5ex -.5em ;
1979 }
1980
1981
1982 /* for bigdelim */
1983 .ldelim, .rdelim { font-size: 200% }
1984
1985
1986 /* center, flushleft, flushright environments */
1987 div.center{text-align:center;}
1988 div.center table {margin-left:auto;margin-right:auto;}
```

```
1989 div.flushleft{text-align:left;}
1990 div.flushleft table {margin-left:0em ; margin-right:auto;}
1991 div.flushright{text-align:right;}
1992 div.flushright table {margin-left:auto ; margin-right: 0em ;}
1993
1994
1995 /* Fancybox */
1996 div.Btrivlist table tr td { padding: .2ex 0em ; }
1997
1998
1999 /* program listing callouts: */
2000 span.callout {
2001     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2002     Geneva, Verdana, sans-serif ;
2003     border-radius: .5em;
2004     background-color:black;
2005     color:white;
2006     padding:0px .25em 0px .25em;
2007     margin: 0 ;
2008     font-weight: bold;
2009     font-size:.72em ;
2010 }
2011
2012 div.programlisting pre.verbatim span.callout{
2013     font-size: .85em ;
2014 }
2015
2016
2017
2018
2019
2020 div.published
2021 {
2022     text-align: center ;
2023     font-variant: normal ;
2024     font-style: italic ;
2025     font-size: 1em ;
2026     margin: 3ex 0em 3ex 0em ;
2027 }
2028
2029 div.subtitle
2030 {
2031     text-align: center ;
2032     font-variant: normal ;
2033     font-style: italic ;
2034     font-size: 1.25em ;
2035     margin: 3ex 0em 3ex 0em ;
2036 }
2037
2038 div.subtitle p { margin: 1ex ; }
```



```
2039
2040 div.author
2041 {
2042     font-variant: normal ;
2043     font-style: normal ;
2044     font-size: 1em ;
2045     margin: 3ex 0em 3ex 0em ;
2046 }
2047
2048 div.oneauthor {
2049     display: inline-block ;
2050     margin: 3ex 1em 0ex 1em ;
2051 }
2052
2053 /*
2054 div.author table {
2055     margin: 3ex auto 0ex auto ;
2056     background: none ;
2057 }
2058
2059 div.author table tbody tr td { padding: .25ex ; }
2060 */
2061
2062 span.affiliation {font-size: .85em ; font-variant: small-caps; }
2063
2064 div.titledate {
2065     text-align: center ;
2066     font-size: .85em ;
2067     font-style: italic;
2068     margin: 6ex 0em 6ex 0em ;
2069 }
2070
2071
2072 nav.topnavigation{
2073     text-align: left ;
2074     padding: 0.5ex 1em 0.5ex 1em ;
2075 /*     margin: 2ex 0em 3ex 0em ; */
2076     margin: 0 ;
2077     border-bottom: 1px solid silver ;
2078     border-top: 1px solid silver ;
2079     clear:right ;
2080 }
2081
2082 nav.botnavigation{
2083     text-align: left ;
2084     padding: 0.5ex 1em 0.5ex 1em ;
2085 /*     margin: 3ex 0em 2ex 0em ; */
2086     margin: 0 ;
2087     border-top: 1px solid silver ;
2088     border-bottom: 1px solid silver ;
```

```
2089     clear:right ;
2090 }
2091
2092
2093 header{
2094     line-height: 1.2 ;
2095     font-size: 1em ;
2096     /*     border-bottom: 2px solid silver ; */
2097     margin: 0px ;
2098     padding: 0ex 1em 0ex 1em ;
2099     text-align:center ;
2100 }
2101
2102 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
2103
2104
2105 footer{
2106     font-size: .85em ;
2107     line-height: 1.2 ;
2108     margin-top: 1ex ;
2109     border-top: 2px solid silver ;
2110     padding: 2ex 1em 2ex 1em ;
2111     clear:right ;
2112     text-align:left ;
2113 }
2114
2115
2116 a.linkhome { font-weight:bold ; font-size: 1em ;}
2117
2118
2119 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
2120
2121 img.lateximage{
2122     padding: 0px 0px 0px 0px ;
2123     box-shadow: none ;
2124     border: none ;
2125     background: none ;
2126     margin: 0px 0px -.15ex 0px ;
2127     /* pdfcrop leaves a slight margin, adjust to baseline */
2128     max-width: 100% ;
2129     border-radius: 0ex ;
2130     border: none ;
2131 }
2132
2133
2134
2135 nav.sidetoc {
2136     font-family: "DejaVu Serif", "Bitstream Vera Serif",
2137         "Lucida Bright", Georgia, serif;
2138     float:right ;
```

```
2139     width: 20%;
2140     border-left: 1px solid silver;
2141     border-top: 1px solid silver;
2142     border-bottom: 1px solid silver;
2143 /*     border-top: 2px solid #808080 ; */
2144     background: #FAF7F4 ;
2145     padding: 2ex 0em 2ex 1em ;
2146     margin: 0ex 0em 2ex 1em ;
2147     font-size:.9em ;
2148     border-radius: 20px 0px 0px 20px ;
2149 }
2150
2151 div.sidetoccontents {
2152 /*     border-top: 1px solid silver ; */
2153     overflow-y: auto ;
2154     width: 100% ;
2155     text-align: left ;
2156 }
2157
2158
2159 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
2160     text-indent: 0 ; }
2161
2162 nav.sidetoc p a {color:black ; font-size: .7em ;}
2163
2164 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
2165     border-bottom: 1px solid silver ; }
2166
2167 nav.sidetoc a:hover {text-decoration: underline ; }
2168
2169
2170
2171 section.textbody { margin: 0ex 1em 0ex 1em ;}
2172
2173
2174 div.multicolshheading { -webkit-column-span: all;
2175     -moz-column-span: all; column-span: all; }
2176 div.multicols { -webkit-columns: 3 380px ;
2177     -moz-columns: 3 380px ; columns: 3 380px ; }
2178 div.multicols p {margin-top: 0ex}
2179
2180
2181
2182 /* Used to support algorithmicx: */
2183 span.floatright { float: right ; }
2184
2185
2186
2187
2188 /* Native LaTeX theorems: */
```

```
2189
2190 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
2191 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
2192
2193
2194 /* theorem, amsthm, and ntheorem packages */
2195
2196 span.theoremheader,
2197 span.theoremheaderplain,
2198 span.theoremheaderdefinition,
2199 span.theoremheaderbreak,
2200 span.theoremheadermarginbreak,
2201 span.theoremheaderchangebreak,
2202 span.theoremheaderchange,
2203 span.theoremheadermargin
2204 {
2205     font-style:normal ; font-weight: bold ; margin-right: 1em ;
2206 }
2207
2208 span.amsthmnameplain,
2209 span.amsthmnamedefinition,
2210 span.amsthmnumberplain,
2211 span.amsthmnumberdefinition
2212 {
2213     font-style:normal ; font-weight: bold ;
2214 }
2215
2216
2217 span.amsthmnameremark,
2218 span.amsthmnumberremark
2219 {font-style:italic ; font-weight: normal ; }
2220
2221
2222 span.amsthmnoteplain,
2223 span.amsthmnotedefinition
2224 {font-style:normal ;}
2225
2226
2227 span.theoremheaderremark,
2228 span.theoremheaderproof,
2229 span.amsthmproofname
2230 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
2231
2232 span.theoremheadersc
2233 {
2234     font-style:normal ;
2235     font-variant: small-caps ;
2236     font-weight: normal ;
2237     margin-right: 1em ;
2238 }
```

```
2239
2240 .theoremdemark {float:right}
2241
2242 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
2243 div.theorembodybreak, div.theorembodynonumberbreak,
2244 div.theorembodymarginbreak,
2245 div.theorembodychangebreak,
2246 div.theorembodychange,
2247 div.theorembodymargin
2248 {
2249     font-style:italic;
2250     margin-top: 3ex ; margin-bottom: 3ex ;
2251 }
2252
2253 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
2254 div.theorembodyplainupright, nonumberplainuprightsc,
2255 div.amsthmbodydefinition, div.amsthmbodyremark,
2256 div.amsthmproof
2257 {
2258     font-style: normal ;
2259     margin-top: 3ex ; margin-bottom: 3ex ;
2260 }
2261
2262 span.amsthmnoteremark {}
2263
2264
2265
2266 /*
2267 For CSS LaTeX and related logos:
2268 Based on:
2269 http://edward.oconnor.cx/2007/08/tex-poshlet
2270 http://nitens.org/taraborelli/texlogo
2271 */
2272
2273 .latexlogofont {
2274     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2275                 "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2276     font-variant: normal ;
2277 }
2278
2279 .latexlogo {
2280     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2281                 "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2282     letter-spacing: .03em ;
2283     font-size: 1.1em;
2284 }
2285
2286 .latexlogo sup {
2287     text-transform: uppercase;
2288     letter-spacing: .03em ;
```

```
2289 font-size: 0.85em;
2290 vertical-align: 0.15em;
2291 margin-left: -0.36em;
2292 margin-right: -0.15em;
2293 }
2294
2295 .latexlogo sub {
2296   text-transform: uppercase;
2297   vertical-align: -0.5ex;
2298   margin-left: -0.1667em;
2299   margin-right: -0.125em;
2300   font-size: 1em;
2301 }
2302
2303 .xetexlogo {
2304   font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2305               "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2306   letter-spacing: .03em ;
2307   font-size: 1.1em;
2308 }
2309
2310 /* A smaller gap between Xe and Tex v.s. LaTeX: */
2311 .xetexlogo sub {
2312   text-transform: uppercase;
2313   vertical-align: -0.5ex;
2314   margin-left: -0.0667em;
2315   margin-right: -0.2em;
2316   font-size: 1em;
2317   letter-spacing: .03em ;
2318 }
2319
2320 /* A large gap between Xe and LaTeX v.s. TeX: */
2321 .xelatexlogo sub {
2322   text-transform: uppercase;
2323   vertical-align: -0.5ex;
2324   margin-left: -0.0667em;
2325   margin-right: -.05em;
2326   font-size: 1em;
2327   letter-spacing: .03em ;
2328 }
2329
2330 .amslogo {
2331   font-family: "TeXGyreChorus", "URW Chancery L",
2332               "Apple Chancery", "ITC Zapf Chancery", "Monotype Corsiva",
2333               "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
2334               "Hoefler Text", Times, "Times New Roman", serif;
2335   font-style: italic;
2336 }
2337
2338 .lyxlogo {
```

```
2339     font-family: "URW Classico", Optima, "Linux Biolinum O",
2340         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
2341         Verdana, sans-serif ;
2342 }
2343
2344
2345
2346
2347 /* Only display top and bottom navigation if a small screen: */
2348 /* Hide the sidetoc if a small screen: */
2349 nav.topnavigation { display:none; }
2350 nav.botnavigation { display:none; }
2351
2352 @media screen and (max-width: 45em) {
2353 /*     nav.sidetoc {display:none;} */
2354     nav.sidetoc {
2355         float: none ;
2356         width: 100% ;
2357         margin: 5ex 0px 5ex 0px ;
2358         padding: 0 ;
2359         border-radius: 0 ;
2360         border-bottom: 1px solid black ;
2361         border-top: 1px solid black ;
2362         box-shadow: none ;
2363     }
2364 /*     nav.topnavigation { display:block } */
2365     nav.botnavigation { display:block }
2366     .marginpar {
2367         max-width: 100%;
2368         float: none;
2369         display:block ;
2370         margin: 1ex 1em 1ex 1em ;
2371     }
2372 }
2373
2374 @media print {
2375     body {
2376         font-family: "Linux Libertine O",
2377             "DejaVu Serif", "Bitstream Vera Serif",
2378             "Liberation Serif", "Nimbus Roman No 9 L",
2379             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2380     }
2381     nav.sidetoc { display:none; }
2382     nav.topnavigation { display: none; }
2383     nav.botnavigation { display: none; }
2384 }
2385
2386 @media handheld {
2387     nav.sidetoc { display:none; }
2388     nav.topnavigation { display:block }
```

```

2389     nav.botnavigation { display:block }
2390 }
2391
2392 @media projection {
2393     nav.sidetoc { display:none; }
2394     nav.topnavigation { display:block }
2395     nav.botnavigation { display:block }
2396 }
2397 \end{filecontents*}
2398 % \end{Verbatim}% for syntax highlighting
2399 \end{warpprint}

```

### 35.5 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.

```

2400 \begin{warpprint}
2401 \begin{filecontents*}{lwarp_sagebrush.css}
2402 @import url("lwarp.css") ;
2403
2404
2405 A:link {color:#105030 ; text-decoration: none ; }
2406 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
2407 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
2408 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
2409
2410
2411
2412 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2413 {
2414     font-family: "URW Classico", Optima, "Linux Biolinum O",
2415         "Linux Libertine O", "Liberation Serif",
2416         "Nimbus Roman No 9 L", "FreeSerif",
2417         "Hoefler Text", Times, "Times New Roman", serif;
2418     font-variant: small-caps ;
2419     font-weight: normal ;
2420     color: #304070 ;
2421     text-shadow: 2px 2px 3px #808080;
2422 }
2423
2424 h1 { /* title of the entire website, used on each page */
2425     font-variant: small-caps ;
2426     color: #304070 ;
2427     text-shadow: 2px 2px 3px #808080;

```



```
2428     background-color: #F7F7F0 ;
2429     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
2430 }
2431
2432 h1 {
2433     border-bottom: 1px solid #304070;
2434     border-top: 2px solid #304070;
2435 }
2436
2437 h2 {
2438     border-bottom: 1px solid #304070;
2439     border-top: 2px solid #304070;
2440     background-color: #F7F7F0 ;
2441     background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
2442 }
2443
2444
2445
2446 div.abstract {
2447     background: #f5f5eb ;
2448     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2449
2450     border: 1px solid silver;
2451     border-radius: 1em ;
2452 }
2453
2454 div.abstract dl {line-height:1.5;}
2455 div.abstract dt {color:#304070;}
2456
2457 div.abstracttitle{
2458     font-family: "URW Classico", Optima, "Linux Biolinum O",
2459         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2460         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2461     font-weight:bold;
2462     font-variant: small-caps ;
2463     font-size:1.5em;
2464     border-bottom: 1px solid silver ;
2465     color: #304070 ;
2466     text-align: center ;
2467     text-shadow: 1px 1px 2px #808080;
2468 }
2469
2470 span.abstracrunintitle{
2471     font-family: "URW Classico", Optima, "Linux Biolinum O",
2472         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2473         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2474     font-weight:bold;
2475 }
2476
2477
```

```
2478 div.epigraph, div.dictum {
2479     background: #f5f5eb ;
2480     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2481
2482     border: 1px solid silver ;
2483     border-radius: 1ex ;
2484     box-shadow: 3px 3px 3px #808080 ;
2485 }
2486
2487
2488 .example {
2489     background-color: #f5f5eb ;
2490     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2491
2492 }
2493
2494 div.exampletitle{
2495     font-family: "URW Classico", Optima, "Linux Biolinum O",
2496         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2497         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2498     font-weight:bold;
2499     font-variant: small-caps ;
2500     border-bottom: 1px solid silver ;
2501     color: #304070 ;
2502     text-align: center ;
2503     text-shadow: 1px 1px 2px #808080;
2504 }
2505
2506
2507 .sidebar {
2508     background-color: #f5f5eb ;
2509     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2510
2511 }
2512
2513 div.sidebartitle{
2514     font-family: "URW Classico", Optima, "Linux Biolinum O",
2515         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2516         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2517     font-weight:bold;
2518     font-variant: small-caps ;
2519     border-bottom: 1px solid silver ;
2520     color: #304070 ;
2521     text-align: center ;
2522     text-shadow: 1px 1px 2px #808080;
2523 }
2524
2525
2526 .fancyvrblabel {
2527     font-family: "URW Classico", Optima, "Linux Biolinum O",
```

```
2528         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2529         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2530     font-weight:bold;
2531     font-variant: small-caps ;
2532     font-size: 1.5em ;
2533     color: #304070 ;
2534     text-align: center ;
2535     text-shadow: 1px 1px 2px #808080;
2536 }
2537
2538 div.minipage {
2539     background-color: #eeeeee7 ;
2540     border: 1px solid silver ;
2541     border-radius: 1ex ;
2542 }
2543
2544 table div.minipage { background: none ; border: none ; }
2545
2546 div.framebox div.minipage {border:none ; background:none}
2547
2548 section.textbody > div.minipage {
2549     box-shadow: 3px 3px 3px #808080 ;
2550 }
2551
2552 div.fboxBlock div.minipage { box-shadow: none ; }
2553
2554 .framed .minipage , .framedleftbar .minipage {
2555     border: none ;
2556     background: none ;
2557     padding: 0ex ;
2558     margin: 0ex ;
2559 }
2560
2561 figure.figure .minipage, figcaption .minipage { border: none; }
2562
2563 div.marginblock div.minipage ,
2564 div.marginparblock div.minipage
2565     { border: none; }
2566
2567 figure , div.marginblock {
2568     background-color: #eeeeee7 ;
2569     border: 1px solid silver ;
2570     border-radius: 1ex ;
2571     box-shadow: 3px 3px 3px #808080 ;
2572 }
2573
2574 figure figure {
2575     border: 1px solid silver ;
2576     margin: 0em ;
2577     box-shadow: none ;
```

```
2578 }
2579
2580 /*
2581 figcaption {
2582     border-top: 1px solid silver ;
2583     border-bottom: 1px solid silver ;
2584     background-color: #e8e8e8 ;
2585 }
2586 */
2587
2588
2589 div.table {
2590     box-shadow: 3px 3px 3px #808080 ;
2591 }
2592
2593 /*
2594 .tnotes {
2595     background: #e8e8e8;
2596     border: 1px solid silver;
2597 }
2598 */
2599
2600
2601 nav.topnavigation{
2602     background-color: #b0b8b0 ;
2603     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
2604 }
2605
2606 nav.botnavigation{
2607     background-color: #b0b8b0 ;
2608     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
2609 }
2610
2611
2612
2613 header{
2614     background-color: #F7F7F0 ;
2615     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
2616 }
2617
2618 footer{
2619     background-color: #F7F7F0 ;
2620     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
2621 }
2622
2623
2624
2625 nav.sidetoc {
2626     background-color: #F7F7F0 ;
2627     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
```

```

2628     box-shadow: 3px 3px 3px #808080 ;
2629     border-radius: 0px 0px 0px 20px ;
2630 }
2631
2632 div.sidetoctitle {color: #304070 ; }
2633
2634 nav.sidetoc a:hover {
2635     color:#006000 ;
2636     text-decoration: none ;
2637     text-shadow:0px 0px 2px #a0a0a0;
2638 }
2639
2640
2641 @media screen and (max-width: 45em) {
2642     nav.sidetoc { border-radius: 0 ; }
2643 }
2644
2645
2646 \end{filecontents*}
2647 % \end{Verbatim}% for syntax highlighting
2648 \end{warpprint}

```

## 35.6 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.

```

2649 \begin{warpprint}
2650 \begin{filecontents*}{lwarp_formal.css}
2651 @import url("lwarp.css") ;
2652
2653
2654
2655 A:link {color:#802020 ; text-decoration:none; }
2656 A:visited {color:#802020 ; text-shadow:none ;}
2657 A:hover {color:#400000 ; text-shadow:none ;}
2658 A:active {color:#C00000 ; text-shadow:none ;}
2659
2660
2661 body {
2662     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2663         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2664         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2665         "Times New Roman", serif;
2666     background: #fffcf5;

```

```
2667 }
2668
2669 span.textrm {
2670     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2671         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2672         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2673         "Times New Roman", serif;
2674 }
2675
2676 span.textsf {
2677     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2678         Geneva, Verdana, sans-serif ;
2679 }
2680
2681
2682
2683 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2684 {
2685     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2686         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2687         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2688         "Times New Roman", serif;
2689     color: #800000 ;
2690     text-shadow: none ;
2691 }
2692
2693 h1, h2 {
2694     background-color: #fffcf5 ;
2695     background-image: none ;
2696     border-bottom: 1px solid #808080;
2697     border-top: 2px solid #808080;
2698 }
2699
2700 div.abstracttitle {
2701     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2702         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2703         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2704         "Times New Roman", serif;
2705     color: black ;
2706     text-shadow: none ;
2707 }
2708
2709 span.abstracrunintitle {
2710     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2711         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2712         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2713         "Times New Roman", serif;
2714     color: black ;
2715     text-shadow: none ;
2716 }
```

```
2717
2718 div.abstract { font-size: 100% }
2719
2720 .sidebar {
2721     background: #fffcf5;
2722     background-image: none ;
2723     margin: 2em 5% 2em 5%;
2724     padding: 0.5em 1em;
2725     border: none ;
2726     border-top : 1px solid silver;
2727     border-bottom : 1px solid silver;
2728     font-size: 90% ;
2729 }
2730
2731 div.sidebartitle{
2732     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2733         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2734         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2735         "Times New Roman", serif;
2736     color: #800000 ;
2737     text-shadow: none ;
2738     border: none ;
2739 }
2740
2741 .example {
2742     background: #fffcf5;
2743     background-image: none ;
2744     margin: 2em 5% 2em 5%;
2745     padding: 0.5em 1em;
2746     border: none ;
2747     border-top : 1px solid silver;
2748     border-bottom : 1px solid silver;
2749 }
2750
2751 div.exampletitle{
2752     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2753         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2754         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2755         "Times New Roman", serif;
2756     color: #800000 ;
2757     text-shadow: none ;
2758     border: none ;
2759 }
2760
2761 div.fancyvrblabel{
2762     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2763         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2764         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2765         "Times New Roman", serif;
2766     color: #800000 ;
```

```
2767     text-shadow: none ;
2768     border: none ;
2769 }
2770
2771
2772
2773 .verse {
2774     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2775         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2776         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2777         "Times New Roman", serif;
2778 }
2779
2780
2781 figure {
2782     margin: 3ex 5% 3ex 5% ;
2783     padding: 1ex 1em 1ex 1em ;
2784     background-color: #fffcf5 ;
2785     overflow-x: auto ;
2786     border: none ;
2787 /*     border-top: 1px solid silver; */
2788 /*     border-bottom: 1px solid silver; */
2789 }
2790
2791
2792 figcaption , .lstlisting {
2793     border: none ;
2794 /*     border-top: 1px solid silver ; */
2795 /*     border-bottom: 1px solid silver ; */
2796     background-color: #fffcf5 ;
2797 }
2798
2799 .tnotes {
2800     background: #fffcf5 ;
2801 }
2802
2803 .theorem {
2804     background: none ;
2805 }
2806
2807 .minipage {
2808     background-color: #fffcf5 ;
2809     border: none ;
2810 }
2811
2812 div.floatrow figure { border: none ; }
2813
2814 figure figure { border: none ; }
2815
2816
```



```

2817 nav.toc, nav.lof, nav.lot, nav.lol {
2818     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2819         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2820         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2821         "Times New Roman", serif;
2822 }
2823
2824 nav.sidetoc {
2825     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2826         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2827         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2828         "Times New Roman", serif;
2829     background-image: linear-gradient(to bottom, #fffcf5, #C0C0C0);
2830     border-radius: 0px 0px 0px 20px ;
2831 }
2832
2833 div.sidetoc{
2834     color: #800000 ;
2835 }
2836
2837 header{
2838     background-color: #e0e0e0 ;
2839     background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
2840     text-align:center ;
2841 }
2842
2843 footer{
2844     background-color: #e0e0e0 ;
2845     background-image: linear-gradient(to bottom, #fffcf5, #b0b0b0);
2846     padding: 2ex 1em 2ex 1em ;
2847     clear:right ;
2848     text-align:left ;
2849 }
2850
2851 nav.botnavigation {
2852     background: #dedcd5 ;
2853     border-top: 1px solid black ;
2854 }
2855 \end{filecontents*}
2856 % \end{Verbatim}% for syntax highlighting
2857 \end{warpprint}

```

## 35.7 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.

```

2858 \begin{warpprint}
2859 \begin{filecontents*}{sample_project.css}
2860 /* ( --- Start of project.css --- ) */
2861 /* ( --- A sample project-specific CSS file for lwarp --- ) */
2862
2863 /* Load default lwarp settings: */
2864 @import url("lwarp.css") ;
2865 /* or lwarp_formal.css, lwarp_sagebrush.css */
2866
2867 /* Project-specific CSS setting follow here. */
2868 /* . . . */
2869
2870 /* ( --- End of project.css --- ) */
2871 \end{filecontents*}
2872 % \end{Verbatim}% for syntax highlighting
2873 \end{warpprint}

```

## 35.8 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.

```

2874 \begin{warpprint}
2875 \begin{filecontents*}{lwarp.xdy}
2876 (require "tex/inputenc/latin.xdy")
2877 (merge-rule "\\PS *" "Postscript")
2878 (require "texindy.xdy")
2879 (require "page-ranges.xdy")
2880 (require "book-order.xdy")
2881 (require "page-ranges.xdy")
2882 (markup-locref :open "\hyperindexref{" :close "}")
2883 (define-location-class "arabic-page-numbers"
2884   ("arabic-numbers") :min-range-length 1)
2885 (define-location-class-order ("roman-page-numbers"
2886   "arabic-page-numbers"
2887   "alpha-page-numbers"
2888   "Roman-page-numbers"
2889   "Alpha-page-numbers"
2890   "see"
2891   "seealso"))
2892 \end{filecontents*}
2893 % \end{Verbatim}% for syntax highlighting

```

```
2894 \end{warpprint}
```

### 35.9 lwarp\_one\_limage.cmd

File `lwarp_one_limage.cmd` Used by lwarp to help make lateximages when using WINDOWS.

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

The arguments are each of the three fields from `lateximages.txt`, and also the base name of the source file.

MikTeX does not allow file `lwarp_one_limage.cmd` to be created directly by `lwarpmk`, so `lwarp_one_limage.txt` is created instead, then copied to `lwarp_one_limage.cmd` by `lwarpmk`. This occurs each time `lwarpmk` used to create `lateximages`.

```
2895 \begin{warpprint}
2896 \begin{filecontents*}{lwarp_one_limage.txt}
2897 @echo off
2898 pdfseparate -f %1 -l %1 %4_html.pdf lateximages\lateximagetemp-%d.pdf"
2899 pdfcrop lateximages\lateximagetemp-%1.pdf lateximages\%3.pdf
2900 pdftocairo -svg lateximages\%3.pdf lateximages\%3.svg
2901 del lateximages\%3.pdf
2902 del lateximages\lateximagetemp-%1.pdf"
2903 exit
2904 \end{filecontents*}
2905 \end{warpprint}
```

### 35.10 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.

```
2906 \begin{warpprint}
2907 \begin{filecontents*}{lwarp_mathjax.txt}
2908 <!-- https://groups.google.com/forum/#!topic/
2909 mathjax-users/jUtewUcE2bY -->
2910 <script type="text/x-mathjax-config">
2911 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
2912     var seteqsectionDefault = {name: "", num: 0};
2913     var seteqsections = {}, seteqsection = seteqsectionDefault;
2914     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
```

```

2915     var AMS = MathJax.Extension["TeX/AMSMath"];
2916     TEX.Definitions.Add({
2917       macros: {
2918         seteqsection: "mySection",
2919         seteqnumber: "mySetEqNumber"
2920       }
2921     });
2922
2923     PARSE.Augment({
2924       mySection: function (name) {
2925         seteqsection.num = AMS.number;
2926         var n = this.GetArgument(name);
2927         if (n === "") {
2928           seteqsection = seteqsectionDefault;
2929         } else {
2930           if (!seteqsections["_"+n])
2931             seteqsections["_"+n] = {name:n, num:0};
2932           seteqsection = seteqsections["_"+n];
2933         }
2934         AMS.number = seteqsection.num;
2935       },
2936       mySetEqNumber: function (name) {
2937         var n = this.GetArgument(name);
2938         if (!n || !n.match(/^ *[0-9]+ *$/))
2939           n = ""; else n = parseInt(n)-1;
2940         <!-- $ syntax highlighting -->
2941         if (n === "" || n < 1)
2942           TEX.Error
2943             ("Argument to "+name+" should be a positive integer");
2944         AMS.number = n;
2945       }
2946     });
2947     MathJax.Hub.Config({
2948       TeX: {
2949         equationNumbers: {
2950           formatTag: function (n)
2951             {return "("+(seteqsection.name+"."+n).replace(/\^\.\/,"")+")"},
2952           formatID: function (n) {
2953             n = (seteqsection.name+'.'+n).replace
2954               (/[<'>&]/g,"").replace(/\^\.\/,"");
2955             return 'mjax-eqn-' + n;
2956           }
2957         }
2958       }
2959     });
2960 });
2961 </script>
2962
2963 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
2964 <script type="text/x-mathjax-config">

```

---

```

2965   MathJax.Ajax.config.path["Contrib"] =
2966     "https://cdn.mathjax.org/mathjax/contrib";
2967 </script>
2968
2969 <!-- https://github.com/burnpanck/MathJax-siunitx -->
2970
2971 <script type="text/x-mathjax-config">
2972   MathJax.Hub.Config({
2973     extensions: ["tex2jax.js","[siunitx]/siunitx.js"],
2974     jax: ["input/TeX","output/HTML-CSS"],
2975     tex2jax: {
2976       inlineMath: [["$","$"],["\\(","\\)"]],
2977       processClass: "tabbing|verse"
2978     },
2979     TeX: {extensions: ["AMSMath.js","AMSsymbols.js", "sinuitx.js"]}
2980   });
2981   MathJax.Ajax.config.path['siunitx'] = 'http://rawgit.com/burnpanck/MathJax-siunitx/master/';
2982 </script>
2983
2984 <script type="text/x-mathjax-config">
2985   MathJax.Hub.Config({
2986     TeX: {
2987       equationNumbers: {
2988         autoNumber: "AMS"
2989       }
2990     }
2991   });
2992 </script>
2993
2994 <!-- Alternative CDN provider: -->
2995 <script type="text/javascript" async
2996 src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?config=TeX-AMS_HTML-full">
2997 </script>
2998
2999 <!-- No longer supported after April 30, 2017: -->
3000 <!--
3001 <script
3002   src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
3003 </script>
3004 -->
3005
3006 \end{filecontents*}
3007 % \end{Verbatim}% for syntax highlighting
3008 \end{warpprint}

```

### 35.11 lwarpmk option

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

```
3009 \begin{LWR@createlwarpmk}
```

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

```
3010 \begin{filecontents*}{lwarpmk.lua}
3011 #!/usr/bin/env texlua
3012
3013 -- Copyright 2016-2018 Brian Dunn
3014
3015 -- Print the usage of the lwarpmk command:
3016
3017 printversion = "v0.50"
3018
3019 function printhelp ()
3020 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
3021 end
3022
3023 function printusage ()
3024 print ( [[
3025
3026 lwarpmk print [project]: Compile the print version if necessary.
3027 lwarpmk print1 [project]: Forced single compile of the print version.
3028 lwarpmk printindex [project]: Process the index for the print version.
3029 lwarpmk printglossary [project]: Process the glossary for the print version.
3030 lwarpmk html [project]: Compile the HTML version if necessary.
3031 lwarpmk html1 [project]: Forced single compile of the HTML version.
3032 lwarpmk htmlindex [project]: Process the index for the html version.
3033 lwarpmk htmlglossary [project]: Process the glossary for the html version.
3034 lwarpmk again [project]: Touch the source code to trigger recompiles.
3035 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
3036 lwarpmk pdftohtml [project]:
3037     For use with latexmk or a Makefile:
3038     Convert project_html.pdf to project_html.html and
3039     individual HTML files.
3040 lwarpmk clean [project]: Remove .aux, .toc, .lof/t, .idx, .ind, .log, *_html_inc.*, .gl*
3041 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
3042 lwarpmk -h: Print this help message.
3043 lwarpmk --help: Print this help message.
3044
3045 ]] )
3046 printconf ()
3047 end
3048
3049 -- Print the format of the configuration file lwarpmk.conf:
3050
```

```

3051 function printconf ( )
3052 print ( [[
3053 An example lwarpmk.conf or <project>.lwarpmkconf project file:
3054 --
3055 opsystem = "Unix"      (or "Windows")
3056 latexname = "pdflatex" (or "lualatex", or "xelatex")
3057 sourcename = "projectname" (the source-code filename w/o .tex)
3058 homehtmlfilename = "index" (or perhaps the project name)
3059 htmlfilename = ""      (or "projectname" - filename prefix)
3060 latexmk = "false"      (or "true" to use latexmk to build PDFs)
3061 language = "english"   (use a language supported by xindy)
3062 xdyfile = "lwarp.xdy"  (or a custom file based on lwarp.xdy)
3063 --
3064 Filenames must contain only letters, numbers, underscore, or dash.
3065 Values must be in "quotes".
3066
3067 ]] ) ;
3068 end
3069
3070
3071 -- Split one large sourcefile into a number of files,
3072 -- starting with destfile.
3073 -- The file is split at each occurrence of <!--|Start file|newfilename|*
3074
3075 function splitfile (destfile,sourcefile)
3076 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
3077 local sfile = io.open(sourcefile)
3078 io.output(destfile)
3079 for line in sfile:lines() do
3080 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
3081 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then
3082     -- split the file
3083     io.output(newfilename) ;
3084 else
3085     -- not a splitpoint
3086     io.write (line .. "\n") ;
3087 end
3088 end -- do
3089 io.close(sfile)
3090 end -- function
3091
3092 -- Incorrect value, so print an error and exit.
3093
3094 function cvalueerror ( line, linenum , cvalue )
3095     print ("lwarpmk: ===")
3096     print ("lwarpmk: " .. linenum .. " : " .. line ) ;
3097     print ("lwarpmk: incorrect variable value \" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
3098     print ("lwarpmk: ===")
3099     printconf ( ) ;
3100     os.exit(1) ;

```

```

3101 end
3102
3103 -- Load settings from the project's "lwarpmk.conf" file:
3104
3105 function loadconf ()
3106 -- Default configuration filename:
3107 local conffile = "lwarpmk.conf"
3108 -- Optional configuration filename:
3109 if arg[2] ~= nil then conffile = arg[2].."lwarpmk.conf" end
3110 -- Default language:
3111 language = "english"
3112 -- Default xdyfile:
3113 xdyfile = "lwarp.xdy"
3114 -- Verify the file exists:
3115 if (lfs.attributes(conffile,"mode")==nil) then
3116     -- file not exists
3117     print ("lwarpmk: ===")
3118     print ("lwarpmk: " .. conffile .. " does not exist.")
3119     print ("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
3120     print ("lwarpmk: ===")
3121     printhelp () ;
3122     os.exit(1) -- exit the entire lwarpmk script
3123 else -- file exists
3124 -- Read the file:
3125 print ("lwarpmk: Reading " .. conffile .. ".")
3126 local cfile = io.open(conffile)
3127 -- Scan each line:
3128 local linenum = 0
3129 for line in cfile:lines() do -- scan lines
3130     linenum = linenum + 1
3131     i,j,cvarname,cvalue = string.find (line,"([%w-]*)%s*=%s*\"([%w%-_.]*)\"") ;
3132 -- Error if incorrect enclosing characters:
3133 if ( i == nil ) then
3134     print ("lwarpmk: ===")
3135     print ("lwarpmk: " .. linenum .. " : " .. line ) ;
3136     print ("lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
3137     print ("lwarpmk: ===")
3138     printconf () ;
3139     os.exit(1) ;
3140 end -- nil
3141 if ( cvarname == "opssystem" ) then
3142     -- Verify choice of opssystem:
3143     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
3144         opssystem = cvalue
3145     else
3146         cvalueerror ( line, linenum , cvalue )
3147     end
3148 elseif ( cvarname == "latexname" ) then
3149     -- Verify choice of LaTeX compiler:
3150     if (

```



```

3151         (cvalue == "pdflatex") or
3152         (cvalue == "xelatex") or
3153         (cvalue == "lualatex")
3154     ) then
3155         latexname = cvalue
3156     else
3157         cvalueerror ( line, linenum , cvalue )
3158     end
3159 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
3160 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
3161 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
3162 elseif ( cvarname == "latexmk" ) then latexmk = cvalue
3163 elseif ( cvarname == "language" ) then language = cvalue
3164 elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue
3165 else
3166     print ("lwarpmk: ===")
3167     print ("lwarpmk: " .. linenum .. " : " .. line ) ;
3168     print ("lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. confname .. ".\n" )
3169     print ("lwarpmk: ===")
3170     printconf () ;
3171 os.exit(1) ;
3172 end -- cvarname
3173 end -- do scan lines
3174 io.close(cfile)
3175 end -- file exists
3176 -- Error if sourcename is "lwarp".
3177 -- This could happen if a local copy of lwarp has recently been recompiled.
3178 if sourcename=="lwarp" then
3179     print ("lwarpmk: ===")
3180     print ("lwarpmk: Lwarp has recently been recompiled in this directory,")
3181     print ("lwarpmk: and \"lwarpmk.conf\" is no longer set for your own project.")
3182     print ("lwarpmk: Recompile your own project using pdf/luaxelatex <projectname>.")
3183     print ("lwarpmk: After a recompile, \"lwarpmk.conf\" will be set for your project,")
3184     print ("lwarpmk: and you may again use lwarpmk.")
3185     print ("lwarpmk: ===")
3186     os.exit(1)
3187 end -- sourcename of "lwarp"
3188 -- Select some operating-system commands:
3189 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
3190     rmname = "rm"
3191     mvname = "mv"
3192     cpname = "cp"
3193     touchnamepre = "touch"
3194     touchnamepost = ""
3195     newtouchname = "touch"
3196     dirslash = "/"
3197     opquote= "\""
3198     cmdgroupopenname = " ( "
3199     cmdgroupclosename = " ) "
3200     seqname = " ; "

```

```
3201     bname = " &"
3202 elseif opsystem=="Windows" then -- For Windows
3203     rmname = "DEL"
3204     mvname = "MOVE"
3205     cpname = "COPY"
3206     touchnamepre = "COPY /b"
3207     touchnamepost = "+,,"
3208     newtouchname = "echo empty >"
3209     dirslash = "\\\"
3210     opquote= "\""
3211     cmdgroupopenname = ""
3212     cmdgroupclosename = ""
3213     seqname = " & "
3214     bname = ""
3215 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
3216 end --- for Windows
3217
3218 -- set xindycmd according to pdflatex vs xelatex/lualatex:
3219 if ( latexname == "pdflatex" ) then
3220     xindycmd = "texindy -C utf8"
3221     glossarycmd = "xindy -C utf8"
3222 else
3223     xindycmd = "xindy -M texindy -C utf8"
3224     glossarycmd = "xindy -C utf8"
3225 end
3226
3227 end -- loadconf
3228
3229
3230 function refreshdate ()
3231 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
3232 end
3233
3234
3235 -- Scan the LaTeX log file for the phrase "Rerun to get",
3236 -- indicating that the file should be compiled again.
3237 -- Return true if found.
3238
3239 function reruntoget (filesorce)
3240 local fsource = io.open(filesorce)
3241 for line in fsource:lines() do
3242 if ( string.find(line,"Rerun to get") ~= nil ) then
3243     io.close(fsource)
3244     return true
3245 end -- if
3246 end -- do
3247 io.close(fsource)
3248 return false
3249 end
3250
```

```
3251
3252-- Compile one time, return true if should compile again.
3253-- fsuffix is "" for print, "_html" for HTML output.
3254
3255function onetime (fsuffix)
3256print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
3257err = os.execute(latexname .. " " .. sourcename..fsuffix)
3258if ( err ~= 0 ) then
3259    print ("lwarpmk: ===")
3260    print ("lwarpmk: Compile error.")
3261    print ("lwarpmk: ===")
3262    os.exit(1)
3263end
3264return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
3265end
3266
3267
3268-- Compile up to five times.
3269-- fsuffix is "" for print, "_html" for HTML output
3270
3271function manytimes (fsuffix)
3272if onetime(fsuffix) == true then
3273if onetime(fsuffix) == true then
3274if onetime(fsuffix) == true then
3275if onetime(fsuffix) == true then
3276if onetime(fsuffix) == true then
3277end end end end end
3278end
3279
3280-- Exit if the given file does not exist.
3281
3282function verifyfileexists (filename)
3283if (lfs.attributes ( filename , "modification" ) == nil ) then
3284    print ("lwarpmk: ===")
3285    print ("lwarpmk: " .. filename .. " not found." ) ;
3286    print ("lwarpmk: ===")
3287    os.exit (1) ;
3288end
3289end
3290
3291
3292-- Convert <project>_html.pdf into HTML files:
3293
3294function pdftohtml ()
3295-- Convert to text:
3296print ("lwarpmk: Converting " .. sourcename
3297    .. "_html.pdf to " .. sourcename .. "_html.html")
3298os.execute("pdftotext -enc UTF-8 -nopgbrk -layout "
3299    .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
3300-- Split the result into individual HTML files:
```

```

3301 splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
3302 end
3303
3304
3305 -- Remove auxiliary files:
3306
3307 function removeaux ()
3308 os.execute ( rmname .. " " ..
3309     sourcename .. ".aux " .. sourcename .. "_html.aux " ..
3310     sourcename .. ".toc " .. sourcename .. "_html.toc " ..
3311     sourcename .. ".lof " .. sourcename .. "_html.lof " ..
3312     sourcename .. ".lot " .. sourcename .. "_html.lot " ..
3313     sourcename .. ".idx " .. sourcename .. "_html.idx " ..
3314     sourcename .. ".ind " .. sourcename .. "_html.ind " ..
3315     sourcename .. ".log " .. sourcename .. "_html.log " ..
3316     sourcename .. ".gl* " .. sourcename .. "_html.gl* " ..
3317     "*_html_inc.*"
3318 )
3319 end
3320
3321 -- Create lateximages based on lateximages.txt:
3322 function createlateximages ()
3323 print ("lwarpmk: Creating lateximages.")
3324 local limagesfile = io.open("lateximages.txt", "r")
3325 if ( limagesfile == nil ) then
3326     print ("lwarpmk: ===")
3327     print ("lwarpmk: \"lateximages.txt\" does not exist.")
3328     print ("lwarpmk: Your project does not use SVG math or other lateximages,")
3329     print ("lwarpmk: or the file has been deleted somehow.")
3330     print ("lwarpmk: Use \"lwarpmk html\" to recompile your project,")
3331     print ("lwarpmk: and recreate \"lateximages.txt\".")
3332     print ("lwarpmk: If your project does not use SVG math or other lateximages,")
3333     print ("lwarpmk: then \"lateximages.txt\" will never exist, and")
3334     print ("lwarpmk: \"lwarpmk limages\" will not be necessary.")
3335     print ("lwarpmk: ===")
3336     os.exit(1)
3337 end
3338 -- Create the lateximages directory, ignore error if already exists
3339 err = os.execute("mkdir lateximages")
3340 -- For Windows, create lwarp_one_limage.cmd:
3341 if opsystem=="Windows" then
3342     err = os.execute (
3343         cpname .. " lwarp_one_limage.txt lwarp_one_limage.cmd"
3344     )
3345     if ( err ~= 0 ) then
3346         print ("lwarpmk: ===")
3347         print ("lwarpmk: File error trying to copy to lwarp_one_limage.cmd")
3348         print ("lwarpmk: ===")
3349         os.exit(1) ;
3350     end

```

```

3351 end -- create lwarp_one_limage.cmd
3352 -- Scan lateximages.txt
3353 for line in limagesfile:lines() do
3354 -- lwimgpage is the page number in the PDF which has the image
3355 -- lwimghash is true if this filename is a hash
3356 -- lwimgname is the lateximage filename root to assign for the image
3357 i,j,lwimgpage,lwimghash,lwimgname = string.find (line,"|(.*)|(.*)|(.*)|")
3358 -- For each entry:
3359 if ( (i~=nil) ) then
3360 -- Skip is this image is hashed and already exists:
3361 local lwimgfullname = "lateximages" .. dirslash .. lwimgname .. ".svg"
3362 if (
3363     (lwimghash ~= "true") or
3364     (lfs.attributes(lwimgfullname,"mode")==nil) -- file not exists
3365 )
3366 then -- not hashed or not exists:
3367 -- Print the name of the file being generated:
3368 print ( "lwarpmk: " .. lwimgname )
3369 -- Touch/create the dest so that only once instance tries to build it:
3370 err = os.execute(newtouchname .. " " .. lwimgfullname)
3371 if ( err ~= 0 ) then
3372     print ("lwarpmk: ===")
3373     print ("lwarpmk: File error trying to touch " .. lwimgfullname)
3374     print ("lwarpmk: ===")
3375     os.exit(1) ;
3376 end
3377 -- Separate out the image into its own single-page pdf:
3378 if opsystem=="Unix" then
3379 -- For Unix / Linux / Mac OS:
3380 err = os.execute(
3381 -- print (
3382 cmdgroupopenname ..
3383 "pdfseparate -f " .. lwimgpage .. " -l " .. lwimgpage .. " " ..
3384     sourcename .. "_html.pdf " ..
3385     "lateximages" .. dirslash .. "lateximagetemp-%d" .. ".pdf" ..
3386     seqname ..
3387 -- Crop the image:
3388 "pdfcrop lateximages" .. dirslash .. "lateximagetemp-" .. lwimgpage .. ".pdf " ..
3389     "lateximages" .. dirslash .. lwimgname .. ".pdf" ..
3390     seqname ..
3391 -- Convert the image to svg:
3392 "pdftocairo -svg lateximages" .. dirslash .. lwimgname .. ".pdf " ..
3393     "lateximages" .. dirslash .. lwimgname .. ".svg" ..
3394     seqname ..
3395 -- Remove the temporary files:
3396 rmname .. " lateximages" .. dirslash .. lwimgname .. ".pdf" .. seqname ..
3397 rmname .. " lateximages" .. dirslash .. "lateximagetemp-" .. lwimgpage .. ".pdf" ..
3398 cmdgroupclosename .. " >/dev/null " .. bgnome
3399 )
3400 elseif opsystem=="Windows" then

```

```

3401 -- For Windows
3402 err = os.execute (
3403     "start /b \"%\" lwarp_one_limage " ..
3404     lwimgpage .. " " ..
3405     lwimghash .. " " ..
3406     lwimgname .. " " ..
3407     sourcename .. " <nul >nul"
3408 )
3409 end -- Windows
3410 if ( err ~= 0 ) then
3411     print ("lwarpmk: ===")
3412     print ("lwarpmk: File error trying to create one lateximage.")
3413     print ("lwarpmk: ===")
3414     os.exit(1)
3415 end
3416 end -- not hashed or not exists
3417 end -- not nil
3418 end -- do
3419 io.close(limagesfile)
3420 print ( "lwarpmk limages: done" )
3421 end -- function
3422
3423
3424 -- Use latexmk to compile source and index:
3425 -- fsuffix is "" for print, or "_html" for HTML
3426 function compilelatexmk ( fsuffix )
3427 -- The recorder option is required to detect changes in <project>.tex
3428 -- while we are loading <project>_html.tex.
3429 err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
3430     .. "-e "
3431     .. opquote
3432     .. "$makeindex = q/" -- $
3433     .. xindycmd
3434     .. " -M " .. xdyfile
3435     .. " -L " .. language .. " /"
3436     .. opquote
3437     .. "-pdflatex=\"\" .. latexname .." %0 %S\" "
3438     .. sourcename..fsuffix ..".tex" ) ;
3439 if ( err ~= 0 ) then
3440     print ("lwarpmk: ===")
3441     print ("lwarpmk: Compile error.")
3442     print ("lwarpmk: ===")
3443     os.exit(1)
3444 end
3445 end
3446
3447
3448
3449 -- lwarpmk --version :
3450

```

```
3451 if (arg[1] == "--version") then
3452   print ( "lwarpmk: " .. printversion )
3453
3454 else -- not --version
3455
3456 -- print intro:
3457
3458 print ("lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package.")
3459
3460 -- lwarpmk print:
3461
3462 if arg[1] == "print" then
3463   loadconf ()
3464   if ( latexmk == "true" ) then
3465     compilelatexmk ("" )
3466     print ("lwarpmk: Done.")
3467   else -- not latexmk
3468     verifyfileexists (sourcename .. ".tex") ;
3469     -- See if up to date:
3470     if (
3471       ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
3472       (
3473         lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3474         lfs.attributes ( sourcename .. ".pdf" , "modification" )
3475       )
3476     ) then
3477       -- Recompile if not yet up to date:
3478       manytimes("")
3479       print ("lwarpmk: Done.") ;
3480     else
3481       print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
3482     end
3483   end -- not latexmk
3484
3485 elseif arg[1] == "print1" then
3486   loadconf ()
3487   verifyfileexists (sourcename .. ".tex") ;
3488   onetime("")
3489   print ("lwarpmk: Done.") ;
3490
3491 -- lwarp printindex:
3492 -- Compile the index then touch the source
3493 -- to trigger a recompile of the document:
3494
3495 elseif arg[1] == "printindex" then
3496   loadconf ()
3497   print ("lwarpmk: Processing the index.")
3498   os.execute(
3499     xindy cmd
3500     .. " -M " .. xdyfile
```

```
3501    .. " -L " .. language
3502    .. " " .. sourcename .. ".idx")
3503print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3504refreshdate ()
3505print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3506print ("lwarpmk: Done.")
3507
3508-- lwarp printglossary:
3509-- Compile the glossary then touch the source
3510-- to trigger a recompile of the document:
3511
3512elseif arg[1] == "printglossary" then
3513loadconf ()
3514print ("lwarpmk: Processing the glossary.")
3515
3516os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
3517    " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
3518    .. sourcename .. ".glo")
3519print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3520refreshdate ()
3521print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3522print ("lwarpmk: Done.")
3523
3524-- lwarpmk html:
3525
3526elseif arg[1] == "html" then
3527loadconf ()
3528if ( latexmk == "true" ) then
3529    compilelatexmk ("_html")
3530    pdftohtml ()
3531    print ("lwarpmk: Done.")
3532else -- not latexmk
3533    verifyfileexists ( sourcename .. ".tex" ) ;
3534    -- See if exists and is up to date:
3535    if (
3536        ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
3537        (
3538            lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3539            lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
3540        )
3541    ) then
3542        -- Recompile if not yet up to date:
3543        manytimes("_html")
3544        pdftohtml ()
3545        print ("lwarpmk: Done.")
3546    else
3547        print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
3548    end
3549end -- not latexmk
3550
```



```
3551 elseif arg[1] == "html1" then
3552     loadconf ()
3553     verifyfileexists ( sourcename .. ".tex" ) ;
3554     onetime("_html")
3555     pdftohtml ()
3556     print ("lwarpmk: Done.")
3557
3558 elseif arg[1] == "pdftohtml" then
3559     loadconf ()
3560     pdftohtml ()
3561
3562 -- lwarpmk htmlindex:
3563 -- Compile the index then touch the source
3564 -- to trigger a recompile of the document:
3565
3566 elseif arg[1] == "htmlindex" then
3567 loadconf ()
3568 print ("lwarpmk: Processing the index.")
3569 os.execute(
3570     xindycmd
3571     .. " -M " .. xdyfile
3572     .. " -L " .. language
3573     .. " " .. sourcename .. "_html.idx"
3574 )
3575 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3576 refreshdate ()
3577 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3578 print ("lwarpmk: Done.")
3579
3580 -- lwarpmk htmlglossary:
3581 -- Compile the glossary then touch the source
3582 -- to trigger a recompile of the document:
3583
3584 elseif arg[1] == "htmlglossary" then
3585 loadconf ()
3586 print ("lwarpmk: Processing the glossary.")
3587
3588 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
3589     "_html -t " .. sourcename .. "_html.glg -o " .. sourcename ..
3590     "_html.gls " .. sourcename .. "_html.glo")
3591
3592 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3593 refreshdate ()
3594 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3595 print ("lwarpmk: Done.")
3596
3597 -- lwarpmk limages:
3598 -- Scan the lateximages.txt file to create lateximages,
3599 -- then touch the source to trigger a recompile.
3600
```

```
3601 elseif arg[1] == "limages" then
3602 loadconf ()
3603 print ("lwarpmk: Processing images.")
3604 createlateximages ()
3605 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3606 refreshdate ()
3607 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3608 print ("lwarpmk: Done.")
3609
3610 -- lwarpmk again:
3611 -- Touch the source to trigger a recompile.
3612
3613 elseif arg[1] == "again" then
3614 loadconf ()
3615 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3616 refreshdate ()
3617 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3618 print ("lwarpmk: Done.")
3619
3620 -- lwarpmk clean:
3621 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
3622
3623 elseif arg[1] == "clean" then
3624 loadconf ()
3625 removeaux ()
3626 print ("lwarpmk: Done.")
3627
3628 -- lwarpmk cleanall
3629 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
3630 -- and also project.pdf, *.html
3631
3632 elseif arg[1] == "cleanall" then
3633 loadconf ()
3634 removeaux ()
3635 os.execute ( rmname .. " " ..
3636     sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
3637     "*.html"
3638 )
3639 print ("lwarpmk: Done.")
3640
3641 -- lwarpmk with no argument :
3642
3643 elseif (arg[1] == nil) then
3644 printhelp ()
3645
3646 -- lwarpmk -h or lwarpmk --help :
3647
3648 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
3649 printusage ()
3650
```

```

3651 else
3652 print ("lwarpmk: Unknown command \"\"..arg[1]..\"\".\n")
3653 printhelp ()
3654 end
3655
3656 end -- not --version
3657 \end{filecontents*}
3658 % \end{Verbatim}% for syntax highlighting
3659 \end{LWR@createlwarpmk}

```

## 36 Stacks

for HTML output: 3660 \begin{warpHTML}



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

### 36.1 Assigning depths

initial depths for empty stack entries:

```
3661 \newcommand*{\LWR@depthnone}{-5}
```

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

```

3662 \newcommand*{\LWR@depthfinished}{-4}
3663 \newcommand*{\LWR@depthpart}{-1}
3664 \newcommand*{\LWR@depthchapter}{0}
3665 \newcommand*{\LWR@depthsection}{1}
3666 \newcommand*{\LWR@depthsubsection}{2}
3667 \newcommand*{\LWR@depthsubsubsection}{3}
3668 \newcommand*{\LWR@depthparagraph}{4}
3669 \newcommand*{\LWR@depthsubparagraph}{5}

```

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

```
3670 \newcommand*{\LWR@depthlist}{6}
```

used by `\item`:

```
3671 \newcommand*{\LWR@depthlistitem}{7}
```

## 36.2 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.

```
3672 \newcommand*{\LWR@closeone}{}% top of the stack
3673 \newcommand*{\LWR@closetwo}{}
3674 \newcommand*{\LWR@closethree}{}
3675 \newcommand*{\LWR@closefour}{}
3676 \newcommand*{\LWR@closefive}{}
3677 \newcommand*{\LWR@closesix}{}
3678 \newcommand*{\LWR@closeseven}{}
3679 \newcommand*{\LWR@closeeight}{}
3680 \newcommand*{\LWR@closenine}{}
3681 \newcommand*{\LWR@closeten}{}
3682 \newcommand*{\LWR@closeeleven}{}
3683 \newcommand*{\LWR@closetwelve}{}%
```

## 36.3 Closing depths

A stack to record the depth of each level:



Note that nested  $\TeX$  structures may push depths which are non-sequential.

*Ex:*

---

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

---

```

3684 \newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
3685 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
3686 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
3687 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
3688 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
3689 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
3690 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
3691 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
3692 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
3693 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}
3694 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
3695 \newcommand*{\LWR@closedepthtwelve}{\LWR@depthnone}

```

### 36.4 Pushing and popping the stack

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

Pushes one return action and its  $\mathbb{E}_T X$  depth onto the stacks.

```

3696 \NewDocumentCommand{\pushclose}{m m}
3697 {
3698 \global\let\LWR@closetwelve\LWR@closeeleven
3699 \global\let\LWR@closeeleven\LWR@closeten
3700 \global\let\LWR@closeten\LWR@closenine
3701 \global\let\LWR@closenine\LWR@closeeight
3702 \global\let\LWR@closeeight\LWR@closeseven
3703 \global\let\LWR@closeseven\LWR@closesix
3704 \global\let\LWR@closesix\LWR@closefive
3705 \global\let\LWR@closefive\LWR@closefour
3706 \global\let\LWR@closefour\LWR@closethree
3707 \global\let\LWR@closethree\LWR@closetwo
3708 \global\let\LWR@closetwo\LWR@closeone
3709 \global\let\LWR@closeone#1
3710 \global\let\LWR@closedepthtwelve\LWR@closedeptheleven
3711 \global\let\LWR@closedeptheleven\LWR@closedepthten
3712 \global\let\LWR@closedepthten\LWR@closedepthnine
3713 \global\let\LWR@closedepthnine\LWR@closedeptheight
3714 \global\let\LWR@closedeptheight\LWR@closedepthseven
3715 \global\let\LWR@closedepthseven\LWR@closedepthsix
3716 \global\let\LWR@closedepthsix\LWR@closedepthfive
3717 \global\let\LWR@closedepthfive\LWR@closedepthfour
3718 \global\let\LWR@closedepthfour\LWR@closedepththree
3719 \global\let\LWR@closedepththree\LWR@closedepthtwo
3720 \global\let\LWR@closedepthtwo\LWR@closedepthone
3721 \global\let\LWR@closedepthone#2
3722 }

```

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

```

3723 \newcommand*{\popclose}
3724 {
3725 \global\let\LWR@closeone\LWR@closetwo
3726 \global\let\LWR@closetwo\LWR@closethree
3727 \global\let\LWR@closethree\LWR@closefour
3728 \global\let\LWR@closefour\LWR@closefive
3729 \global\let\LWR@closefive\LWR@closesix
3730 \global\let\LWR@closesix\LWR@closeseven
3731 \global\let\LWR@closeseven\LWR@closeeight
3732 \global\let\LWR@closeeight\LWR@closenine
3733 \global\let\LWR@closenine\LWR@closeten
3734 \global\let\LWR@closeten\LWR@closeeleven
3735 \global\let\LWR@closeeleven\LWR@closetwelve
3736 \global\let\LWR@closedepthone\LWR@closedepthtwo
3737 \global\let\LWR@closedepthtwo\LWR@closedepththree
3738 \global\let\LWR@closedepththree\LWR@closedepthfour
3739 \global\let\LWR@closedepthfour\LWR@closedepthfive
3740 \global\let\LWR@closedepthfive\LWR@closedepthsix
3741 \global\let\LWR@closedepthsix\LWR@closedepthseven
3742 \global\let\LWR@closedepthseven\LWR@closedeptheight
3743 \global\let\LWR@closedeptheight\LWR@closedepthnine
3744 \global\let\LWR@closedepthnine\LWR@closedepthten
3745 \global\let\LWR@closedepthten\LWR@closedeptheleven
3746 \global\let\LWR@closedeptheleven\LWR@closedephtwelve
3747 }

3748 \end{warpHTML}

```

## 37 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: 3749 \begin{warpHTML}

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

```

3750 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
3751 \ifstrempy{#3}%
3752 {\csdef{#1#2}{}}%
3753 {\expandafter\edef\csname #1#2\endcsname{\expandonce#3}}%
3754 }

```

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

```

3755 \newcommand*{\LWR@getexparray}[2]{\csuse{#1#2}}

3756 \end{warpHTML}

```

## 38 Sanitizing labels and filenames

Special handling for underscores in labels and filenames.

**for HTML output:** 3757 \begin{warpHTML}

`\LWR@sanitized`    The sanitized version of what was given to `\LWR@sanitize`. Characters are set to their detokenized versions. Required for underscores in labels and filenames.

```

3758 \newcommand*{\LWR@sanitized}{}

```

`\LWR@sanitize`     $\{\langle text \rangle\}$

Sanitizes the text and returns the result in `\LWR@sanitized`.

```

3759 \newcommand*{\LWR@sanitize}[1]{%
3760 \LWR@traceinfo{\LWR@sanitize: !#1!}%
3761 \edef\LWR@sanitized{#1}%
3762 \LWR@traceinfo{\LWR@sanitize expanded: !\LWR@sanitized!}%
3763 \edef\LWR@sanitized{\detokenize\expandafter{\LWR@sanitized}}%
3764 \LWR@traceinfo{\LWR@sanitize result: !\LWR@sanitized!}%
3765 }

3766 \end{warpHTML}

```

## 39 HTML entities

**for HTML output:** 3767 \begin{warpHTML}

HTML entites and HTML Unicode entities:

```
3768 \let\LWR@origampersand\&
```

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

```
3769 \newcommand*{\HTMLentity}[1]{%
3770 % \LWR@traceinfo{HTMLentity \detokenize{#1}}%
3771 \begingroup%
3772 \LWR@FBcancel%
3773 \LWR@origampersand#1;%
3774 \endgroup
3775 % \LWR@traceinfo{HTMLentity done}%
3776 }
```

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

```
3777 \newcommand*{\HTMLunicode}[1]{\HTMLentity{\#x#1}}
```

```
\&
```

```
3778 \renewrobustcmd*{\&}{\HTMLentity{amp}}
```

```
\textless
```

```
\textgreater
```

```
3779 \let\LWR@origtextless\textless
3780 \renewcommand*{\textless}{\HTMLentity{lt}}
3781
3782 \let\LWR@origtextgreater\textgreater
3783 \renewcommand*{\textgreater}{\HTMLentity{gt}}

3784 \end{warpHTML}
```

## 40 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:** 3785 \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.

```
3786 \providecommand*{\BaseJobname}{\jobname}
```

`\HTMLFilename` The prefix for all generated HTML files other than the home page, defaulting to empty. See section 7.3.1.

```
3787 \providecommand*{\HTMLFilename}{}
```

`\HomeHTMLFilename` The filename of the home page, defaulting to the `\BaseJobname`. See section 7.3.1.

```
3788 \providecommand*{\HomeHTMLFilename}{\BaseJobname}
```

`\SetHTMLFileNumber` `{\langle number \rangle}`

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

```
3789 \newcommand*{\SetHTMLFileNumber}[1]{%
3790 \setcounter{LWR@htmlfilenumber}{#1}%
3791 }
```

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

Defaults to use section names in the filenames.

```
3792 \newbool{FileSectionNames}
3793 \booltrue{FileSectionNames}
```

```
3794 \end{warpall}
```

**for HTML output:** `3795 \begin{warpHTML}`

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

```
3796 \newcounter{LWR@htmlfilenumber}
3797 \setcounter{LWR@htmlfilenumber}{0}
```

`\LWR@htmlsectionfilename` `{\langle htmlfilenumber or name \rangle}`

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

```
3798 \newcommand*{\LWR@htmlsectionfilename}[1]{%
3799 \LWR@traceinfo{\LWR@htmlsectionfilename A !\detokenize{#1}!}%
```

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

```
3800 % \LWR@traceinfo{about to assign temp}%
3801 \edef\LWR@tempone{#1}%
3802 \LWR@traceinfo{about to compare with ??}%
3803 \ifthenelse{\equal{\LWR@tempone}{??}}{%
3804 {\LWR@traceinfo{found ??}}%
3805 {\LWR@traceinfo{not found ??}}%
3806 \LWR@traceinfo{about to compare with zero or empty}%
3807 \ifthenelse{%
3808   \equal{\LWR@tempone}{0}%
3809   \OR \equal{\LWR@tempone}{}%
3810   \OR \equal{\LWR@tempone}{??}}%
3811 }%
3812 {%
3813   \LWR@traceinfo{\LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
3814   \HomeHTMLFilename.html%
3815 }%
```

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

```
3816 {%
3817   \LWR@traceinfo{\LWR@htmlsectionfilename C \LWR@tempone}%
3818   \ifthenelse{%
3819     \equal{\HTMLFilename}{ } \AND
3820     \equal{\LWR@tempone}{Index} \OR
3821     \equal{\LWR@tempone}{index}}%
3822   }%
3823   {%
3824     \LWR@traceinfo{Prefixing the index name with an underscore.}%
3825     \_#1.html%
3826   }%
```

Otherwise, create a filename with the chosen prefix:

```
3827   {\HTMLFilename#1.html}%
3828 }%
3829 \LWR@traceinfo{\LWR@htmlsectionfilename Z}%
3830 }
```

\LWR@htmlrefsectionfilename {<label>}

Prints the filename for the given label

```
3831 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
3832 \LWR@traceinfo{\LWR@htmlrefsectionfilename: !\detokenize{#1}!}%
```

\LWR@nullfonts to allow math in a section name.

```
3833 \begingroup%
3834 \LWR@nullfonts%
3835 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
3836 \endgroup%
3837 \LWR@traceinfo{\LWR@htmlrefsectionfilename: done}%
3838 }

3839 \end{warpHTML}
```

## 41 Homepage link

**for HTML output:** 3840 \begin{warpHTML}

**\LinkHome** May be used wherever you wish to place a link back to the homepage. The filename must be detokenized for underscores.

```
3841 \newcommand*{\LinkHome}{%
3842 \LWR@subhyperrefclass{%
3843 \HomeHTMLFilename.html}%
3844 {Home}{linkhome}%
3845 }
```

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

```
3846 \newcommand*{\LWR@topnavigation}{
3847 \LWR@htmlclassline{nav}{topnavigation}{\LinkHome}
3848 }
```

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

```
3849 \newcommand*{\LWR@botnavigation}{
3850 \LWR@htmlclassline{nav}{botnavigation}{\LinkHome}
3851 }
```

```
3852 \end{warpHTML}
```

## 42 \LWRPrintStack diagnostic tool



Diagnostics tool: Prints the  $\text{\LaTeX}$  nesting depth values for the stack levels. `\LWR@startpars` is used before printing the stack, so that `\LWRPrintStack` may be called from anywhere in the normal text flow.

**for HTML output:** 3853 `\begin{warpHTML}`

`\LWRPrintStack` Prints the closedepth stack.

```
3854 \newcommand*{\LWR@subprintstack}{
3855 \LWR@closedepthone\ \LWR@closedepthtwo\ \LWR@closedepththree\
3856 \LWR@closedepthfour\ \LWR@closedepthfive\ \LWR@closedepthsix\
3857 \LWR@closedepthseven\ \LWR@closedeptheight\ \LWR@closedepthnine\
3858 \LWR@closedepthten\ \LWR@closedeptheleven\ \LWR@closedepthtwelve\
3859 }
3860
3861 \newcommand*{\LWRPrintStack}{
3862 \LWR@startpars
3863 \LWR@subprintstack
3864 }

3865 \end{warpHTML}
```

**for PRINT output:** 3866 `\begin{warpprint}`

```
3867 \newcommand*{\LWRPrintStack}{}

3868 \end{warpprint}
```

## 43 Closing stack levels

**for HTML output:** 3869 `\begin{warpHTML}`

Close one nested level:

```
3870 \newcommand*{\LWR@closeoneprevious}{%
3871
3872 \LWR@closeone
3873
3874 \popclose
3875 }
```

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

```
3876 \newcommand*\LWR@closeprevious}[1]{
3877 \LWR@traceinfo{\LWR@closeprevious to depth #1, depths are \LWR@subprintstack}%
```

Close any pending paragraph:

```
3878 \LWR@stoppars%
```

Close anything nested deeper than the desired depth. First close anything deeper, then at most one of the same level.

```
3879 \whileboolexpr{test{\ifnumcomp{\LWR@closedepthone}{>}{#1}}}%
3880 {%
3881     \LWR@traceinfo{\LWR@closeprevious: closing out depth \LWR@closedepthone}%
3882     \LWR@closeoneprevious%
3883 }%
3884 \ifboolexpr{test{\ifnumcomp{\LWR@closedepthone}{=}{#1}}}%
3885 {%
3886     \LWR@traceinfo{\LWR@closeprevious: closing out depth \LWR@closedepthone}%
3887     \LWR@closeoneprevious%
3888 }{}%
3889 \LWR@traceinfo{\LWR@closeprevious: done, depths are \LWR@subprintstack}%
3890 }

3891 \end{warpHTML}
```

## 44 PDF pages and styles

**for HTML output:** 3892 `\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.

```
3893 \newcommand*\LWR@forcenewpage){%
3894 \ifinner\else%
3895 \LWR@stoppars\LWR@orignewpage\LWR@startpars%
3896 \fi%
3897 }
```

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

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

3898 \renewcommand*{\pagestyle}[1]{}

\thispagestyle {\langle style \rangle}

3899 \renewcommand*{\thispagestyle}[1]{}

\markboth    {\langle left \rangle} {\langle right \rangle}

3900 \renewcommand*{\markboth}[2]{}

\markright   {\langle right \rangle}

3901 \renewcommand*{\markright}[1]{}

\raggedbottom

3902 \renewcommand*{\raggedbottom}{}

\flushbottom

3903 \renewcommand*{\flushbottom}{}

\sloppy

3904 \renewcommand*{\sloppy}{}

\fussy

3905 \renewcommand*{\fussy}{}

\pagenumbering * {\langle commands \rangle}

3906 \RenewDocumentCommand{\pagenumbering}{s m}{}

3907 \end{warpHTML}
```

## 45 HTML tags, spans, divs, elements

for HTML output: 3908 \begin{warpHTML}

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

```

3909 \newcommand*{\LWR@tagtitle}{h1}
3910 \newcommand*{\LWR@tagtitleend}{/h1}
3911 \newcommand*{\LWR@tagpart}{h2}
3912 \newcommand*{\LWR@tagpartend}{/h2}
3913 \newcommand*{\LWR@tagchapter}{h3}
3914 \newcommand*{\LWR@tagchapterend}{/h3}
3915 \newcommand*{\LWR@tagsection}{h4}
3916 \newcommand*{\LWR@tagsectionend}{/h4}
3917 \newcommand*{\LWR@tagsubsection}{h5}
3918 \newcommand*{\LWR@tagsubsectionend}{/h5}
3919 \newcommand*{\LWR@tagsubsubsection}{h6}
3920 \newcommand*{\LWR@tagsubsubsectionend}{/h6}
3921 \newcommand*{\LWR@tagparagraph}{span class="paragraph"}
3922 \newcommand*{\LWR@tagparagraphend}{/span}
3923 \newcommand*{\LWR@tagsubparagraph}{span class="subparagraph"}
3924 \newcommand*{\LWR@tagsubparagraphend}{/span}
3925
3926 \newcommand*{\LWR@tagregularparagraph}{p}

```

## 45.2 Babel-French

Adjust babel-french for HTML spaces. So far, this only works for pdf<sub>l</sub>atex and xel<sub>l</sub>atex.

*(Emulates or patches code by DANIEL FLIPO.)*

```

3927 \providecommand*{\LWR@FBcancel}{}
3928
3929 \AtBeginDocument{%
3930 \ifundefined{frenchbsetup}%
3931 {}%
3932 {%
3933     \frenchbsetup{FrenchFootnotes=false}%
3934 %
3935     \LetLtxMacro\LWR@FBcancel\NoAutoSpacing%
3936     \renewrobustcmd*{\FBcolonspace}{%
3937         \begingroup%
3938         \LWR@FBcancel%
3939         \LWR@origampersand{} \, %
3940         \endgroup%
3941     }%
3942     \renewrobustcmd*{\FBthinspace}{%
3943         \begingroup%
3944         \LWR@FBcancel%
3945         \LWR@origampersand\#x202f;% \,
3946         \endgroup%

```

```

3947 }%
3948 \renewrobustcmd*{\FBguillspace}{%
3949     \begingroup%
3950     \LWR@FBcancel%
3951     \LWR@origampersand{} \nbsp;% ~, for \og xyz \fg{}
3952     \endgroup%
3953 }%
3954 \DeclareDocumentCommand{\FBmedkern}{-}{%
3955     \begingroup%
3956     \LWR@FBcancel%
3957     \LWR@origampersand{\#x202f;% \,
3958     \endgroup%
3959 }%
3960 \DeclareDocumentCommand{\FBthickkern}{-}{%
3961     \begingroup%
3962     \LWR@FBcancel%
3963     \LWR@origampersand{} \nbsp;% ~
3964     \endgroup%
3965 }%
3966 \renewrobustcmd*{~}{\HTMLentity{nbsp}}% was overwritten by babel-french
3967 \iffBunicode%
3968 \else%
3969     \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}%
3970     \DeclareTextCommandDefault{\FBtextellipsis}{\textellipsis\xspace}%
3971 \fi%
3972 }%
3973 }

```

### 45.3 HTML tags

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

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

```

3974 \newcommand*{\LWR@htmltagc}[1]{%
3975 {%
3976 \LWR@traceinfo{\LWR@htmltagc !\detokenize{#1}!}%
3977 \begingroup%
3978 \LWR@FBcancel%
3979 \ifmmode\else\protect\LWR@origttfamily\fi%
3980 \protect\LWR@origtextless%
3981 #1%
3982 \protect\LWR@origtextgreater%
3983 \endgroup%
3984 \LWR@traceinfo{\LWR@htmltagc: done}%
3985 }%
3986 }

```



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.

```

3987 \newcommand*{\LWR@nestspanitem}{%
3988 \if@newlist\else{\LWR@htmltagc{br /}}\fi%
3989 \LWR@origitem%
3990 }
3991
3992 \newenvironment*{LWR@nestspan}
3993 {%
3994 \LWR@traceinfo{LWR@nestspan starting}%
3995 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
3996 {%
3997   \LWR@traceinfo{LWR@nestspan: inside a lateximage}%
3998 }%
3999 {% not in a lateximage
4000   \LWR@traceinfo{LWR@nestspan: NOT inside a lateximage}%
4001   \addtocounter{LWR@spandepth}{1}%
4002   \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
4003   \RenewDocumentEnvironment{BlockClass}{o m}{-}{-}%
4004   \renewcommand{\BlockClassSingle}[2]{##2}%
4005   \renewcommand{\LWR@forcenewpage}{}%
4006   \renewcommand{\LWR@liststart}{%
4007     \let\item\LWR@nestspanitem%
4008   }%
4009   \renewcommand{\LWR@listend}{\LWR@htmltagc{br /}\LWR@htmltagc{br /}}%
4010 }% not in a lateximage
4011 \LWR@traceinfo{LWR@nestspan starting: done}%
4012 }% starting env
4013 {% ending env
4014 \LWR@traceinfo{LWR@nestspan ending}%
4015 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
4016 }%
4017 {\addtocounter{LWR@spandepth}{-1}}%
4018 \LWR@traceinfo{LWR@nestspan ending: done}%
4019 }
4020
4021 \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.

```

4022 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
4023 \LWR@ensuredoingapar%
4024 \LWR@htmltagc{#1}%
4025 \begin{LWR@nestspan}%
4026 #2%
4027 \LWR@htmltagc{/#1}%
4028 \end{LWR@nestspan}%
4029 }

```

`\LWR@htmlspanclass` [`<style>`] [`<class>`] [`<text>`]

```

4030 \NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{%
4031 \LWR@traceinfo{LWR@htmlspanclass #2}%
4032 \LWR@ensuredoingapar%
4033 \LWR@subhtmlmlelementclass{span}[#1]{#2}%
4034 \begin{LWR@nestspan}%
4035 #3%
4036 \LWR@htmltagc{/span}%
4037 \LWR@traceinfo{LWR@htmlspanclass done}%
4038 \end{LWR@nestspan}%
4039 }

```

`\LWR@htmltag` [`<tag>`]

Print an HTML tag: `<tag>`

```

4040 \newcommand*{\LWR@htmltag}[1]{%
4041 % \LWR@traceinfo{LWR@htmltagb !\detokenize{#1}!}%
4042 \LWR@htmltagc{#1}%
4043 % \LWR@traceinfo{LWR@htmltagb: done}%
4044 }

```

## 45.4 Block tags and comments

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

```

\LWR@htmlopencomment
\LWR@htmlclosecomment
4045 \newcommand*{\LWR@htmlopencomment}{%
4046 {%
4047 % \LWR@traceinfo{LWR@htmlopencomment}%
4048 \begingroup%
4049 \LWR@FBcancel%

```

```

4050 \ifmode\else\protect\LWR@origttfamily\fi%
4051 \LWR@origmbox{\LWR@origtextless{!}{-}{-}}%
4052 \endgroup%
4053 }%
4054 }
4055
4056 \newcommand*{\LWR@htmlclosecomment}{%
4057 {%
4058 % \LWR@traceinfo{\LWR@htmlclosecomment}%
4059 \begingroup%
4060 \LWR@FBcancel%
4061 \ifmode\else\protect\LWR@origttfamily\fi%
4062 \LWR@origmbox{!}{-}{-}\LWR@origtextgreater}%
4063 \endgroup%
4064 }%
4065 }

```

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

```

4066 \newcommand{\LWR@htmlcomment}[1]{%
4067 \LWR@htmlopencomment}%
4068 {%
4069 \LWR@origttfamily% break ligatures
4070 #1%
4071 }%
4072 \LWR@htmlclosecomment{}}

```

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

```

4073 \newcommand{\LWR@htmlblockcomment}[1]
4074 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars}

```

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

```

4075 \newcommand*{\LWR@htmlblocktag}[1]{%
4076 \LWR@stoppars%
4077 \LWR@htmltag{#1}%
4078 \LWR@startpars%
4079 }

```

## 45.5 Div class and element class

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

Factored and reused in several places.

The trailing spaces allow more places for a line break.

```
4080 \NewDocumentCommand{\LWR@subhtmlmlelementclass}{m O{} m}{%
4081 \LWR@traceinfo{\LWR@subhtmlmlelementclass #1 #3}%
4082 \ifblank{#2}%
4083 {\LWR@htmltag{#1 class="#3"}}% empty option
4084 {\LWR@htmltag{#1 class="#3" style="#2"}}% non-empty option
4085 \LWR@traceinfo{\LWR@subhtmlmlelementclass done}%
4086 }
```

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

```
4087 \NewDocumentCommand{\LWR@htmlmlelementclass}{m o m}{%
4088 \LWR@stoppars%
4089 \LWR@subhtmlmlelementclass{#1}[#2]{#3}%
4090 \LWR@startpars%
4091 }
```

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

```
4092 \newcommand*{\LWR@htmlmlelementclassend}[2]{%
4093 \LWR@stoppars%
4094 \LWR@htmltag{/#1}%
4095 \ifbool{HTMLDebugComments}{%
4096     \LWR@htmlcomment{End of #1 ‘‘#2’’}%
4097 }{}%
4098 \LWR@startpars%
4099 }
```

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

```
4100 \NewDocumentCommand{\LWR@htmldivclass}{o m}{%
4101 \LWR@htmlmlelementclass{div}[#1]{#2}%
4102 }
```

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

```
4103 \newcommand*{\LWR@htmldivclassend}[1]{%
4104 \LWR@htmlmlelementclassend{div}{#1}%
4105 }
```

## 45.6 Single-line elements

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

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

```
4106 \NewDocumentCommand{\LWR@htmlelementclassline}{m o m +m}{%
4107 \LWR@stoppars
4108 \LWR@subhtmlelementclass{#1}[#2]{#3}%
4109 #4%
4110 \LWR@htmltag{/#1}
4111 \LWR@startpars
4112 }
```

## 45.7 HTML5 semantic elements

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

```
4113 \newcommand*{\LWR@htmlelement}[1]{%
4114 \LWR@htmlblocktag{#1}
4115 }
```

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

```
4116 \newcommand*{\LWR@htmlelementend}[1]{%
4117 \LWR@stoppars
4118 \LWR@htmltag{/#1}
4119 \LWR@startpars
4120 }
4121
4122 \end{warpHTML}
```

## 45.8 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 [81](#).

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

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

for HTML output: 

```
4123 \begin{warpHTML}
4124 \NewDocumentEnvironment{BlockClass}{o m}%
```

```

4125 {%
4126     \LWR@origpar%
4127     \LWR@htmldivclass[#1]{#2}%
4128 }
4129 {\LWR@htmldivclassend{#2}}
4130 \end{warpHTML}

```

**for PRINT output:**

```

4131 \begin{warpprint}
4132 \NewDocumentEnvironment{BlockClass}{o m}{-}{-}%
4133 \end{warpprint}

```

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

**for HTML output:**

```

4134 \begin{warpHTML}
4135 \newcommand{\BlockClassSingle}[2]{%
4136     \LWR@origpar%
4137     \LWR@htmllementclassline{div}{#1}{#2}%
4138 }
4139 \end{warpHTML}

```

**for PRINT output:**

```

4140 \begin{warpprint}
4141 \newcommand{\BlockClassSingle}[2]{#2}
4142 \end{warpprint}

```

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

**for HTML output:**

```

4143 \begin{warpHTML}
4144 \NewDocumentCommand{\InlineClass}{o m +m}{%
4145     \LWR@htmlspanclass[#1]{#2}{#3}%
4146 }
4147 \end{warpHTML}

```

**for PRINT output:**

```

4148 \begin{warpprint}
4149 \NewDocumentCommand{\InlineClass}{o m +m}{#3}%
4150 \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:**

```

4151 \begin{warpHTML}
4152 \NewDocumentEnvironment{LWR@BlockClassWP}{m m m}%
4153 {%
4154     \LWR@stoppars%
4155     \ifbool{FormatWP}%
4156 {%
4157         \addtocounter{LWR@thisautoidWP}{1}%
4158         \LWR@htmltag{%
4159             div class="#3" %

```

```

4160      id="\LWR@origmbox{autoidWP-\arabic{\LWR@thisautoidWP}}"%
4161      \ifblank{#1}{\ style="#1"}%
4162    }%
4163 }% FormatWP
4164 {% not FormatWP
4165   \LWR@htmltag{%
4166     div class="#3"%
4167     \ifblank{#2}{\ style="#2"}%
4168   }%
4169 }% not FormatWP
4170 \LWR@startpars%
4171 }
4172 {\LWR@htmldivclassend{#3}}
4173 \end{warpHTML}

```

**for PRINT output:**

```

4174 \begin{warpprint}
4175 \NewDocumentEnvironment{\LWR@BlockClassWP}{m m m}{\}{\}%
4176 \end{warpprint}

```

## 45.9 Closing HTML tags

**for HTML output:**

```

4177 \begin{warpHTML}

```

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

```

4178 \newcommand*{\LWR@printclosepart}
4179   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
4180 \newcommand*{\LWR@printclosechapter}
4181   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
4182 \newcommand*{\LWR@printclosesection}
4183   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
4184 \newcommand*{\LWR@printclosesubsection}
4185   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
4186 \newcommand*{\LWR@printclosesubsubsection}
4187   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
4188 \newcommand*{\LWR@printcloseparagraph}
4189   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
4190 \newcommand*{\LWR@printclosesubparagraph}
4191   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}

```

Lists require closing HTML tags:

```

4192 \newcommand*{\LWR@printcloselistitem}
4193   {\LWR@htmltag{/li}}
4194 \newcommand*{\LWR@printclosedescitem}

```

```

4195     {\LWR@htmltag{/dd}}
4196 \newcommand*{\LWR@printcloseitemize}
4197     {\LWR@htmltag{/ul}}
4198 \newcommand*{\LWR@printcloseenumerate}
4199     {\LWR@htmltag{/ol}}
4200 \newcommand*{\LWR@printclosedescription}
4201     {\LWR@htmltag{/dl}}

4202 \end{warpHTML}

```

## 46 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 47 has high-level commands which allow paragraph-tag generation to start/stop. Even when allowed (`\LWR@doingstartpars`), tags are not generated until a  $\LaTeX$  paragraph is being used (`\LWR@doingapar`). `LWR@lateximagedepth` is used to prevent nesting tags inside a `lateximage`. `LWR@spandepth` is used to prevent nesting paragraph tags inside a paragraph, which became important inside `\fbox` commands and other spans.

**for HTML output:** 4203 `\begin{warpHTML}`

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

```

4204 \newcounter{LWR@spandepth}
4205 \setcounter{LWR@spandepth}{0}

```

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

```

4206 \newbool{LWR@doingstartpars}
4207 \boolfalse{LWR@doingstartpars}

```

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

```

4208 \newbool{LWR@doingapar}
4209 \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:

```

4210 \newcommand*{\LWR@ensuredoingapar}{%
4211 \ifbool{LWR@doingstartpars}%

```



```

4212 {\global\booltrue{LWR@doingapar}}%
4213 {}%
4214 }

```

\LWR@openparagraph

```

4215 \newcommand*{\LWR@openparagraph}
4216 {%

```

See if paragraph handling is enabled:

```

4217 \ifbool{LWR@doingstartpars}%
4218 {% handling pars

```

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

```

4219   \ifboolexpr{
4220       test {\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}} or
4221       test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}}
4222   }% nested par tags?

```

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

```

4223   {}% no nested par tags

```

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

```

4224   {% yes nest par tags
4225       \LWR@htmltagc{\LWR@tagregularparagraph}%

```

Now have started a paragraph.

```

4226       \global\booltrue{LWR@doingapar}%

```

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

```

4227       \let\par\LWR@closeparagraph%
4228   }% end of yes nest par tags
4229 }% end of handling pars
4230 {}% not handling pars
4231 }

```

\LWR@closeparagraph

```
4232 \newcommand*{\LWR@closeparagraph}
4233 {%
4234 \LWR@traceinfo{\LWR@closeparagraph}%
```

See if paragraph handling is enabled:

```
4235 \ifbool{\LWR@doingapar}%
```

If currently in paragraph mode:

```
4236 {% handling pars
```

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

```
4237 \ifboolexpr{
4238     test {\ifnumcomp{\value{\LWR@lateximagedepth}}{>}{0}} or
4239     test {\ifnumcomp{\value{\LWR@spandepth}}{>}{0}}
4240 }%
```

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

```
4241 {% no nested par tags
4242     \ifboolexpr{
4243         test {\ifnumcomp{\value{\LWR@spandepth}}{>}{0}} and
4244         test {\ifnumcomp{\value{\LWR@lateximagedepth}}{=}{0}}
4245     }%
4246     {\ifbool{\LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
4247     }%
4248 }% no nested par tags
```

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

```
4249 {% yes nest par tags
```

Print a closing tag and some extra vertical space:

```
4250 \unskip%
4251 \LWR@htmltagc{/\LWR@tagregularparagraph}%
4252 \LWR@orignewline%
```

No longer doing a paragraph:

```
4253 \global\boolfalse{\LWR@doingapar}%
4254 % Disable the special \env{minipage} \& \cs{hspace} interaction
4255 % until a new minipage is found:
```

```

4256 %    \begin{macrocode}
4257       \global\boolfalse{LWR@minipagethispar}%
4258     }% end of yes nest par tags
4259 }% end of handling pars

```

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

```

4260 {% not handling pars
4261     \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
4262     {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
4263     }%
4264 }% not handling pars

```

In most cases, finish with a  $\LaTeX$  `\par`, but in the case of paragraphs between lines in a tabular fetch the next token instead:

```

4265 \ifboolexpr{%
4266     not bool {LWR@doingapar} and
4267     test {\ifnumcomp{\value{LWR@tabulardepth}}{>}{0}} and
4268     test {
4269         \ifnumcomp{\value{LWR@tabulardepth}}{=}{\value{LWR@tabularpardepth}}
4270     } and
4271     bool {LWR@intabularmetadata} and
4272     not bool {LWR@tableparcell} and
4273     test {\ifnumcomp{\value{LWR@lateximagedepth}}{=}{0}}
4274 }%
4275 {%
4276     \LWR@getmynexttoken%
4277 }{%
4278     \LWR@origpar%
4279 }%
4280 }

4281 \end{warpHTML}

```

## 47 Paragraph start/stop handling

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

Section 46 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: 4282 \begin{warpHTML}

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

```
4283 \newcommand*{\LWR@startpars}%
4284 {%
4285 \LWR@traceinfo{\LWR@startpars}%
```

Ignore if inside a span:

```
4286 \ifnumcomp{\value{\LWR@spandepth}}{>}{0}%
4287 {%}
4288 {%
```

See if currently handling HTML paragraphs:

```
4289 \ifbool{\LWR@doingstartpars}%
```

If already in paragraph mode, do nothing.

```
4290 {%}
```

If not currently in paragraph mode:

```
4291 {%
```

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

```
4292 \PushPreHook{par}{\LWR@openparagraph}%
```

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

```
4293 \let\par\LWR@closeparagraph
4294
4295 }% an intentionally blank line
```

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

```
4296 \global\setbool{\LWR@doingstartpars}{true}%
```

No `<par>` tag yet to undo:

```
4297 \global\boolfalse{\LWR@doingapar}%
4298 }% nestspan
4299 \LWR@traceinfo{\LWR@startpars: done}%
4300 }
```

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

```
4301 \newcommand*{\LWR@stoppars}%
4302 {%
```

Ignore if inside a span:

```
4303 \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
4304 {}%
4305 {%
```

See if currently handling HTML paragraphs:

```
4306 \ifbool{LWR@doingapar}%
```

if currently in an HTML paragraph:

```
4307 {%
```

Print a closing tag:

```
4308 \unskip%
4309 \LWR@htmltagc{/LWR@tagregularparagraph}%
4310 \LWR@orignewline%
```

No longer have an open HTML paragraph:

```
4311 \global\boolfalse{LWR@doingapar}%
```

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

```
4312 \global\boolfalse{LWR@minipagethispar}
4313
4314 }% an intentionally blank line
```

If was not in an HTML paragraph:

```
4315 {}%
```

See if currently allowing HTML paragraphs:

```
4316 \ifbool{LWR@doingstartpars}%
```

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

```
4317 {\ClearPreHook{par}}%
```

Else: do nothing

```
4318    {}%
```

no longer in paragraph mode

```
4319    \global\setbool{LWR@doingstartpars}{false}%
```

no <p> tag to undo:

```
4320    \global\boolfalse{LWR@doingapar}%
```

```
4321 }% nestspan
```

```
4322 }
```

```
4323 \end{warpHTML}
```

## 48 Page headers and footers

**for HTML & PRINT:** 4324 \begin{warpall}

In the following, catcode is manually changed back and forth without groups, since new macros are being defined which must not be contained within the groups.

```
4325 \newcommand{\LWR@firstpagetop}{} % for the home page alone
```

```
4326 \newcommand{\LWR@pagetop}{} % for all other pages
```

```
4327 \newcommand{\LWR@pagebottom}{}%
```

\HTMLFirstPageTop {<text and logos>}

```
4328 \newcommand{\HTMLFirstPageTop}[1]{%
```

```
4329    \renewcommand{\LWR@firstpagetop}{#1}%
```

```
4330 }
```

\HTMLPageTop {<text and logos>}

```
4331 \newcommand{\HTMLPageTop}[1]{%
```

```
4332    \renewcommand{\LWR@pagetop}{#1}%
```

```
4333 }
```

\HTMLPageBottom {<text and logos>}

```
4334 \newcommand{\HTMLPageBottom}[1]{%
```

```
4335    \renewcommand{\LWR@pagebottom}{#1}%
```

```
4336 }
```

```
4337 \end{warpall}
```

## 49 CSS

**for HTML output:** 4338 \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.

```
4339 \newcommand*{\LWR@currentcss}{lwarp.css}
```

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

```
4340 \newcommand*{\CSSFilename}[1]{%
4341 \renewcommand*{\LWR@currentcss}{#1}%
4342 \@onelevel@sanitize\LWR@currentcss%
4343 }
4344
4345 \end{warpHTML}
```

**for PRINT output:** 4346 \begin{warpprint}  
4347 \newcommand\*{\CSSFilename}[1]{}  
4348 \end{warpprint}

## 50 Title, HTML meta author, HTML meta description

**for HTML output:** 4349 \begin{warpHTML}

**\title** `{\langle title \rangle}` Modified to remember \thetitle, which is used to set the HTML page titles.

```
4350 \let\LWR@origtitle\title
4351
4352 \renewcommand*{\title}[1]{%
4353   \LWR@origtitle{#1}%
4354   \begingroup%
4355     \renewcommand{\thanks}[1]{}%
4356     \protected@xdef\thetitle{#1}%
4357   \endgroup%
4358 }
```

```
4359 \end{warpHTML}
```

**for HTML & PRINT:** 4360 \begin{warpall}

`\HTMLTitle`  $\{\langle Titlename \rangle\}$  The Title to place into an HTML meta tag. The default is to use the document `\title`'s setting.

```
4361 \providecommand{\thetitle}{}
4362
4363 \newcommand{\theHTMLTitle}{\thetitle}
4364
4365 \newcommand{\HTMLTitle}[1]{\renewcommand{\theHTMLTitle}{#1}}
```

`\HTMLAuthor`  $\{\langle authorname \rangle\}$  The author to place into an HTML meta tag. If none given, the default is `\theauthor`, which is empty unless the titling package is used.

```
4366 \providecommand{\theauthor}{}
4367
4368 \newcommand{\theHTMLAuthor}{\theauthor}
4369
4370 \newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}
```

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



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


`\HTMLDescription`  $\{\langle New HTML meta description. \rangle\}$  Assigns the HTML file's description meta tag.

```
4371 \newcommand{\LWR@currentHTMLDescription}{}
4372
4373 \newcommand{\HTMLDescription}[1]{%
4374 \renewcommand{\LWR@currentHTMLDescription}{#1}
4375 }
4376
4377 \end{warpall}
```

## 51 Footnotes

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



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

Several kinds of footnotes are used: in a regular page, in a minipage, or as thanks in the titlepage. Each of these is handle differently.

### 51.1 Regular page footnotes

In HTML documents, footnotes are placed at the bottom of the web page using the  $\text{\LaTeX}$  box `\LWR@footnotes`. Using this instead of the original `\footins` box avoids having footnotes be printed by the output routine, since footnotes should be printed per HTML page instead of per PDF page.

See section 51.4 for the implementation.

### 51.2 Minipage footnotes

See section 51.5 for how minipage footnotes are gathered. See section 80.3 for how minipage footnotes are placed into the document.

### 51.3 Titlepage thanks

See section 58.7 for titlepage footnotes.

### 51.4 Regular page footnote implementation

**for HTML & PRINT:** 4378 `\begin{warpall}`

**Ctrl** `FootnoteDepth` Determines how deeply to place footnotes in the HTML files, similar to `tocdepth`.  
**Default:** 5 The default of 5 places footnotes before each `\subparagraph` or higher. See table 7 for a table of  $\text{\LaTeX}$  section headings.

```
4379 \newcounter{FootnoteDepth}
4380 \setcounter{FootnoteDepth}{5}
```

```
4381 \end{warpall}
```

**for HTML output:** 4382 `\begin{warpHTML}`

Patch  $\LaTeX$  footnotes to use a new `\box` instead of an insert for lwarp footnotes. This avoids having the original `\footins` appear at the bottom of a `lateximage`, which is on its own new page.

```
4383 \newbox\LWR@footnotes
```

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

```
\@makefntext  {\langle text\rangle}
```

```
4384 \long\def\@makefntext#1{\textsuperscript{\@thefnmark}~#1}
```

```
\@makefnmark
```

```
4385 \def\@makefnmark{\hbox{\textsuperscript{\@thefnmark}}}
```

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

In a `lateximage` during HTML output, the `lateximage` is placed inside a print-mode `minipage`, but the footnotes are broken out by:

```
\def\@mpfn{footnote}
\def\thempfn{\thefootnote}
\let\@footnotetext\LWR@footnotetext
```

```
\LWR@footnotetext  {\langle text\rangle}
```

```
4386 \long\def\LWR@footnotetext#1{%
4387 \LWR@traceinfo{\LWR@footnotetext}%
4388 \global\setbox\LWR@footnotes=\vbox{%
```

Add to any current footnotes:

```
4389 \unvbox\LWR@footnotes%
```

Remember the footnote number for `\ref`:

```
4390 \protected@edef\@currentlabel{%
4391 \csname p@footnote\endcsname\@thefnmark%
4392 }% \@currentlabel
```

Open a group:

```
4393 \color@begingroup%
```

Use HTML superscripts even inside a lateximage:

```
4394 \renewrobustcmd{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

Use paragraph tags if in a tabular data cell or a lateximage:

```
4395 \ifthenelse{%
4396   \boolean{LWR@doingstartpars} \AND%
4397   \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
4398 }%
4399 {}%
4400 {\LWR@htmltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
4401 \@makefntext{#1}%
```

Closing paragraph tag:

```
4402 \ifthenelse{%
4403   \boolean{LWR@doingstartpars} \AND%
4404   \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
4405 }%
4406 {\par}%
4407 {%
4408   \LWR@htmltagc{/LWR@tagregularparagraph}%
4409   \LWR@orignewline%
4410 }%
```

Close the group:

```
4411 \color@endgroup%
4412 }% vbox
4413 }%
```

```
\@footnotetext {\text}
```

```
4414 \let\@footnotetext\LWR@footnotetext
```

## 51.5 Minipage footnote implementation

Patch  $\LaTeX$  minipage footnotes to use a new `\box` instead of an insert for lwarp minipage footnotes. This avoids having the original `\@mpfootins` appear at the bottom of a lateximage, which is on its own new page.

```
4415 \newbox\LWR@mpfootnotes
```

`\@mpfootnotetext` `{\text}`

```

4416 \long\def\@mpfootnotetext#1{%
4417 \LWR@traceinfo{\@mpfootnotetext}%
4418 \global\setbox\LWR@mpfootnotes\vbox{%
4419   \unvbox\LWR@mpfootnotes
4420   \reset@font\footnotesize
4421   \hsize\columnwidth
4422   \@parboxrestore
4423   \protected@edef\@currentlabel
4424   {\csname p@mpfootnote\endcsname\@thefnmark}%
4425   \color@begingroup

```

Use paragraph tags if in a tabular data cell or a lateximage:

```

4426   \ifthenelse{%
4427     \boolean{LWR@doingstartpars} \AND%
4428     \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
4429   }%
4430   {}%
4431   {\LWR@htmltagc{\LWR@tagregularparagraph}}%

4432   \@makefnfntext{%
4433     \ignorespaces#1%
4434   }%

```

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

```

4435   \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
4436   {}%
4437   {%
4438     \LWR@htmltagc{/\LWR@tagregularparagraph}%
4439     \LWR@orignewline%
4440   }%
4441   \color@endgroup%
4442 }% vbox
4443 \LWR@traceinfo{\@mpfootnotetext: done}%
4444 }

```

`\thempfootnote` Redefined to remove the `\itshape`, which caused an obscure compiling error in some situations.

```

4445 \AtBeginDocument{
4446 \def\thempfootnote{\@alph{c@mpfootnote}}
4447 }

```

## 51.6 Printing pending footnotes

`\LWR@printpendingfootnotes` Enclose the footnotes in a class, print, then clear.

```

4448 \newcommand*{\LWR@printpendingfootnotes}{%
4449 \ifvoid\LWR@footnotes\else
4450   \LWR@forcenewpage
4451   \begin{BlockClass}{footnotes}
4452   \LWR@origmedskip
4453   \unvbox\LWR@footnotes
4454   \setbox\LWR@footnotes=\vbox{}
4455   \end{BlockClass}
4456 \fi
4457 }
```

`\LWR@maybeprintpendingfootnotes` `{<depth>}` Used to print footnotes before sections only if formatting for an EPUB or word processor:

```

4458 \newcommand*{\LWR@maybeprintpendingfootnotes}[1]{%
4459 \ifboolexpr{
4460   not test{ifnumcomp{#1}{>}{\value{FootnoteDepth}}} or
4461   bool{FormatEPUB} or
4462   bool{FormatWP}
4463 }%
4464 {\LWR@printpendingfootnotes}%
4465 {}%
4466 }
```

`\LWR@printpendingmpfootnotes` Enclose the minipage footnotes in a class, print, then clear.

```

4467 \newcommand*{\LWR@printpendingmpfootnotes}{%
4468 \ifvoid\LWR@mpfootnotes\else
4469   \LWR@forcenewpage
4470   \begin{BlockClass}{footnotes}
4471   \LWR@origvspace*{\baselineskip}
4472   \unvbox\LWR@mpfootnotes
4473   \setbox\LWR@mpfootnotes=\vbox{}
4474   \end{BlockClass}
4475 \fi
4476 }

4477 \end{warpHTML}
```

## 52 Marginpars

`\marginpar` [*left*] [*right*] `\marginpar` may contains paragraphs, but in order to remain inline with the surrounding text `lwarp` nullifies block-related macros inside the `\marginpar`. Paragraph breaks are converted to `<br />` tags.

`\marginparBlock` [*left*] [*right*] To include block-related macros, use `\marginparBlock`, which takes the same arguments but creates a `<div>` instead of a `<span>`. A line break will occur in the text where the `\marginBlock` occurs.

for HTML output: 4478 `\begin{warphTML}`

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

```
4479 \renewcommand{\marginpar}[2] [] {%
4480 \ifbool{FormatWP}%
4481 {%
4482 \begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{\marginblock}
4483 #2
4484 \end{LWR@BlockClassWP}
4485 }%
4486 {%
4487 \LWR@htmlspanclass{marginpar}{#2}%
4488 }%
4489 }
```

`\marginparBlock` [*left*] [*right*]

For use when the `marginpar` will be more than one paragraph, and/or contains more than simple text.

HTML version.

```
4490 \newcommand{\marginparBlock}[2] [] {%
4491 \ifbool{FormatWP}%
4492 {%
4493 \begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{\marginblock}
4494 #2
4495 \end{LWR@BlockClassWP}
4496 }%
4497 {%
4498 \begin{BlockClass}[width:2in; float:right; margin:10pt]{marginparblock}
4499 #2
4500 \end{BlockClass}
4501 }%
4502 }
```

`\reversemarginpar`

```
4503 \renewcommand*{\reversemarginpar}{}

```

`\normalmarginpar`

```
4504 \renewcommand*{\normalmarginpar}{}

```

```
4505 \end{warpHTML}

```

**for PRINT output:** `4506 \begin{warpprint}`

`\marginparBlock` [*left*] [*right*]

For use when the marginpar will be more than one paragraph, and/or contains more than simple text.

Print version.

```
4507 \LetLtxMacro\marginparBlock\marginpar

```

```
4508 \end{warpprint}

```

## 53 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:** `4509 \begin{warppall}`

**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.

```
4510 \newcounter{FileDepth}

```

```
4511 \setcounter{FileDepth}{-5}

```

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

```
4512 \newbool{CombineHigherDepths}

```

```
4513 \booltrue{CombineHigherDepths}

```

```
4514 \end{warpall}
```

**for HTML output:** 4515 \begin{warpHTML}

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

```
4516 \newcommand*{\LWR@thisfilename}{}
```

`\LWR@thisnewfilename` The filename being sanitized.

```
4517 \newcommand*{\LWR@thisnewfilename}{}
```

`\LWR@filenamenoblanks` `{\langle filename \rangle}`

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



Be sure that this does not result in filename collisions! Use the optional TOC caption entry parameter for formatting. Remember to `\protect`  $\TeX$  commands which appear in section names and TOC captions.

```
4518 \newcommand*{\LWR@filenamenoblanks}[1]{%
```

```
4519 \begingroup
```

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

```
4520 \LWR@nullfonts%
```

```
4521 \renewcommand*{\LWR@htmltag}[1]{}%
```

Replaces common symbols and short words with hyphens:

```
4522 % \LWR@traceinfo{\LWR@filenamenoblanks \#1: !\#1!}%
```

```
4523 \edef\LWR@thisnewfilename{\#1}%
```

```
4524 \LWR@traceinfo{\LWR@filenamenoblanks \edef: !\LWR@thisnewfilename!}%
```

```
4525 \fullexpandarg%
```

Convert spaces into hyphens:

```
4526 \StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctuation into hyphens:

```
4527 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
```

```
4528 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
```

```
4529 \StrSubstitute{\LWR@thisnewfilename}{%}
```

```
4530 {\LWR@origampersand}{-}[\LWR@thisnewfilename]
```



```

4531 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
4532 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
4533 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
4534 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
4535 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
4536 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
4537 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
4538 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
4539 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
4540 \StrSubstitute{\LWR@thisnewfilename}%
4541     {\textless}{-}[\LWR@thisnewfilename]
4542 \StrSubstitute{\LWR@thisnewfilename}%
4543     {\textgreater}{-}[\LWR@thisnewfilename]
4544 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]

4545 \StrSubstitute{\LWR@thisnewfilename}{\_}{-}[\LWR@thisnewfilename]

4546 \StrSubstitute{\LWR@thisnewfilename}{\ }{-}[\LWR@thisnewfilename]
4547 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
4548 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
4549 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
4550 \StrSubstitute{\LWR@thisnewfilename}{\|}{-}[\LWR@thisnewfilename]
4551 \StrSubstitute{\LWR@thisnewfilename}%
4552     {\textbackslash}{-}[\LWR@thisnewfilename]
4553 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
4554 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
4555 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
4556 %     "~{}" for babel
4557 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
4558 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
4559 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]

```

Convert short words:

```

4560 \StrSubstitute{\LWR@thisnewfilename}{-s}{-}[\LWR@thisnewfilename]
4561 \StrSubstitute{\LWR@thisnewfilename}{-S}{-}[\LWR@thisnewfilename]
4562 \StrSubstitute{\LWR@thisnewfilename}{-a}{-}[\LWR@thisnewfilename]
4563 \StrSubstitute{\LWR@thisnewfilename}{-A}{-}[\LWR@thisnewfilename]
4564 \StrSubstitute{\LWR@thisnewfilename}{-an}{-}[\LWR@thisnewfilename]
4565 \StrSubstitute{\LWR@thisnewfilename}{-AN}{-}[\LWR@thisnewfilename]
4566 \StrSubstitute{\LWR@thisnewfilename}{-to}{-}[\LWR@thisnewfilename]
4567 \StrSubstitute{\LWR@thisnewfilename}{-TO}{-}[\LWR@thisnewfilename]
4568 \StrSubstitute{\LWR@thisnewfilename}{-by}{-}[\LWR@thisnewfilename]
4569 \StrSubstitute{\LWR@thisnewfilename}{-BY}{-}[\LWR@thisnewfilename]
4570 \StrSubstitute{\LWR@thisnewfilename}{-of}{-}[\LWR@thisnewfilename]
4571 \StrSubstitute{\LWR@thisnewfilename}{-OF}{-}[\LWR@thisnewfilename]
4572 \StrSubstitute{\LWR@thisnewfilename}{-and}{-}[\LWR@thisnewfilename]
4573 \StrSubstitute{\LWR@thisnewfilename}{-AND}{-}[\LWR@thisnewfilename]

```

```

4574 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
4575 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
4576 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
4577 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

4578 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
4579 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
4580 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
4581 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
4582 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
4583 %      emdash
4584 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
4585 %      endash
4586 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
4587 \endgroup
4588 }

```

**Ctr** Remembers which autopage label was most recently generated. Used to avoid duplicates.

**LWR@previousautopagelabel**

```

4589 \newcounter{LWR@previousautopagelabel}
4590 \setcounter{LWR@previousautopagelabel}{-1}

```

**\LWR@newautopagelabel** *{\pagenumber counter}*

```

4591 \newcommand*{\LWR@newautopagelabel}[1]{%
4592 \ifnumequal{\value{LWR@previousautopagelabel}}{\value{page}}{%
4593 }{% no action if this autopage label has already been defined
4594 {%
4595   \label{autopage-\arabic{#1}}%
4596   \setcounter{LWR@previousautopagelabel}{\value{page}}
4597 }%
4598 }

```

**\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.

```

4599 \newcommand*{\LWR@newhtmlfile}[1]{
4600 \LWR@traceinfo{\LWR@newhtmlfile}

```

At the bottom of the ending file:

```
4601 \LWR@htmlelementclassend{section}{textbody}
4602
4603 \LWR@printpendingfootnotes
4604
```

No footer between files if EPUB:

```
4605 \ifbool{FormatEpub}{
4606 {}
4607 {
4608     \LWR@htmlelement{footer}
4609
4610     \LWR@pagebottom
4611
4612     \LWR@htmlelementend{footer}
4613 }
```

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

```
4614 \ifthenelse{\boolean{FormatEpub}\OR\boolean{FormatWP}}{
4615 {}
4616 {\ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}}
```

End of this HTML file:

```
4617 \LWR@stoppars
4618 \LWR@htmltag{/body}\LWR@orignewline
4619 \LWR@htmltag{/html}\LWR@orignewline
4620 \LWR@orignewpage
4621
4622 \addtocounter{LWR@htmlfilenumber}{1}%
```

If using a filename, create a version without blanks. The filename without blanks will be placed into \LWR@thisfilename. If not using a filename, the file number will be used instead.

```
4623 \ifbool{FileSectionNames}%
4624 {\LWR@filenamenoblanks{#1}}
4625 {\renewcommand*\LWR@thisfilename{\arabic{LWR@htmlfilenumber}}}
```

Include an HTML comment to instruct lwarpmk where to split the files apart. Uses pipe-separated fields for `split_html.gawk`. Uses monospaced font with ligatures disabled for everything except the title.

```
4626 \LWR@traceinfo{LWR@newhtmlfile: about to print start file}%
```

\LWR@nullfonts to allow math in a section name.

```
4627 \begingroup%
4628 \LWR@nullfonts%
4629 \LWR@htmlblockcomment{%
4630 |Start file|}%
4631 \LWR@htmlsectionfilename{\LWR@thisfilename}|}%
4632 }
4633 \endgroup%
```

At the top of the starting file:

```
4634 \LWR@stoppars
4635

4636 \LWR@filestart{ -- #1}% there is an Emdash in front of the #1
4637
```

Track the page numbers:

```
4638 \setcounter{LWR@latestautopage}{\value{page}}%
4639 \LWR@newautopagelabel{LWR@latestautopage}%
```

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

```
4640 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4641 {}
4642 {\LWR@topnavigation}
4643
```

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

```
4644 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4645 {}
4646 {
4647   \LWR@htmlelement{header}
4648
4649   \LWR@pagetop
4650
4651   \LWR@htmlelementend{header}
4652 }
4653
```

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

```
4654 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4655 {}
4656 {\ifcvoid{thetitle}{}\{\LWR@printthetitle}}
```

4657

No sidetoc if formatting for an EPUB or word processor:

```
4658 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4659 {}
4660 {\LWR@sidetoc}
4661
```

Start of the <tbody>:

```
4662 \LWR@htmlclass{section}{tbody}
4663
```

Keep paragraph tags disabled for now:

```
4664 \LWR@stoppars
4665
```

```
4666 \LWR@traceinfo{\LWR@newhtmlfile: done}
4667 }
```

```
4668 \end{warpHTML}
```

## 54 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing  $\TeX$  code and packages. Formatting is handled by CSS, so the emulated code has much less work to do than the print versions.

**Unicode** Section names and the resulting filenames with accented characters are partially supported, depending on the ability of  $\text{pdflatex}$  to generate characters and  $\text{pdftotext}$  to read them. If extra symbols appear in the text, it may be that  $\text{pdflatex}$  is actually producing a symbol over or under a character, resulting in  $\text{pdftotext}$  picking up the accent symbol separately.



$\text{Xe}\mathbb{L}\mathbb{A}\mathbb{T}\mathbb{E}\mathbb{X}$  and  $\text{Lua}\mathbb{L}\mathbb{A}\mathbb{T}\mathbb{E}\mathbb{X}$  directly support accented section and file names.

**for HTML output:** 4669 \begin{warpHTML}

## 54.1 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.

```
4670 \newbool{LWR@forcinghtmlpage}
4671 \boolfalse{LWR@forcinghtmlpage}
4672
4673 \newcommand*{\ForceHTMLPage}{%
4674 \global\booltrue{LWR@forcinghtmlpage}%
4675 }
```

`\ForceHTMLTOC` For HTML output, forces the next section to have a TOC entry, even if starred. For use with `\printindex` and others which generate a starred section which should be in the TOC so that it may be accessed via HTML. Not necessary if used with `tocbibind`. Also see `\ForceHTMLPage`.

For print output, no effect.

```
4676 \newbool{LWR@forcinghtmltoc}
4677 \boolfalse{LWR@forcinghtmltoc}
4678
4679 \newcommand*{\ForceHTMLTOC}{%
4680 \global\booltrue{LWR@forcinghtmltoc}%
4681 }
```

```
4682 \end{warpHTML}
```


**for PRINT output:**

```
4683 \begin{warpprint}
4684 \newcommand*{\ForceHTMLPage}{}
4685 \newcommand*{\ForceHTMLTOC}{}
4686 \end{warpprint}
```

**for HTML output:**

```
4687 \begin{warpHTML}
```

## 54.2 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.

```
4688 \newbool{LWR@mainmatter}
```

```

4689 \DeclareDocumentCommand{\mainmatter}{-}{-}{%
4690 \booltrue{LWR@mainmatter}%
4691 }

```

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

```

4692 \DeclareDocumentCommand{\frontmatter}{-}{-}{%
4693 \boolfalse{LWR@mainmatter}%
4694 }

```

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

```

4695 \DeclareDocumentCommand{\backmatter}{-}{-}{%
4696 \boolfalse{LWR@mainmatter}
4697 }

```

### 54.3 Sectioning support macros

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

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

```

4698 \newcommand*{\LWR@sectionnumber}[1]{%
4699 \InlineClass{sectionnumber}{#1}%
4700 }

```

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

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

Create an autosection tag.

```

4701 \newcommand*{\LWR@createautosec}[1]{%
4702 \LWR@htmltag{#1 id="\LWR@origmbox{autosec-\arabic{page}}"}%
4703 }

```

`\LWR@pushoneclose`  $\{\langle depth\rangle\}$   $\{\langle printclose\rangle\}$  Stacks the new sectioning level's closing tag, to be used when this section is closed some time later.



`\LWR@stoppars` must be executed first.

```

4704 \NewDocumentCommand{\LWR@pushoneclose}{m m}{%

```

```

4705 \LWR@traceinfo{LWR@pushoneclose #1}%
4706     \pushclose{#2}{#1}%
4707 }

```

`\LWR@startnewdepth`  $\{\langle depth \rangle\}$   $\{\langle printclose \rangle\}$

Closes currently stacked tags of a lesser level, then opens the new nesting level by saving this new sectioning level's closing tag for later use.



`\LWR@stoppars` must be executed first.

```

4708 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%

```

Close any stacked sections up to this new one.

```

4709 \LWR@closeprevious{#1}%

```

Push a new section depth:

```

4710 \LWR@pushoneclose{#1}{#2}%
4711 }

```

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.

```

4712 \newcounter{LWR@prevFileDepth}
4713 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}

```

`\@secCNTformat`  $\{\langle sectiontype \rangle\}$

```

4714 \def\@secCNTformat#1{\csname the#1\endcsname\protect\quad}

```

`\simplechapterdelim` Used by `tocbibind` and `anonchap`.

```

4715 \newcommand*{\simplechapterdelim}{\}

```

`\@chapCNTformat`  $\{\langle sectiontype \rangle\}$

`\let` to `\@secCNTformat` by default, but may be redefined by `\simplechapter` and `\restorechapter` from `tocbibind` or `anonchap`.

```

4716 \let\@chapCNTformat\@secCNTformat

```



Ctr LWR@currentautosec Records the page number when the section was created. If a math expression is included in the section name, and SVG math is used, the corresponding `lateximage` will cause the page number to change by the time the following autosec label is created.

```
4717 \newcounter{LWR@currentautosec}
```

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

The common actions for the high-level sectioning commands.

```
4718 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
4719 \LWR@traceinfo{LWR@section |#2| |#3|}%
4720 \LWR@traceinfo{LWR@section: not an empty section}%
4721 \LWR@stoppars%
```

Cancel special minipage horizontal space interaction:

```
4722 \global\boolfalse{LWR@minipagethispar}%
```

Start a new HTML file unless starred, and if is a shallow sectioning depth.

Exception: Also start a new HTML file for `\part*`, for appendix.

Generate a new  $\LaTeX$  page so that toc and index page number points to the section:

```
4723 \LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
4724 \IfBooleanT{#1}{\LWR@traceinfo{LWR@section: starred}}}%
4725 \ifbool{LWR@forcinghtmlpage}{\LWR@traceinfo{LWR@section: forcinghtmlpage}}{}}}%
4726 \ifthenelse{%
4727   \(%
4728     \(\NOT\equal{#1}{\BooleanTrue}\)\OR%
4729     \(\cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}\)\OR%
4730     \(\boolean{LWR@forcinghtmlpage}\)\)%
4731   \)%
4732   \AND%
4733   \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}}%
4734   \AND%
4735   \(%
4736     \NOT\boolean{CombineHigherDepths}\OR%
4737     \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}}%
4738   \)%
4739   \AND%

4740   \(% phantomsection
4741     \NOT\isempty{#3}%
4742     \OR%
4743     \(\NOT\equal{#1}{\BooleanTrue}\)\)%
```

```
4744    \)%
4745 }%
```

If so: start a new HTML file:

```
4746 {% new file
4747    \LWR@traceinfo{LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
4748    \IfNoValueTF{#2}%
```

If no optional entry

```
4749    {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
4750    {\LWR@newhtmlfile{#2}}%
4751 }% new file
```

Else: No new HTML file:

```
4752 {% not new file
```

Generate a new  $\TeX$  page so that toc and index page number points to the section:

```
4753    \LWR@traceinfo{LWR@section: not a new HTML file, about to LWR@orignewpage}%
4754    \LWR@orignewpage%
4755
4756 }% not new file
```

Remember this section's name for \nameref:

```
4757 \IfValueT{#3}{%
4758 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
4759 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
4760 }%
```

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

```
4761
4762 \ifbool{HTMLDebugComments}{%
4763    \begingroup%
4764    \LWR@nullfonts%
4765    \LWR@htmlcomment{Opening #4 ‘‘#3’’}%
4766    \endgroup%
```

```
4767 }{}%
4768
```

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

```
4769 \ifthenelse{%
4770     \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
4771 }%
4772 {\LWR@startpars}%
4773 {}%
```

Create the opening tag with an autosec:

```
4774 \LWR@traceinfo{LWR@section: about to LWR@createautosec}%
4775 \LWR@createautosec{\csuse{LWR@tag#4}}%
```

```
4776 \setcounter{LWR@currentautosec}{\value{page}}
```

Check if starred:

```
4777 \IfBooleanTF{#1}%
4778 {%
4779 \LWR@traceinfo{LWR@section: starred}%
```

Starred, but also forcing a TOC entry, so add unnumbered TOC name or regular name:

```
4780 \ifbool{LWR@forcinghtmltoc}%
4781 {\addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}}%
4782 {}%
4783 }% starred
```

Not starred, so step counter and add to TOC:

```
4784 {% not starred
```

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

```
4785     \ifthenelse{%
4786         \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
4787     }%
4788     {% if secnumdepth
```

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

```
4789         \LWR@traceinfo{LWR@section: about to test main matter}%
```

```

4790      \ifbool{LWR@mainmatter}%
4791      {%
4792          \LWR@traceinfo{LWR@section: yes mainmatter}%
4793          \refstepcounter{#4}%

```

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

```

4794          \LWR@traceinfo{LWR@section: about to addcontentsline}%
4795          \addcontentsline{toc}{#4}%
4796          {%
4797              \protect\numberline{\csuse{the#4}}%
4798              {\ignorespaces\IfValueTF{#2}{#2}{#3}\protect\relax}%
4799          }%
4800          \LWR@traceinfo{LWR@section: finished addcontentsline}%
4801      }% end of if main matter

```

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

```

4802      {% not main matter
4803          \LWR@traceinfo{LWR@section: no main matter}%
4804          \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4805      }% end of not main matter
4806  }% end of secnumdepth

```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```

4807      {%
4808          \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4809      }%

```

For part, print the section type:

```

4810      \ifbool{LWR@mainmatter}%
4811      {%
4812          \ifthenelse{%
4813              \(<\cntttest{\csuse{LWR@depth#4}}{<=}%
4814              {\value{secnumdepth}}\>) \AND%
4815              \(<\cntttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\>)%
4816          }%
4817          {\csuse{#4name}~{}}%
4818          {}%

```

Print the section number:

```

4819          \LWR@traceinfo{LWR@section: about to print section number}%
4820          \ifthenelse{%
4821              \cntttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
4822          }%

```

```

4823      {%
4824          \ifstrequal{#4}{chapter}%
4825          {\protect\LWR@sectionnumber{\@chapcntformat{#4}}}%
4826          {\protect\LWR@sectionnumber{\@seccntformat{#4}}}%
4827      }%
4828      {}%
4829      \LWR@traceinfo{LWR@section: finished print section number}%
4830      }{}%
4831 }% end of not starred

```

Print the section name:

```

4832 \LWR@traceinfo{LWR@section: about to print the section name}%
4833 #3%

```

Close the heading tag, such as /H2:

```

4834 \LWR@traceinfo{LWR@section: about to close the heading tag}%
4835 \LWR@htmltag{\csuse{LWR>tag#4end}}%

```

Generate a  $\LaTeX$  label:

```

4836 \LWR@traceinfo{LWR@section: about to create the LaTeX label}%
4837 \LWR@newautopagelabel{LWR@currentautosec}%

```

Start paragraph handing unless is an inline paragraph or subparagraph:

```

4838 \ifthenelse{%
4839     \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}%
4840 }%
4841 {\LWR@startpars}%
4842 {}%

```

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.

```

4843 \ifthenelse{%
4844     \NOT\equal{#1}{\BooleanTrue}\OR%
4845     \cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}%
4846 }%
4847 {% not starred

```

```

4848 \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
4849 }% not starred
4850 {}%

```

Reset to defaults if not a phantomsection:

```

4851 \ifstrempy{#3}%
4852 {}%
4853 {%
4854 \global\boolfalse{LWR@forcinghtmlpage}%
4855 \global\boolfalse{LWR@forcinghtmltoc}%
4856 }%
4857 %
4858 \LWR@traceinfo{LWR@section: done}%
4859 }

```

## 54.4 \section and friends

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

```

4860 \newcommand{\part@preamble}{}% for koma-script
4861
4862 \DeclareDocumentCommand{\part}{s o m}{%
4863 \LWR@maybeprintpendingfootnotes{\LWR@depthpart}%
4864 \LWR@stoppars%
4865
4866 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
4867
4868 \LWR@section{#1}{#2}{#3}{part}%
4869
4870 \part@preamble% for koma-script
4871 \renewcommand{\part@preamble}{}%
4872 }

```

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

```

4873 \let\@printcites\relax% for quotchap package
4874
4875 \newcommand{\chapter@preamble}{}% for koma-script
4876
4877 \@ifundefined{chapter}
4878 {}
4879 {%
4880 \DeclareDocumentCommand{\chapter}{s o o m}{%
4881 \IfValueTF{#2}{

```

```

4882 \LWR@traceinfo{chapter #2}%
4883 }{
4884 \LWR@traceinfo{chapter #4}%
4885 }
4886 \LWR@maybeprintpendingfootnotes{\LWR@depthchapter}%
4887 \LWR@stoppars%
4888
4889 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
4890
4891 \LWR@section{#1}{#2}{#4}{chapter}%
4892
4893 \@printcites% for quotchap package
4894
4895 \chapter@preamble% for koma-script
4896 \renewcommand{\chapter@preamble}{}%
4897 }
4898 }

```

`\section * [(TOC name)] [(heading name)] {(name)}`

```

4899 \DeclareDocumentCommand{\section}{s o m}{%
4900 \IfValueTF{#2}{
4901 \LWR@traceinfo{section #2}%
4902 }{
4903 \LWR@traceinfo{section #4}%
4904 }
4905 \LWR@maybeprintpendingfootnotes{\LWR@depthsection}%
4906 \LWR@stoppars%
4907
4908 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
4909
4910 \LWR@section{#1}{#2}{#4}{section}%
4911 }

```

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

```

4912 \DeclareDocumentCommand{\subsection}{s o m}{%
4913 \LWR@maybeprintpendingfootnotes{\LWR@depthsubsection}%
4914 \LWR@stoppars%
4915
4916 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
4917
4918 \LWR@section{#1}{#2}{#3}{subsection}%
4919 }

```

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

```

4920 \DeclareDocumentCommand{\subsubsection}{s o m}{%
4921 \LWR@maybeprintpendingfootnotes{\LWR@depthsubsubsection}%
4922 \LWR@stoppars%
4923
4924 \LWR@startnewdepth{\LWR@depthsubsubsection}%
4925 {\LWR@printclosesubsubsection}%
4926
4927 \LWR@section{#1}{#2}{#3}{subsubsection}%
4928 }

```

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

```

4929 \DeclareDocumentCommand{\paragraph}{s o m}{%
4930 \LWR@maybeprintpendingfootnotes{\LWR@depthparagraph}%
4931 \LWR@stoppars%
4932
4933 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
4934
4935 \LWR@section{#1}{#2}{#3}{paragraph}%
4936 }

```

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

```

4937 \DeclareDocumentCommand{\subparagraph}{s o m}{%
4938 \LWR@maybeprintpendingfootnotes{\LWR@depthsubparagraph}%
4939 \LWR@stoppars%
4940
4941 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
4942
4943 \LWR@section{#1}{#2}{#3}{subparagraph}%
4944 }

4945 \end{warpHTML}

```

## 55 Starting a new file

**for HTML & PRINT:** 4946 \begin{warpall}

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

```

4947 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
4948
4949 \newcommand*{\HTMLLanguage}[1]{%
4950 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
4951 }

```



```
4952 \end{warpall}
```

**for HTML output:** 4953 \begin{warpHTML}

```
\LWR@filestart {\title_suffix}
```

Creates the opening HTML tags.

```
4954 \newcommand*{\LWR@filestart}[1]{
4955 \LWR@traceinfo{\LWR@filestart}
```

Locally temporarily disable direct-formatting commands:

```
4956 \begingroup
4957 \LWR@nullfonts
```

Create the page's HTML header:

```
4958 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
```

The language is user-adjustable:

```
4959 \LWR@htmltag{html lang="\LWR@currentHTMLLanguage"}\LWR@orignewline
```

Start of the meta data:

```
4960 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
4961 \LWR@htmltag{meta charset="UTF-8" /\LWR@orignewline
```

Author:

```
4962 \ifthenelse{\equal{\theHTMLAuthor}{}}{%
4963 {}%
4964 {\LWR@htmltag{meta name="author" content="\theHTMLAuthor" /\LWR@orignewline}}%
```

lwarp is the generator:

```
4965 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /\LWR@orignewline
4966 \LWR@orignewline
```

If there is a description, add it now:

```
4967 \ifdefempty{\LWR@currentHTMLDescription}{}{%
```

```

4968 \LWR@htmltag{%
4969 meta name="description" content="\LWR@currentHTMLDescription" /}%
4970 \LWR@orignewline
4971 }%

```

Mobile-friendly viewport:

```

4972 \LWR@htmltag{meta name="viewport" %
4973 content="width=device-width, initial-scale=1.0" /}%
4974 \LWR@orignewline

```

IE patch:

```

4975 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
4976 \LWR@htmltag{%
4977 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"%
4978 \LWR@htmltag{/script}\LWR@orignewline
4979 \LWR@htmltag{![endif]{-}{-}}\LWR@orignewline

```

The page's title:

```

4980 \ifthenelse{equal{\theHTMLTitle}{}}%
4981 {}%
4982 {\LWR@htmltag{title}\theHTMLTitle#1\LWR@htmltag{/title}\LWR@orignewline}%

```

The page's stylesheet:

```

4983 \LWR@htmltag{%
4984 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
4985 \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.

```

4986 \ifbool{mathjax}%
4987 {%
4988 \begingroup%
4989 \LWR@restoreoriglists%
4990 \boolfalse{LWR@verbtags}
4991 \verbatiminput{lwarp_mathjax.txt}%
4992 \booltrue{LWR@verbtags}
4993 \endgroup%
4994 \LWR@stoppars
4995 }% end of mathjax
4996 {}%

```

End of the header:

```
4997 \LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
4998 \LWR@htmltag{body}\LWR@orignewline
4999 \endgroup
5000 \LWR@traceinfo{LWR@filestart: done}
5001 }
```

```
5002 \end{warpHTML}
```

## 56 Starting HTML output

**for HTML output:** 5003 \begin{warpHTML}

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

```
5004 \catcode'\$=\active
5005 \newcommand*{\LWR@LwarpStart}
5006 {%
5007 \LWR@traceinfo{LWR@lwarpStart}
```

If formatting for a word processor, force filedepth to single-file only, force HTML debug comments off.

```
5008 \ifbool{FormatWP}{%
5009   \setcounter{FileDepth}{-5}%
5010   \boolfalse{HTMLDebugComments}%
5011 }{}
```

Expand and detokenize `\HomeHTMLFilename` and `\HTMLFilename`:

```
5012 \edef\LWR@strresult{\HomeHTMLFilename}
5013 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
5014 \edef\LWR@strresult{\HTMLFilename}
5015 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn and empty page style:

```
5016 \LWR@origonecolumn%
5017 \LWR@origpagestyle{empty}%
```

Reduce chance of line overflow in verbatim environments:

```
5018 \LWR@origscriptsize%
```

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

```
5019 \LWR@origraggedright%
5020 \LetLtxMacro{\}\{\LWR@endofline}%
```

Spread the lines for pdftotext to read them well:

```
5021 \linespread{1.3}%
```

For pdftotext to reliably identify paragraph splits:

```
5022 \setlength{\parindent}{0pt}
5023 \setlength{\parskip}{2ex}
```

For the lateximages record file:

```
5024 \immediate\openout\LWR@lateximagesfile=lateximages.txt
```

Removes space around the caption in the HTML:

```
5025 \setlength{\belowcaptionskip}{0ex}
5026 \setlength{\abovecaptionskip}{0ex}
```

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

```
5027 \renewcommand{\ps@plain}{}%
```

<code>\centering</code> <code>\raggedleft</code> <code>\raggedright</code>	<p>Not used in the HTML environment:</p> <pre>5028 \renewcommand*{\centering}{} 5029 \renewcommand*{\raggedleft}{} 5030 \renewcommand*{\raggedright}{}%</pre>
--	---

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:

```
5031 \LetLtxMacro{\LWR@origtabular}{\tabular}
5032 \LetLtxMacro{\LWR@origendtabular}{\endtabular}
5033 \LetLtxMacro{\tabular}{\LWR@tabular}
5034 \LetLtxMacro{\endtabular}{\endLWR@tabular}
```

Float captions:

```
5035 \let\LWR@origcaption\caption
```

Labels: `\ltx@label` is used in `amsmath` environments and is also patched by `cleveref`.

Label in HTML

```
5036 \let\LWR@origltx@label\ltx@label
5037 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
5038 \let\TeX\LWR@TeX
5039 \let\LaTeX\LWR@LaTeX
5040 \let\LuaTeX\LWR@LuaTeX
5041 \let\LuaLaTeX\LWR@LuaLaTeX
5042 \let\XeTeX\LWR@XeTeX
5043 \let\XeLaTeX\LWR@XeLaTeX
5044 \let\ConTeXt\LWR@ConTeXt
```

Not yet started any paragraph handling:

```
5045 \global\boolfalse{LWR@doingapar}
5046 \global\boolfalse{LWR@doingstartpars}
```

Document and page settings:

```
5047 \mainmatter
5048 \LWR@origpagenumbering{arabic}
```

Start a new HTML file and a header:

```
5049 \LWR@traceinfo{LWR@lwarpStart: Starting new file.}
5050 \LWR@filestart{}
5051 \LWR@traceinfo{LWR@lwarpStart: Generating first header.}
5052 \LWR@htmltag{header}\LWR@orignewline
5053 \LWR@startpars
5054 \LWR@firstpagetop
5055 \LWR@stoppars
5056 \LWR@htmltag{/header}\LWR@orignewline
5057 \LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
5058 \LWR@htmltag{section class="textbody"}
```

Patch the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native  $\TeX$  environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```
5059 \LWR@patchlists
```

Ensure that math mode is active to call `lwarp`'s patches:

```
5060 \catcode'\$=\active
```

Required for `\nameref` to work with SVG math:

```
5061 \immediate\write\@mainaux{\catcode'\string$\active}%
5062 \LetLtxMacro\LWR@syntaxhighlightone$% balance for editor syntax highlighting
```

Allow HTML paragraphs to begin:

```
5063 \LWR@startpars
5064 \LWR@traceinfo{\LWR@lwarpStart: done}
5065 }
5066 \catcode'\$=3% math shift until lwarp starts

5067 \end{warpHTML}
```

## 57 Ending HTML output

**for HTML output:** 5068 `\begin{warpHTML}`

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

```
5069 \newcommand*{\LWR@requesttoc}[2]{%
5070 \ifbool{#1}
5071 {
5072     \expandafter\newwrite\csuse{tf@#2}
5073     \immediate\openout \csuse{tf@#2} \jobname.#2\relax
5074 }{}}
5075 }
```

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

```
5076 \newcommand*{\LWR@LwarpEnd}
5077 {
5078 \LWR@stoppars
5079 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
5080 \LWR@htmllementclassend{section}{textbody}
```

Print any pending footnotes:

```
5081 \LWR@printpendingfootnotes
```

Create the footer:

```
5082 \LWR@htmlelement{footer}
5083
5084 \LWR@pagebottom
5085
5086 \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.

```
5087 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
5088 {}
5089 {
5090   \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{
5091 }

5092 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
5093 \LWR@htmltag{/body}\LWR@orignewline
5094 \LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
5095 \LWR@orignewpage
```

For lateximage commands:

```
5096 \immediate\closeout\LWR@lateximagesfile
5097 }

5098 \end{warpHTML}
```

## 58 Title page

package support





load order

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

`\published` and `\subtitle`

If using the titling package, additional titlepage fields for `\published` and `\subtitle` may be added by using `\AddSubtitlePublished` in the preamble. See section 58.8.

<b>affiliation</b>	lwarp provides for the <code>\author</code> macro an additional <code>\affiliation</code> 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 <code>\theauthor</code> in the main text.
<b>reusing titlepage information</b>	The titling package maintains the definitions of <code>\thetitle</code> , <code>\theauthor</code> , 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 <code>\thanks</code> and <code>\affiliation</code> removed, and for <code>\author</code> the <code>\and</code> is replaced to generate a simple inline list of authors separated by commas. Note: <code>\theauthor</code> does not work well with <code>authblk</code> unless the traditional $\LaTeX$ syntax is used.
 <b><code>\theauthor</code>, <code>authblk</code></b>	
<b>custom titlepages</b>	<code>\printtitle</code> , <code>\printauthor</code> , etc., are provided for use inside a custom titlepage or titlingpage environment, and these retain the <code>\thanks</code> and <code>\affiliation</code> .
<code>\printthanks</code>	<code>\printthanks</code> has been added to force the printing of thanks inside a titlingpage environment when <code>\maketitle</code> is not used.
	Inside a <code>\titlepage</code> or <code>\titlingpage</code> environment, use <code>\thanks</code> instead of <code>\footnote</code> for acknowledgements, etc.

## 58.1 Setting the title, etc.

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

<code>\author</code> <code>\and</code>	<code>{\author}</code> While using <code>\maketitle</code> and print mode, the author is treated as a single-column tabular and the <code>\and</code> 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
	<code>\author{Name \\ Affiliation \and Second Name \\ Second Affiliation}</code>
	For HTML, the entire author block is placed inside a <code>&lt;div&gt;</code> of class <code>author</code> , and each individual author is inside a <code>&lt;div&gt;</code> of class <code>oneauthor</code> .
<code>\@title</code> <code>\@author</code> <code>\@date</code>	<code>\@title</code> , <code>\@author</code> , etc. store the values as originally assigned, including any <code>\thanks</code> , <code>\and</code> , or <code>\affiliation</code> . These are low-level macros intended to be used by other macros only inside a titlepage or titlingpage, and are used by <code>\maketitle</code> . 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 <code>&lt;div&gt;</code> s of class <code>oneauthor</code> , all of which are combined into one <code>&lt;div&gt;</code> of class <code>author</code> .
<code>\printtitle</code> <code>\printauthor</code> <code>\printdate</code>	<code>\printtitle</code> , etc. are user-level macros intended to be used in custom titlepage or titlingpage environments in cases where <code>\maketitle</code> is not desired. These commands preserve the <code>\thanks</code> , etc., and should not be used in the main text.



`\thetitle` `\theauthor`, and `\thedata` are available if titling has been loaded, and are sanitized user-level versions from which have been removed the `\thanks` and `\affiliation`, and `\and` is changed for inline text usage. The author is printed inline without `\affiliation` or `\thanks`, with `\and` placing commas between multiple authors. Thus, these commands are to be used in the main text whenever the user wishes to refer to the document's title and such. One practical use for this is to place the authors at the bottom of each HTML page, such as:

`\HTMLPageBottom` `{\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  $\TeX$  syntax for `\author`, optionally with `lwarp`'s `\affiliation` macro as well.

⚠ **`affiliations`** After `\maketitle` has completed, `\theauthor` retains the definition of the author, but `\and` is changed to become a comma and a space, intending to print the authors names separated by spaces. This fails when affiliations are included on their own table rows.

**`\affiliation`** A solution, provide here, is to define a macro `\affiliation` which, during `\maketitle`, starts a new row and adds the affiliation, but after `\maketitle` is finished `\affiliation` is re-defined to discard its argument, thus printing only the author names when `\author` is later used inline.

## 58.2 `\if@titlepage`

**for HTML & PRINT:** 5099 `\begin{warpall}`

`\if@titlepage` Some classes do not provide `\if@titlepage`. In this case, provide it and force it false.

```
5100 \ifcvoid{@titlepagefalse}{
5101   \newif\if@titlepage
5102   \@titlepagefalse
5103 }{}

5104 \end{warpall}
```

### 58.3 Changes for \affiliation

`\affiliation`  $\{\langle text \rangle\}$

Adds the affiliation to the author for use in `\maketitle`.

Inside `titlepage`, this macro prints its argument. Outside, it is null.

**for HTML & PRINT:** 5105 `\begin{warpall}`  
 5106 `\providerobustcmd{\affiliation}[1]{}`  
 5107 `\end{warpall}`

**for PRINT output:** 5108 `\begin{warpprint}`

5109 `\AtBeginEnvironment{titlepage}{`  
 5110 `\renewrobustcmd{\affiliation}[1]{\ \ \textsc{\small#1}}`  
 5111 `}`  
 5112  
 5113 `\AtBeginDocument{`  
 5114 `\@ifpackageloaded{titling}{`  
 5115 `\AtBeginEnvironment{titlingpage}{`  
 5116 `\renewrobustcmd{\affiliation}[1]{\ \ \textsc{\small#1}}`  
 5117 `}`  
 5118 `}{}% titling loaded`  
 5119 `}% AtBeginDocument`

5120 `\end{warpprint}`

**for HTML output:** 5121 `\begin{warpHTML}`

Env `titlepage` Sets up a `<div>` of class `titlepage`. Provided even for memoir class, since it is used by `\maketitle`.

5122 `\DeclareDocumentEnvironment{titlepage}{}`  
 5123 `{`  
 5124 `\renewrobustcmd{\affiliation}[1]{\ \ \InlineClass{affiliation}{##1}}`  
 5125 `\LWR@printpendingfootnotes`  
 5126 `\LWR@forcenewpage`  
 5127 `\BlockClass{titlepage}`  
 5128 `}`  
 5129 `{`  
 5130 `\endBlockClass`  
 5131 `\LWR@printpendingfootnotes`  
 5132 `}`

5133 `\end{warpHTML}`

## 58.4 Printing the thanks

**for HTML & PRINT:** 5134 \begin{warpall}

\printthanks Forces the \thanks to be printed.

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

5135 \newcommand\*{\printthanks}{\@thanks}

5136 \end{warpall}

## 58.5 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:** 5137 \begin{warpHTML}

\printtitle

5138 \newcommand\*{\printtitle}

5139 {

5140 \LWR@stoppars

5141 \LWR@htmltag{\LWR@tagtitle}%

5142 \@title%

5143 \LWR@htmltag{\LWR@tagtitleend}

5144 \LWR@startpars

5145 }

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

5146 \newcommand\*{\LWR@printthetitle}

5147 {

5148 \LWR@stoppars

5149 \LWR@htmltag{\LWR@tagtitle}%

5150 \thetitle%

5151 \LWR@htmltag{\LWR@tagtitleend}

5152 \LWR@startpars

5153 }

\printauthor HTML version.

5154 \newcommand\*{\printauthor}{

The entire author block is contained in a <div> named author:

```
5155 \begin{BlockClass}{author}
```

\and finishes one author and starts the next:

```
5156 \renewcommand{\and}{%
5157 \end{BlockClass}
5158 \begin{BlockClass}{oneauthor}
5159 }
```

Individual authors are contained in a <div> named oneauthor:

```
5160 \begin{BlockClass}{oneauthor}
5161 \@author
5162 \end{BlockClass}
5163 \end{BlockClass}
5164 }
```

\printdate

```
5165 \newcommand*{\printdate}{%
5166 \begin{BlockClass}{titledate}
5167 \@date
5168 \end{BlockClass}
5169 }
```

```
5170 \end{warpHTML}
```

## 58.6 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:** 5171 \begin{warpprint}

\printtitle

```
5172 \newcommand*{\printtitle}{\Huge\@title}}
```

\printauthor Print mode.

```
5173 \newcommand*{\printauthor}
5174 {\large\begin{tabular}{t}{c}\@author\end{tabular}}}
```

`\printdate`

```
5175 \newcommand*{\printdate}{\small\textit{\@date}}
```

```
5176 \end{warpprint}
```

## 58.7 \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 51 for other kinds of footnotes.

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

for HTML output: 5177 `\begin{warppHTML}`

```
5178 \@ifclassloaded{memoir}
5179 {
5180 \newcommand{\LWR@setfootnoteseries}{%
5181     \renewcommand\thefootnote{\@arabic\c@footnote}%
5182 }
5183 }{% not memoir
5184 \if@titlepage
5185 \newcommand{\LWR@setfootnoteseries}{%
5186     \renewcommand\thefootnote{\@arabic\c@footnote}%
5187 }
5188 \else
5189 \newcommand{\LWR@setfootnoteseries}{%
5190     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
5191 }
5192 \fi
5193 }{% not memoir
```

`\LWR@maketitlesetup` Patches `\thanks` macros.

```
5194 \newcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
5195 \LWR@setfootnoteseries%
5196 \def\@makefnmark{\textsuperscript{\thefootnote}}
```

`\thefootnote`  $\Rightarrow$  `\nameuse{arabic}{footnote}`, or  
`\thefootnote`  $\Rightarrow$  `\nameuse{fnsymbol}{footnote}`

Redefine the footnote text:

```
5197 \long\def\@makefntext##1{%
```

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

```
5198 \textsuperscript{\@thefnmark}~%
```

```
\makethanksmark => \thanksfootmark => \tamark =>
\@thefnmark => \itshape a (or similar)
```

Print the text:

```
5199 ##1%
5200 }%
5201 }
```

```
\@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`.

```
5202 \def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
5203   \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLUnicode{2016}}\or
5204   **\or \HTMLentity{dagger}\HTMLentity{dagger} \or
5205   \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.

```
5206 \renewcommand*\maketitle{%
```

An HTML titlepage `<div>` is used for all classes.

```
5207 \begin{titlepage}
```

Set up special patches:

```
5208 \LWR@maketitlesetup
```

Typeset the title, etc:

```
5209 \@maketitle
```

Immediately generate any \thanks footnotes:

```
5210 \@thanks
```

Close the HTML titlepage div and cleanup:

```
5211 \end{titlepage}
5212 \setcounter{footnote}{0}%
5213 \global\let\thanks\relax
5214 \global\let\maketitle\relax
5215 \global\let\@maketitle\relax
5216 \global\let\@thanks\@empty
5217 \global\let\@author\@empty
5218 \global\let\@date\@empty
5219 \global\let\@title\@empty
5220 \global\let\title\relax
5221 \global\let\author\relax
5222 \global\let\date\relax
5223 \global\let\and\relax
5224 }
```

\@maketitle HTML mode. Typesets the title, etc.:

```
5225 \DeclareDocumentCommand{\@maketitle}{-}{%
5226   \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
5227   \@title
5228   \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
5229   \begin{BlockClass}{author}}
```

For IEEEtran class:

```
5230   \renewcommand*{\cr}{-}
5231   \renewcommand*{\crr}{-}
5232   \renewcommand*{\noalign}{-}

5233   \renewcommand{\and}{-}
5234   \end{BlockClass}
5235   \begin{BlockClass}{oneauthor}
5236 }
5237 \begin{BlockClass}{oneauthor}
5238   \@author
5239 \end{BlockClass}
5240 \end{BlockClass}
5241 \begin{BlockClass}{titledate}
5242   \@date
5243 \end{BlockClass}
5244 }
```

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

```
5245 \newcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
5246 \@thanks
```

Set up special patches:

```
5247 \LWR@maketitlesetup
```

Typeset the title, etc:

```
5248 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
5249 \@thanks
```

```
5250 }
```

```
5251 \end{warpHTML}
```

## 58.8 `\published` and `\subtitle`

`\subtitle` and `\published` To add `\subtitle` and `\published` to the titlepage, load the titling package and use `\AddSubtitlePublished` in the preamble.

The default `lwarp.css` has definitions for the `published` and `subtitle` classes.

After titling is loaded, `\AddSubtitlePublished` is created, which when used then creates a number of additional macros, and also assigns some of the titling hooks. 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`.

 **titling hooks**

**for HTML & PRINT:** `5252 \begin{warpall}`

`\AddSubtitlePublished` Adds `\published` and `\subtitle`, and related.

```
5253 \AfterPackage{titling}{%
```

```
5254 \newcommand*{\AddSubtitlePublished}{%
```

```
5255
```

```
5256 \newcommand{\@published}{}
```



```

5257
5258 \newcommand{\published}[1]{\gdef\@published{##1}}
5259
5260 \renewcommand*{\maketitlehooka}{\printpublished}
5261
5262 \newcommand*{\printpublished}{%
5263 \warpprintonly{\begin{center}\unskip}%
5264 \begin{BlockClass}{published}%
5265 \warpprintonly{\large\itshape}%
5266 \@published%
5267 \end{BlockClass}%
5268 \warpprintonly{\end{center}}}%
5269 }
5270
5271 \newcommand{\@subtitle}{}
5272
5273 \newcommand{\subtitle}[1]{\gdef\@subtitle{##1}}
5274
5275 \renewcommand*{\maketitlehookb}{\printsubtitle}
5276
5277 \newcommand*{\printsubtitle}{%
5278 \warpprintonly{\begin{center}\unskip}%
5279 \begin{BlockClass}{subtitle}%
5280 \warpprintonly{\Large\itshape}%
5281 \@subtitle%
5282 \end{BlockClass}%
5283 \warpprintonly{\end{center}}}%
5284 }
5285
5286 }% \AddSubtitlePublished
5287 }% AfterPackage

5288 \end{warpall}

```

## 59 Abstract

The following code replaces the  $\LaTeX$  default, and will itself be replaced later if the abstract package is loaded.

for HTML output: 5289 \begin{warpHTML}

\abstractname User-redefinable title for the abstract.

Also over-written by the babel package.

```

5290 \providecommand*{\abstractname}{Abstract}

```

Some classes allow an optional name, so it is allowed here.

Env **abstract**

```

5291 \DeclareDocumentEnvironment{abstract}{0{\abstractname}}
5292 {
5293 \LWR@forcenewpage
5294 \BlockClass{abstract}
5295 \BlockClassSingle{abstracttitle}{#1}
5296 }
5297 {
5298 \endBlockClass
5299 }

5300 \end{warpHTML}

```

## 60 Quote and verse

### 60.1 Citations and attributions

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

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

**for HTML output:**

```

5301 \begin{warpHTML}
5302 \newcommand{\attribution}[1]{%
5303 \InlineClass{attribution}{--\,#1}}% emdash
5304 \end{warpHTML}

```

**for PRINT output:**

```

5305 \begin{warpprint}
5306 \newcommand{\attribution}[1]{\textsc{---\,#1}}
5307 \end{warpprint}

```

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

**for HTML output:**

```

5308 \begin{warpHTML}
5309 \newcommand{\citetitle}[1]{%
5310 \InlineClass{citetitle}{--\,#1}}% emdash
5311 \end{warpHTML}

```

**for PRINT output:**

```

5312 \begin{warpprint}
5313 \newcommand{\citetitle}[1]{\textsl{---\,#1}}
5314 \end{warpprint}

```

## 60.2 Quotes, quotations

for HTML output: 5315 `\begin{warpHTML}`

Env quote

```

5316 \renewenvironment*{quote}
5317 {
5318 \LWR@forcenewpage
5319 \LWR@htmlblocktag{blockquote}
5320 }
5321 {\LWR@htmlblocktag{/blockquote}}
5322
5323 \renewenvironment*{quotation}
5324 {
5325 \LWR@forcenewpage
5326 \LWR@htmlblocktag{blockquotation}
5327 }
5328 {\LWR@htmlblocktag{/blockquotation}}

5329 \end{warpHTML}

```

## 60.3 Verse

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

---

```

\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}

```


---

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

Len `\leftmargini`

Len `\TMLvleftskip`

Len `\TMLleftmargini`

 **spacing** 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:** 5330 \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 \leftskip and \leftmargini for print output, and optionally select different values for HTML.

Len \TMLvleftskip Sets \vleftskip inside a verse environment in HTML.

```
5331 \newlength{\TMLvleftskip}
5332 \setlength{\TMLvleftskip}{1em}
```

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

```
5333 \newlength{\TMLleftmargini}
5334 \setlength{\TMLleftmargini}{4.5em}
```

```
5335 \end{warpall}
```

## 61 Verbatim and tabbing

**for HTML & PRINT:** 5336 \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.

```
5337 \newlength{\VerbatimHTMLWidth}
5338 \setlength{\VerbatimHTMLWidth}{4in}
5339 \end{warpall}
```

**for HTML output:** 5340 \begin{warpHTML}

Bool LWR@verbtags Used to temporarily turn off verbatim tags while doing \verbatiminput in the HTML head.

```
5341 \newbool{LWR@verbtags}
5342 \booltrue{LWR@verbtags}
```

\LWR@atbeginverbatim [*<style>*] {*<class>*}

Encloses a verbatim environment with the given css class.

```
5343 \newcommand*{\LWR@atbeginverbatim}[2] []
5344 {%
```

Avoid excessive space between lines:

```
5345 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
5346 \LWR@stoppars%
```

Create a new pre of the given class. The tags may temporarily be turned off for internal use, such as loading the MATHJAX script.

```
5347 \ifbool{\LWR@verbtags}{%
5348   \LWR@htmltag{pre class="#2"
5349   \ifthenelse{\equal{#1}{}}{}{style="#1"}}%
5350 }\LWR@orignewline% pre
5351 }{}%
```

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.

```
5352 \begingroup%
```

```
5353 \LWR@orignormalsize%
5354 \LWR@origttfamily%
```

Since inside a <pre>, restore the original list processing:

```
5355 \LWR@restoreoriglists%
```

Turn off babel-french extra space before punctuation:

```
5356 \LWR@FBcancel%
```

Do not produce HTML tags for \hspace inside a verse par. Restore plain  $\TeX$  \hspace functionality:

```
5357 \let\LTxMacro{\hspace}\LWR@orighspace}%
5358 }
```

`\LWR@afterendverbatim` Finishes enclosing a verbatim environment.

```

5359 \newcommand*{\LWR@afterendverbatim}{%
5360
5361 \endgroup%

```

At the end of the environment, close the pre:

```

5362 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}
5363
5364 }{}%

```

Resume regular paragraph handling:

```

5365 \LWR@startpars%
5366 }

```

`\verbatiminput`    `{\langle filename \rangle}`

Patch `\verbatiminput` to add HTML tags:

```

5367 \let\LWRV@origverbatim@input\verbatim@input
5368
5369 \renewcommand{\verbatim@input}[2]{%
5370 \ifbool{LWR@verbtags}{\LWR@forcenewpage}{}%
5371 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%%
5372 \LWRV@origverbatim@input{#1}{#2}%
5373 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
5374 }

```

Env    `verbatim`

```

5375 \AfterEndPreamble{
5376 \LWR@traceinfo{Patching verbatim.}
5377 \AtBeginEnvironment{verbatim}{%
5378 \LWR@forcenewpage%
5379 \LWR@atbeginverbatim{verbatim}\unskip\LWR@origvspace*{-2\baselineskip}%
5380 }
5381 \AfterEndEnvironment{verbatim}{%
5382    \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
5383 }
5384 }

```

Env    `tabbing`    The `tabbing` environment works, except that `svg math` and `lateximages` do not yet work inside the environment.

```

5385 \AfterEndPreamble{
5386 \LWR@traceinfo{Patching tabbing.}

```

```

5387 \AtBeginEnvironment{tabbing}{%
5388 \LWR@forcenewpage
5389 \LWR@atbeginverbatim{tabbing}\unskip\LWR@origvspace*{-\baselineskip}%
5390 }
5391 \AfterEndEnvironment{tabbing}{%
5392     \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
5393 }
5394 }

5395 \end{warpHTML}

```

## 62 Theorems

`\newtheorem`  $\{\langle text \rangle\}$  [ $\langle counter \rangle$ ] -or- [ $\langle oldname \rangle$ ]  $\{\langle text \rangle\}$

A few minor changes are made to supply HTML tags.

- The entire theorem is placed into a `<div>` of class `theoremcontents`.
- The label for each theorem is placed inside a `<span>` of class `theoremlabel`.
- The contents are placed inside a `<div>` of class `theoremcontents`.

**for HTML output:** 5396 `\begin{warpHTML}`

`\@begintheorem`  $\{\langle name \rangle\}$   $\{\langle number \rangle\}$

```

5397 \renewcommand{\@begintheorem}[2]{%
5398 \LWR@forcenewpage
5399 \BlockClass{theoremcontents}
5400 \trivlist
5401 \item[\InlineClass{theoremlabel}{#1\ #2\ }]\itshape
5402 }

```

`\@opargbegintheorem`  $\{\langle name \rangle\}$   $\{\langle number \rangle\}$   $\{\langle oparg \rangle\}$

```

5403 \renewcommand{\@opargbegintheorem}[3]{%
5404 \LWR@forcenewpage
5405 \BlockClass{theoremcontents}
5406 \trivlist
5407 \item[\InlineClass{theoremlabel}{#1\ #2\ (#3)\ }]\itshape
5408 }

```

`\@endtheorem`

```
5409 \renewcommand*{\@endtheorem}{%
5410 \endtrivlist
5411 \endBlockClass% theoremcontents
5412 }

5413 \end{warpHTML}
```

## 63 Lists

The environments `itemize`, `enumerate`, and `description` are patched when `lwarp` is started. These patches support the standard  $\TeX$  environments, as well as those of `enumerate`, `enumitem`, and `paralist`, and at least the French version of `babel`. Additional patches are done on a package-specific basis.

The  $\TeX$  source for `itemize` and `enumerate` are found in `source2e`, but the source for `description` is found in `article.cls`, etc.


**empty item** To have an empty item, use `\mbox{}` or a trailing backslash. This forces a new line in print output, matching the new line which will appear in HTML output. Ex:

---

```
begin{itemize}
item \mbox{}
    \begin{itemize}
...
    \end{itemize}
item \
    \begin{itemize}
...
    \end{itemize}
```

---

`\makelabel` While inside a list environment, `lwarp` nullifies a number of  $\TeX$  horizontal skip and fill commands, allowing the user to define `\makelabel` for print mode while HTML mode ignores those commands.

 **label font** When defining `\makelabel` in a list environment, use `\textbf` etc. instead of `\bfseries`.



## 63.1 List environment

for HTML output: 5414 `\begin{warpHTML}`

`\LWR@printcloselist` May be locally redefined by `enumerate` or `description`.

```
5415 \newcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
```

`\LWR@printopenlist` May be locally redefined by `enumerate` or `description`.

```
5416 \newcommand*{\LWR@printopenlist}{\ul style="\LWR@origmbox{list-style-type:none}}}
```

`\@mklab` Removes PDF spacing.

```
5417 \AtBeginDocument{
5418 \def\@mklab#1{%
5419 %   \hfil %
5420 %   #1}
5421 \let\makelabel\@mklab
5422 }
```

`\@donoparitem` Modified for HTML output by replacing  $\TeX$  boxes with plain text. Also removes PDF spacing.

```
5423 \def\@donoparitem{%
5424   \@donoparitemfalse
5425 %   \global\setbox\@labels\hbox{\hskip -\leftmargin
5426 %                               \unhbox\@labels
5427 %                               \hskip \leftmargin}}%
5428 %   \if@minipage\else
5429 %     \@tempskipa\lastskip
5430 %     \vskip -\lastskip
5431 %     \advance\@tempskipa\outerparskip
5432 %     \advance\@tempskipa -\parskip
5433 %     \vskip\@tempskipa
5434 %   \fi
5435 }
```

`\@item` Modified for HTML output by replacing  $\TeX$  boxes with plain text. Also removes PDF spacing.

```
5436 \def\LWR@HTML@item[#1]{%
5437 \LWR@traceinfo{\@item}
5438 \if@donoparitem
5439   \@donoparitem
```

```

5440 \else
5441 % \if@inlabel
5442 % \indent
5443 \par
5444 % \fi
5445 \ifhmode
5446 % \unskip\unskip
5447 \par
5448 \fi
5449 \if@newlist
5450 \if@nobreak
5451 \@nbitem
5452 \else
5453 % \addpenalty\@beginparpenalty
5454 % \addvspace\@topsep
5455 % \addvspace{-\parskip}%
5456 \fi
5457 \else
5458 % \addpenalty\@itempenalty
5459 % \addvspace\itemsep
5460 \fi
5461 \global\@inlabeltrue
5462 \fi
5463 % \everypar{%
5464 % \@minipagefalse
5465 % \global\@newlistfalse

5466 % \if@inlabel
5467 % \global\@inlabelfalse

5468 % {\setbox\z@\lastbox
5469 % \ifvoid\z@
5470 % \kern-\itemindent
5471 % \fi}%

5472 % \box\@labels
5473 % \penalty\z@
5474 % \fi

5475 % \if@nobreak
5476 % \@nobreakfalse
5477 % \clubpenalty \@M
5478 % \else
5479 % \clubpenalty \@clubpenalty
5480 % \everypar{%
5481 % \fi}%

5482 \if@noitemarg
5483 \@noitemargfalse

```

```

5484     \if@nmbrrlist

5485         \refstepcounter\@listctr
5486     \fi
5487 \fi

5488     \makelabel{#1}%
5489 %     \sbox\@tempboxa{\makelabel{#1}%
5490 %     \global\setbox\@labels\hbox{%
5491 %         \unhbox\@labels
5492 %         \hskip \itemindent
5493 %         \hskip -\labelwidth
5494 %         \hskip -\labelsep
5495 %         \ifdim \wd\@tempboxa >\labelwidth
5496 %             \box\@tempboxa

5497 %     \else
5498 %         \hbox to\labelwidth {\unhbox\@tempboxa}%
5499 %     \fi
5500 %     \hskip \labelsep}%
5501 \ignorespaces%
5502 }

```

\@nbitem

```

5503 \def\@nbitem{%
5504 %     \@tempskipa\@outerparskip
5505 %     \advance\@tempskipa -\parskip
5506 %     \addvspace\@tempskipa
5507 }

```

\LWR@listitem [*<label>*]

Handles \item inside a list, itemize, or enumerate.

See \LWR@openparagraph where extra \hspace is used to leave room for the label while inside a list during paragraph construction.

```

5508 \newcommand*\LWR@listitem{%
5509 \LWR@stoppars%
5510 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem}%
5511 \LWR@htmltag{li}%
5512 \LWR@startpars%
5513 \LWR@origitem%
5514 }

```

`\LWR@nulllistfills` Nullifies various TeX fill commands, in case they are used inside `\makelabel`.

```
5515 \newcommand*\LWR@nulllistfills{%
5516 \renewcommand*\hss{%}%
5517 \renewcommand*\llap[1]{##1}%
5518 \renewcommand*\rlap[1]{##1}%
5519 \renewcommand*\hfil{%}%
5520 \renewcommand*\hfilneg{%}%
5521 \renewcommand*\hfill{%}%
5522 }
```

Env `list` `{\label}` `{\commands}`

```
5523 \newcommand*\LWR@liststart{%
5524 \LWR@traceinfo{\LWR@liststart}%
5525 \LWR@stoppars%
5526 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloselist}%
5527 \LWR@htmltag{\LWR@printopenlist}\LWR@orignewline%
5528 \LWR@startpars%
5529 \setlength{\topsep}{0pt}%
5530 \setlength{\partopsep}{0pt}%
5531 \setlength{\itemsep}{0pt}%
5532 \setlength{\parsep}{0pt}%
5533 \setlength{\leftmargin}{0pt}%
5534 \setlength{\rightmargin}{0pt}%
5535 \setlength{\listparindent}{0pt}%
5536 \setlength{\itemindent}{0pt}%
5537 \setlength{\labelsep}{1em}%
5538 \LWR@nulllistfills%
5539 }
```

```
5540 \newcommand*\LWR@listend{%
5541 \LWR@traceinfo{\LWR@listend}%
5542 \LWR@stoppars%
5543 \LWR@closeprevious{\LWR@depthlist}%
5544 \LWR@startpars%
5545 }
```

## 63.2 Itemize

`\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.

```

5546 \newcommand*{\LWR@itemizeitem}{%
5547 \LWR@stoppars%
5548 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloseitem}%
5549 \LWR@htmltag{li}%
5550 \LWR@startpars%
5551 \LWR@origitem%
5552 }
```

Env `itemize` [*options*]

```

5553 \newcommand*{\LWR@itemizestart}{%
5554 \renewcommand*{\LWR@printcloseitem}{\LWR@printcloseitemize}
5555 \renewcommand*{\LWR@printopenlist}{\ul style="\LWR@origmbox{list-style-type:none}}
5556 \let\item\LWR@itemizeitem%
5557 \LWR@nulllistfills%
5558 }
```

### 63.3 Enumerate

An HTML unordered list is used with customized  $\text{\LaTeX}$ -generated labels.

Env `enumerate` [*options*]

```

5559 \newcommand*{\LWR@enumeratestart}{%
5560 \renewcommand*{\LWR@printcloseitem}{\LWR@printcloseitemize}
5561 \renewcommand*{\LWR@printopenlist}{\ul style="\LWR@origmbox{list-style-type:none}}
5562 \let\item\LWR@itemizeitem%
5563 \LWR@nulllistfills%
5564 }
```

### 63.4 Description

`\LWR@descitem` [*label*] Handles an `\item` inside a description.

```

5565 \newcommand*{\LWR@descitem}[1][]%
5566 {%
5567 \LWR@stoppars%
5568 \LWR@setlatestname{#1}%
5569 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem}%

```

Temporarily disable `\hspace`, which `article.cls`, etc. use per `\item` for descriptions only. This causes `lwarp` to mistakenly place an empty span between HTML list tags.

```
5570 \LetLtxMacro{\hspace}{\LWR@nohspace}%
```

Process the original `\item` code:

```
5571 \LWR@origitem[]%
```

Restore `\hspace` for use in the item text:

```
5572 \LetLtxMacro{\hspace}{\LWR@hspace}%
5573 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
5574 \LWR@orignewline%
5575 \LWR@htmltag{dd}%
5576 \LWR@startpars%
5577 }
```

Env `description` [*options*]

```
5578 \newcommand*{\LWR@descriptionstart}{%
5579 \renewcommand*{\LWR@printcloselist}{\LWR@printclosedescription}
5580 \renewcommand*{\LWR@printopenlist}{dl}
5581 \let\item\LWR@descitem%
5582 \LWR@nulllistfills%
5583 }
```

## 63.5 Patching the lists

`\LWR@patchlists` Patches list environments.

`\LWR@patchlists` remembers `\item` as defined by whatever packages have been loaded, then patches the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native  $\TeX$  environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```
5584 \newcommand*{\LWR@patchlists}{%
5585   \LetLtxMacro\item\LWR@listitem%
5586   \LetLtxMacro\@item\LWR@HTML@item%
5587   \renewcommand*{\@trivlist}{%
5588     \LWR@traceinfo{@trivlist start}%
5589     \LWR@liststart%
5590     \LWR@orig@trivlist%
5591     \LWR@traceinfo{@trivlist done}%
5592   }%
```

```

5593 \renewcommand*{\trivlist}{%
5594     \LWR@traceinfo{trivlist}%
5595     \LWR@origtrivlist%
5596 }%
5597 \renewcommand*{\endtrivlist}{%
5598     \LWR@traceinfo{endtrivlist start}%
5599     \LWR@origendtrivlist\LWR@listend%
5600     \LWR@traceinfo{endtrivlist done}%
5601 }%
5602 \renewcommand*{\itemize}{%
5603     \LWR@itemizestart\LWR@origitemize%
5604 }%
5605 \renewcommand*{\enumerate}{%
5606     \LWR@enumeratestart\LWR@origenumerate%
5607 }%
5608 \renewcommand*{\description}{%
5609     \LWR@descriptionstart\LWR@origdescription%
5610 }%
5611 }

```

`\LWR@restoreoriglists` Restores the original trivlist environment.

```

5612 \newcommand*{\LWR@restoreoriglists}{%
5613 \LWR@traceinfo{\LWR@restoreoriglists}%
5614 \LetLtxMacro\item\LWR@origitem%
5615 \LetLtxMacro\@item\LWR@orig@item%
5616 \let\@trivlist\LWR@orig@trivlist%
5617 \let\trivlist\LWR@origtrivlist%
5618 \let\endtrivlist\LWR@origendtrivlist%
5619 \LetLtxMacro\itemize\LWR@origitemize%
5620 \LetLtxMacro\enditemize\LWR@endorigitemize%
5621 \LetLtxMacro\enumerate\LWR@origenumerate%
5622 \LetLtxMacro\endenumerate\LWR@endorigenumerate%
5623 \LetLtxMacro\description\LWR@origdescription%
5624 \LetLtxMacro\enddescription\LWR@endorigdescription%
5625 \let\@mklab\LWR@orig@mklab%
5626 \let\makelabel\LWR@origmakelabel%
5627 \let\@donoparitem\LWR@orig@donoparitem%
5628 \let\@nbitem\LWR@orig@nbitem%
5629 }

5630 \end{warpHTML}

```

## 64 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

### 64.1 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
```

⚠ floatrow

This includes before and after defining any macro which used \ttabbox from floatrow.

⚠ tabular inside another  
environment

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

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

#### Cell contents:



⚠ paragraphs

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

⚠ `\multirow`

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

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

`vposn`

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

- The `\multirow` documentation regarding colored cells recommends using a negative number of rows. This will not work with `lwarp`, so `\warpprintonly` and `\warphTMLonly` must be used to make versions for print and HTML.
- See section 232.2 for `\multicolumnrow`.

⚠ `\multicolumn` &  
`\multirow`

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

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

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

⚠ skipped cells

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

⚠ empty cells

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

`vposn`

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

⚠ macro in a table  
custom macros

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

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

### Column specifiers:

⚠ \* column specification

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

@ and !

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

`\multirow`

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

⚠ `\newcolumnntype`

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

**Rules:**

vertical rules

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

width and trim

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

full-width rules

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

combined rules

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

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

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

⚠ `\warpprintonly`  
misplaced `\noalign`

- For `\toprule` and `\bottomrule`, when combined with a `warpprint` or `warpHTML` environment, if a “misplaced `\noalign`” error occurs, change

```
This & That \endhead
```

to

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

and likewise with the other `\end` headings. Keep the `\endfirsthead` row unchanged, as it is still relevant to HTML output.

**colortbl:**

⚠ row/cell color

Only use `\rowcolor` and `\cellcolor` at the start of a row, in that order.

`colortbl` ignores the overhang arguments.

**Other:**

longtable headings

- `tabularx` ignores the width, but `X` columns do produce paragraph columns or multicolumns.
- For `longtable`, place headings and footings which do not apply to HTML inside `\warpprintonly{}`.

⚠ `S` columns

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

algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:

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

## 64.2 Token lookahead

Used by `\LWR@futurenonSPACElet` to look at the next token.

**for HTML output:** 5631 `\begin{warpHTML}`

`\LWR@mynexttoken`

5632 `\newcommand\LWR@mynexttoken\relax`

`\futurelet` copies the next token then executes a function to analyze

`\LWR@futurenonSPACElet` does the same, but ignores intervening white space

Based on the booktabs style:

`\LWR@futurenonSPACElet`

```
5633 \def\LWR@futurenonSPACElet#1{\def\LWR@cs{#1}%
5634 \afterassignment\LWR@fns lone\let\nexttoken= }
5635 \def\LWR@fns lone{\expandafter\futurelet\LWR@cs\LWR@fns ltwo}
5636 \def\LWR@fns ltwo{%
5637 \expandafter\ifx\LWR@cs\@sptoken\let\next=\LWR@fns lthree%
5638 \else\let\next=\nexttoken\fi\next}
5639 \def\LWR@fns lthree{\afterassignment\LWR@fns lone\let\next= }
```

`\LWR@getmynexttoken` Looks ahead and copies the next token into `\LWR@mynexttoken`.

```
5640 \newcommand*{\LWR@getmynexttoken}{%
5641 \LWR@traceinfo{\LWR@getmynexttoken}%
5642 % nothing must follow this next line
5643 \LWR@futurenonSPACElet\LWR@mynexttoken\LWR@tabledatacolumn tag
5644 }
```

## 64.3 Tabular variables

Bool `LWR@startedrow` True if should print a row tag before this column.

```
5645 \newbool{LWR@startedrow}
5646 \boolfalse{LWR@startedrow}
```

Bool LWR@tabularcelladded True if have added a data cell for this position.

```
5647 \newbool{LWR@tabularcelladded}
5648 \boolfalse{LWR@tabularcelladded}
```

Bool LWR@doinghline True if the next row will have an hline or midrule above it. Also used for \midrule.

```
5649 \newbool{LWR@doinghline}
5650 \boolfalse{LWR@doinghline}
```

Bool LWR@doingtbrule True if the next row will have a top/bottom rule above it.

```
5651 \newbool{LWR@doingtbrule}
5652 \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.

```
5653 \newbool{LWR@doingcmidrule}
5654 \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.

```
5655 \newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty \multirow cell, and thus there is no data tag to close.

```
5656 \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.

```
5657 \newbool{LWR@skippingmcolrowcell}
```

Bool LWR@skipatbang True if just finished a \multicolumn so should not create the trailing @ or ! columns table data cells.

```
5658 \newbool{LWR@skipatbang}
```

Bool LWR@emptyatbang True if finishing a row and should print empty @ or ! column table data cells.

```
5659 \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.

```
5660 \newbool{LWR@intabularmetadata}
5661 \boolfalse{LWR@intabularmetadata}
```

Ctr LWR@tabularDepth Tracks whether & is being used inside a tabular.

```
5662 \newcounter{LWR@tabulardepth}
5663 \setcounter{LWR@tabulardepth}{0}
```

Ctr LWR@tabularpardepth Tracks whether should look ahead at the next token when encountering a \par while processing tabular contents.

When LWR@tabularpardepth is deeper than LWR@tabulardepth then lwarp has started looking at the contents of the tabular, and thus any \pars encountered must be followed by another token lookahead.

```
5664 \newcounter{LWR@tabularpardepth}
5665 \setcounter{LWR@tabularpardepth}{0}
```

```
5666 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
5667 \newcommand*{\LWR@pposition}{}
5668 \newcommand*{\LWR@pleft}{}
5669 \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 lrrccpc, exactly one letter per  $\LaTeX$  table column, without @, !, >, <, or the vertical bar.

```
5670 \newcommand*{\LWR@tablecolspec}{}%
```

\LWR@strresult Holds the result of Str functions.

```
5671 \providecommand*{\LWR@strresult}{}
5672 \providecommand*{\LWR@strresulttwo}{}%
```

\LWR@origcolspec Holds the original column specs given to tabular.

```
5673 \newcommand*{\LWR@origcolspec}{}%
```

Ctr LWR@tablecolspecwidth Holds the number of tokens in the table columns specification.

This includes one for each @, !, <, > column, and also one for each of the parameters of p, @, !, <, > columns, and three for each D column.

(This is not the total # of  $\text{\TeX}$  columns in the table.)

```
5674 \newcounter{LWR@tablecolspecwidth}
```

Ctr LWR@tablecolspecindex While parsing the  $\text{\TeX}$  table column specification, starts at 1 and is incremented per token of the specification. While producing the table, resets to 1 at the start of the table and also at each end of line, and is incremented by 1 by each ampersand.

```
5675 \newcounter{LWR@tablecolspecindex}
```

Ctr LWR@tablecolindex While parsing the  $\text{\TeX}$  table column specification, starts at 1 and is incremented per token of the specification. While producing the table, resets to 1 at the start of the table and also at each end of line, and is incremented by 1 by each ampersand.

```
5676 \newcounter{LWR@tablecolindex}
```

Ctr LWR@tabletotalcols While parsing a table column specification, begins at 0 and increments by 1 per  $\text{\TeX}$  table column. Eventually holds the final number of  $\text{\TeX}$  table columns in each row, not counting @ and ! columns. (In HTML, @ and ! cells become their own columns, but are not included in LWR@tabletotalcols.)

```
5677 \newcounter{LWR@tabletotalcols}
```

Ctr LWR@tabletotalcolsnext Holds the next  $\text{\TeX}$  table column index while parsing, equal to one more than LWR@tabletotalcols.

```
5678 \newcounter{LWR@tabletotalcolsnext}
```

LWR@colatspec A data array of specifications for @ columns. The leftmost's index is leftedge, the others are counter values. See section 37.

LWR@colbangspec A data array of specifications for ! columns. The leftmost's index is leftedge, the others are counter values. See section 37.

LWR@colbeforespec A data array of specifications for > columns.

LWR@colafterspec A data array of specifications for < columns.

LWR@colbarspec A data array of specifications for vertical rules.

## 64.4 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/11638/where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860)

\LWR@instertatbangcols

```
5679 \newcommand*{\LWR@insertatbangcols}{%
5680 \ifbool{\LWR@skipatbang}%
5681 {}%
5682 {%
5683     \LWR@printatbang{at}{\arabic{\LWR@tablecolindex}}%
5684     \LWR@printatbang{bang}{\arabic{\LWR@tablecolindex}}%
5685 }%
5686 }
```

\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.

```
5687 \newcommand*{\LWR@closetabledatacell}{%
5688 \global\booltrue{\LWR@intabularmetadata}%
5689 \ifbool{\LWR@exitingtabular}{}%
5690 {% not exiting tabular
5691     \ifboolexpr{bool{\LWR@skippingmrowcell} or bool{\LWR@skippingmcolrowcell}}%
5692     {%
```

If not skipping a \multicolumnrow cell, insert the @ and ! columns after this non-existent column.

```
5693     \ifbool{\LWR@skippingmcolrowcell}%
5694     {}%
5695     {\LWR@insertatbangcols}%
5696 }%
5697 {% not skippingmrowcell
```

Insert any < then any @ and ! column contents, unless muted for the \bottomrule or a \multicolumn:

```
5698     \unskip%
5699     \ifboolexpr{%
5700         bool{\LWR@tabularmutemods} or
5701         bool{\LWR@skipatbang} or
5702         bool{\LWR@emptyatbang}
5703     }%
5704     {}%
5705     {\LWR@getexparray{\LWR@colafterspec}{\arabic{\LWR@tablecolindex}}}%

```

Close paragraphs:

```
5706      \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
5707      \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell.

Close any color <div>s.

```
5708      \whileboolexpr{test {\ifnumcomp{\value{LWR@cellcolordepth}}{>}{0}}}{%
5709          \LWR@htmltag{/div}\LWR@orignewline%
5710          \addtocounter{LWR@cellcolordepth}{-1}%
5711      }%
```

Skip the @ and ! cells if are closing a multicolumn cell.

```
5712      \leavevmode\unskip\LWR@htmltag{/td}\LWR@orignewline%
5713      \global\booltrue{LWR@tabularcelladded}%
5714      \LWR@insertatbangcols%
5715      }% not skipping mrowcell
5716      }% not exiting tabular
5717      \global\boolfalse{LWR@skippingmrowcell}%
5718      \global\boolfalse{LWR@skippingmcolrowcell}%
5719      \global\boolfalse{LWR@skipatbang}%
```

Color control. Column is set by >{} for each cell, so it must be cleared here.

```
5720 \renewcommand*{\LWR@cellHTMLcolor}{}
5721 \renewcommand*{\LWR@columnHTMLcolor}{}
5722 \setcounter{LWR@cellcolordepth}{0}
5723 }
```

When not used inside a tabular, & performs its original function as recorded here ( with catcode 4 ).

```
5724 \let\LWR@origampmacro&

5725 \end{warpHTML}
```

#### 64.4.1 Localizing & catcodes

**for HTML & PRINT:** 5726 \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.

```
5727 \newcommand{\StartDefiningTabulars}{%
5728 \LWR@traceinfo{StartDefiningTabulars}%
5729 \warpHTMLonly{\catcode'\&=\active}%
5730 }
```

`\EndDefiningTabulars` Place after defining something with & in it.

```
5731 \newcommand{\EndDefiningTabulars}{%
5732 \LWR@traceinfo{EndDefiningTabulars}%
5733 \warpHTMLonly{\catcode'\&=4}%
5734 }

5735 \end{warpall}
```

### 64.4.2 Handling &

**for HTML output:** 5736 \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.

```
5737 \newcommand*\LWR@tabularampersand{%
5738 \LWR@traceinfo{LWR@tabularampersand}%
5739 \ifnumcomp{\value{LWR@tabulardepth}}{>}{0}%
5740 {%
```

If not skipping a multirow cell, close the current data cell.

```
5741 \unskip%
5742 \LWR@closetabledatacell%
```

Move to the next column.

```
5743 \addtocounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
5744 \boolfalse{LWR@tabularcelladded}%
```

Look at the next token to decide multi or single column data tag.

```
5745 \LWR@getmynexttoken%
5746 }%
```

If not inside a tabular, performs the original action:

```
5747 {\LWR@origampmacro}%
5748 }
```

& 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.

### 64.4.3 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.

```
5749 \newcommand*{\LWR@tabularfinishrow}{%
```

If not exiting the tabular, or doing a rule, or have already started a row, finish this row:

```
5750 \ifbool{expr}{%
5751   not bool {LWR@exitingtabular} or%
5752   bool{LWR@doingtbrule} or%
5753   bool{LWR@doingcmidrule} or%
5754   bool{LWR@doinghline} or%
5755   bool{LWR@startedrow}}%
5756 }
```

To locally temporarily turn off `LWR@exitingtabular` so that table data tags will still be generated:

```
5757 \begingroup%
```

If generating a final row for the \bottomrule borders, turn off the @, !, <, and > column output:

```
5758 \ifbool{LWR@exitingtabular}{%
5759     \booltrue{LWR@tabularmutemods}%
5760 }{}%
```

Reenable the table data tags until finished with the final row:

```
5761 \global\boolfalse{LWR@exitingtabular}%
```

Generate table data tags and ampersands until the right edge:

```
5762 \whilebool{expr}{%
5763     test {
5764         \ifnumcomp{value{LWR@tablecolindex}}{<}{value{LWR@tabletotalcols}}
5765     } or %
5766     (%
5767         bool{LWR@intabularmetadata} and%
5768         not bool{LWR@tabularcelladded} and%
5769         test {
5770             \ifnumcomp{value{LWR@tablecolindex}}{=}{value{LWR@tabletotalcols}}
5771         }%
5772     )%
5773 }%
5774 {%
5775     \LWR@tabledatasinglecolumn tag%
```

The following is essentially \LWR@tabularampersand with LWR@emptyatbang added to empty the following cells:

```
5776     \LWR@closetabledata cell%
5777     \addtocounter{LWR@tablecolindex}{1}%
5778     \boolfalse{LWR@tabularcelladded}%
5779     \global\booltrue{LWR@emptyatbang}%
```

Starts the next cell:

```
5780     \ifnumcomp{value{LWR@tablecolindex}}{<}{value{LWR@tabletotalcols}}%
5781     {\LWR@getmynexttoken}%
5782     }{}%
5783 }%
```

Reenable the original LWR@exitingtabular to close the entire table:

```
5784 \endgroup%
5785 \global\boolfalse{LWR@emptyatbang}%
5786 }{}% ifbool{expr
5787 }
```

## 64.5 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendofline

Throws away options \\[dim] or \\\*

\LWR@tabularendofline

```
5788 \NewDocumentCommand{\LWR@tabularendofline}{s o}{%
```

Finish the row:

```
5789 \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}{%
5790 {\LWR@tabularfinishrow}%
5791 {\LWR@closetabledatacell}%
5792 \LWR@htmltag{/tr}\LWR@orignewline%
```

xcolor row color support:

```
5793 \@rowcolor%
```

No longer inside a data cell:

```
5794 \global\booltrue{LWR@intabularmetadata}%
```

Not yet started a table row:

```
5795 \global\boolfalse{LWR@startedrow}%
```

Additional setup:

```
5796 \global\boolfalse{LWR@doinghline}%
5797 \global\boolfalse{LWR@doingtbrule}%
5798 \global\boolfalse{LWR@doingcmidrule}%
5799 \LWR@clearmidrules%
5800 \renewcommand*{\LWR@rowHTMLcolor}{}%
```

Start at first column:

```
5801 \setcounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
5802 \boolfalse{LWR@tabularcelladded}%
```

Look at the next token to decide between single column data tag or a special case:

```
5803 \LWR@getmynexttoken%
5804 }
```

## 64.6 Parsing @, >, <, !, bar columns

Holds the parsed argument for @, >, <, or ! columns:

```
5805 \newcommand*{\LWR@colparameter}{}
```

`\LWR@parseatcolumn` Handles @{text} columns.

```
5806 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
5807 \LWR@traceinfo{at column}%
5808 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token into `\LWR@colparameter`, expanding once:

```
5809 \LWR@traceinfo{about to read the next token:}%
5810 \expandarg%
5811 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
5812 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
5813 \LWR@traceinfo{have now read the next token}%
5814 \ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
5815 {% left edge of the table:
5816   \LWR@traceinfo{at the left edge}%
5817   \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@colparameter}%
5818   \LWR@traceinfo{at the left edge: %
5819   \LWR@getexparray{LWR@colatspec}{leftedge}}%
5820 }%
5821 {% not at the left edge:
5822   \LWR@traceinfo{not at the left edge}%
5823   \LWR@setexparray{LWR@colatspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
5824   \LWR@traceinfo{at \arabic{LWR@tabletotalcols}: %
5825   \LWR@getexparray{LWR@colatspec}{\arabic{LWR@tabletotalcols}}}%
5826 }%
5827 \let\LWR@colparameter\relax%
5828 \booltrue{LWR@validtablecol}%
5829 }
```

`\LWR@parsebangcolumn` Handles `!{text}` columns.

```
5830 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
5831 \LWR@traceinfo{bang column}%
5832 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token into `\LWR@colparameter`, expanding once:

```
5833 \LWR@traceinfo{about to read the next token:}%
5834 \expandarg%
5835 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
5836 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
5837 \LWR@traceinfo{have now read the next token}%
5838 \ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
5839 {% left edge of the table:
5840   \LWR@traceinfo{at the left edge}%
5841   \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@colparameter}%
5842 }%
5843 {% not at the left edge:
5844   \LWR@traceinfo{not at the left edge}%
5845   \LWR@setexparray{LWR@colbangspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
5846   \LWR@traceinfo{bang \arabic{LWR@tabletotalcols}: \LWR@colparameter!}%
5847 }%
5848 \let\LWR@colparameter\relax%
5849 \booltrue{LWR@validtablecol}%
5850 }
```

`\LWR@parsebeforecolumn` Handles `>{text}` columns.

```
5851 \newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
5852 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into `\LWR@colparameter`:

```
5853 \expandarg%
5854 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
5855 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
5856 \LWR@setexparray{LWR@colbefore}{{\arabic{LWR@tabletotalcolsnext}}{\LWR@colparameter}}%
5857 \let\LWR@colparameter\relax%
5858 \booltrue{LWR@validtablecol}%
5859 }
```

\LWR@parseaftercolumn Handles <{text} columns.

```
5860 \newcommand*{\LWR@parseaftercolumn}{%
```

Move to the next token after the '<':

```
5861 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into \LWR@colparameter:

```
5862 \expandarg%
5863 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
5864 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
5865 \LWR@setexparray{LWR@colafterspec}{{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}}%
5866 \let\LWR@colparameter\relax%
5867 \booltrue{LWR@validtablecol}%
5868 }
```

\LWR@parsebarcolumn Handles vertical rules.

```
5869 \newcommand*{\LWR@parsebarcolumn}{%
5870 \LWR@traceinfo{bar column}%
```

Remember the bar at this position:

```
5871 \ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
5872 {% left edge of the table:
5873   \LWR@setexparray{LWR@colbarspec}{leftedge}{tvertbarl}%
5874 }%
5875 {% not at the left edge:
5876   \LWR@setexparray{LWR@colbarspec}{{\arabic{LWR@tabletotalcols}}{tvertbarr}}%
5877 }%
5878 \booltrue{LWR@validtablecol}%
5879 }
```

## 64.7 Parsing ‘l’, ‘c’, or ‘r’ columns

`\LWR@parsenormalcolumn`  $\{ \langle thiscolumn \rangle \}$

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```
5880 \newcommand*{\LWR@parsenormalcolumn}[1]{%
5881 \appto\LWR@tablecolspec{#1}%
5882 \addtocounter{\LWR@tabletotalcols}{1}%
5883 \addtocounter{\LWR@tabletotalcolsnext}{1}%
5884 \LWR@traceinfo{normal column \arabic{\LWR@tabletotalcols}: #1}%
5885 \LWR@setexparray{\LWR@colatspec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5886 \LWR@setexparray{\LWR@colbangspec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5887 \LWR@setexparray{\LWR@colbeforespec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5888 \LWR@setexparray{\LWR@colafterspec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5889 \LWR@setexparray{\LWR@colbarspec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5890 \booltrue{\LWR@validtablecol}%
5891 }
```

## 64.8 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn`  $\{ \langle thiscolumn \rangle \}$  The width will be ignored.

```
5892 \newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type:

```
5893 \LWR@parsenormalcolumn{#1}%
```

Skips the following width token:

```
5894 \addtocounter{\LWR@tablecolspecindex}{1}%
5895 }
```

## 64.9 Parsing ‘D’ columns

From the dcolumn package.

`\LWR@parseDcolumn`  $\{ \langle thiscolumn \rangle \}$  The three parameters will be ignored.

```
5896 \newcommand*{\LWR@parseDcolumn}[1]{%
```



Table 8: Tabular baseline

l	p	m	b	r
			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

Converts to the given column type.

```
5897 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
5898 \addtocounter{LWR@tablecolspecindex}{3}%
5899 }
```

64.10 Parsing the column specifications


 HTML css cannot exactly match the  $\TeX$  concept of a baseline for a table row. Table 8 shows the  $\TeX$  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 9 for details.

Table 9 describes how each kind of column is converted to HTML.

```
Bool LWR@validtablecol True if found a valid table column type.
```

```
5900 \newbool{LWR@validtablecol}
```

```
Bool LWR@opttablecol True if found a table column optional argument.
```

```
5901 \newbool{LWR@opttablecol}
```

```
\LWR@parsetablecols {\<colspecs>}
```

Scans the column specification left to right.

Builds `\LWR@tablecolspec` with the final specification, one column per entry. The final number of cells in each row is stored in `LWR@tabletotalcols`.

Table 9: Tabular HTML column conversions

---

<b>l, r, c:</b>	Converted to table cells without paragraph tags. Uses css <code>vertical-align:middle</code> so that top or bottom-aligned cells may go above or below this cell.
<b>p:</b>	Converted to table cells with paragraph tags. Ref: Table 8, $\text{\LaTeX}$ places the top line of a parbox aligned with the rest of the text line, so css <code>vertical-align:bottom</code> is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the css may be desired, changing <code>td.tdp</code> , <code>td.tdP</code> , <code>td.tdprule</code> , and <code>td.tdPrule</code> to <code>vertical-align: middle</code> . Another possibility is to change L,R,C, and P to <code>vertical-align: top</code> and not worry about the alignment of B and M cells or trying to approximate $\text{\LaTeX}$ baselines.
<b>m:</b>	With paragraph tags, css <code>vertical-align:middle</code> .
<b>b:</b>	With paragraph tags, css <code>vertical-align:top</code> so that the bottom of the text is closest to the middle of the text line.
<b>P, M, B:</b>	Horizontally-centered versions.
<b>S:</b>	Converted to 'r'. Ignores optional argument. From the <code>siunitx</code> package.
<b>D:</b>	Converted to 'c'. From the <code>dcolumn</code> package.
<b>@, !, &gt;, &lt;:</b>	One each, in that order.
<b> :</b>	Vertical rule.
<b>Unknown:</b>	Converted to 'l'.
<b>\newcolumn:</b>	Currently treated as unknown.

---

```
5902 \newcommand*{\LWR@parsetablecols}[1]{%
5903 \LWR@traceinfo{\LWR@parsetablecols}%
```

Remember the original supplied column spec:

```
5904 \renewcommand*{\LWR@origcolspec}{#1}%
```

Remove spaces:

```
5905 \expandarg%
5906 \StrSubstitute{\LWR@origcolspec}{ }{}[\LWR@origcolspec]%
```

Clear the parsed resulting column spec:

```
5907 \renewcommand*{\LWR@tablecolspec}{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
5908 \setcounter{\LWR@tabletotalcols}{0}%
5909 \setcounter{\LWR@tabletotalcolsnext}{1}%
5910 \LWR@setexparray{\LWR@colatspec}{leftedge}{}%
5911 \LWR@setexparray{\LWR@colatspec}{1}{}%
5912 \LWR@setexparray{\LWR@colatspec}{2}{}%
5913 \LWR@setexparray{\LWR@colatspec}{3}{}%
5914 \LWR@setexparray{\LWR@colbangspec}{leftedge}{}%
5915 \LWR@setexparray{\LWR@colbangspec}{1}{}%
5916 \LWR@setexparray{\LWR@colbangspec}{2}{}%
5917 \LWR@setexparray{\LWR@colbangspec}{3}{}%
5918 \LWR@setexparray{\LWR@colbeforespec}{1}{}%
5919 \LWR@setexparray{\LWR@colbeforespec}{2}{}%
5920 \LWR@setexparray{\LWR@colbeforespec}{3}{}%
5921 \LWR@setexparray{\LWR@colafterspec}{1}{}%
5922 \LWR@setexparray{\LWR@colafterspec}{2}{}%
5923 \LWR@setexparray{\LWR@colafterspec}{3}{}%
5924 \LWR@setexparray{\LWR@colbarspec}{leftedge}{}%
5925 \LWR@setexparray{\LWR@colbarspec}{1}{}%
5926 \LWR@setexparray{\LWR@colbarspec}{2}{}%
5927 \LWR@setexparray{\LWR@colbarspec}{3}{}%
```

Starting at the first column specification:

```
5928 \setcounter{\LWR@tablecolspecindex}{1}%
```

Place the colspecs string length into \LWR@strresult, and remember the number of characters in the column specification:

```
5929 \expandarg%
```

```

5930 \StrLen{\LWR@origcolspec}{\LWR@strresult}%
5931 \fullexpandarg%
5932 \LWR@traceinfo{original column spec length: \LWR@strresult}%
5933 \setcounter{\LWR@tablecolspecwidth}{\LWR@strresult}%

```

Haven't seen any optional arguments so far

```

5934 \boolfalse{\LWR@opttablecol}%

```

Scan through the column specifications:

```

5935 \whilebool{expr}{%
5936     not test{%
5937         \ifnumcomp{\value{\LWR@tablecolspecindex}}{>}{\value{\LWR@tablecolspecwidth}}{%
5938             }%
5939 }%
5940 {%

```

Place the next single-character column type into \LWR@strresult:

```

5941 \expandarg%
5942 \StrChar{\LWR@origcolspec}{\arabic{\LWR@tablecolspecindex}}{\LWR@strresult}%
5943 \LWR@traceinfo{position \arabic{\LWR@tablecolspecindex}: \LWR@strresult}%
5944 \fullexpandarg%

```

Not yet found a valid column type:

```

5945 \boolfalse{\LWR@validtablecol}%

```

Skip over any optional arguments, such as siunitx S column:

```

5946 \IfStrEq{\LWR@strresult}{[]}{\booltrue{\LWR@opttablecol}}{}%

```

Throw away anything found inside the optional argument:

```

5947 \ifbool{\LWR@opttablecol}%
5948 {}% inside an optional argument
5949 {% not an optional tabular argument

```

Not inside an optional argument, so consider the column type:

```

5950 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{}%
5951 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{}%
5952 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{}%
5953 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{}%
5954 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{}%
5955 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{}%
5956 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{}%

```

```

5957 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{}%
5958 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
5959 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
5960 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
5961 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
5962 \IfStrEq{\LWR@strresult}{|}{\LWR@parsebarcolumn}{}%
5963 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
5964 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
5965 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%

```

From the dcolumn package:

```

5966 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{}%

```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```

5967 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%

```

---

Many people define centered versions “P”, “M”, and “B”:

```

\newcolumnntype{P}[1]{>{\centering\arraybackslash}p{#1}}

```

---

```

5968 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
5969 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
5970 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%

```

If this column was an invalid column type, convert it to an l column:

```

5971 \ifbool{LWR@validtablecol}{}{}%
5972   \LWR@traceinfo{invalid column type: \LWR@strresult}%
5973   \LWR@parsenormalcolumn{l}%
5974 }%
5975 }% not an optional column argument

```

If read the closing bracket, no longer inside the optional argument:

```

5976 \IfStrEq{\LWR@strresult}{]}{\boolfalse{LWR@opttablecol}}{}%

```

Move to the next character:

```

5977 \addtocounter{LWR@tablecolspecindex}{1}%
5978 }% whiledo
5979 \LWR@traceinfo{LWR@parsetablecols: done}%
5980 }%

```

### 64.11 colortbl and xparse tabular color support

These macros provide a minimal emulation of some colortbl macros which might appear between table cells. If colortbl is loaded, these macros will be replaced with functional versions.

For each of the HTML colors below, the text for the HTML color is set if requested, but the macro is empty if none has been set.

`\rownum` Reserve a counter register.

```
5981 \ifundefined{rownum}{\newcount\rownum}{}
```

`\@rowcolors` Emulated in case xcolor is not used.

```
5982 \newcommand*{\@rowcolors}{}{}
```

`\@rowc@lors` Emulated in case xcolor is not used.

```
5983 \newcommand*{\@rowc@lors}{}{}
```

`\LWR@xcolorrowHTMLcolor` Emulated xcolor row color.

```
5984 \newcommand*{\LWR@xcolorrowHTMLcolor}{}{}
```

`\LWR@columnHTMLcolor` HTMLstyle code for the column color.

```
5985 \newcommand*{\LWR@columnHTMLcolor}{}{}
```

`\LWR@rowHTMLcolor` HTMLstyle code for the row color.

```
5986 \newcommand*{\LWR@rowHTMLcolor}{}{}
```

`\LWR@cellHTMLcolor` HTMLstyle code for the cell color.

```
5987 \newcommand*{\LWR@cellHTMLcolor}{}{}
```

`\LWR@ruleHTMLcolor` HTMLstyle code for the cell color.

```
5988 \newcommand*{\LWR@ruleHTMLcolor}{}{}
```

Inside an HTML tabular, each of `\columncolor` etc. is `\let` to the `\LWR@HTML` versions below. When colortbl is loaded, its definitions override the following.

`\columncolor` [*⟨model⟩*] {*⟨color⟩*} [*⟨left overhang⟩*] [*⟨right overhang⟩*]

`\LWR@HTMLcolumncolor` [*⟨model⟩*] {*⟨color⟩*} [*⟨left overhang⟩*] [*⟨right overhang⟩*]

5989 `\NewDocumentCommand{\LWR@HTMLcolumncolor}{0{named} m o o}{}`

`\rowcolor` [*⟨model⟩*] {*⟨color⟩*} [*⟨left overhang⟩*] [*⟨right overhang⟩*]

`\LWR@HTMLrowcolor` [*⟨model⟩*] {*⟨color⟩*} [*⟨left overhang⟩*] [*⟨right overhang⟩*] Used before starting a tabular data cell, thus `\LWR@getmynexttoken`.

5990 `\NewDocumentCommand{\LWR@HTMLrowcolor}{0{named} m o o}{\LWR@getmynexttoken}`

`\cellcolor` [*⟨model⟩*] {*⟨color⟩*} [*⟨left overhang⟩*] [*⟨right overhang⟩*]

`\LWR@HTMLcellcolor` [*⟨model⟩*] {*⟨color⟩*} [*⟨left overhang⟩*] [*⟨right overhang⟩*]

5991 `\NewDocumentCommand{\LWR@HTMLcellcolor}{0{named} m o o}{}`

`\arrayrulecolor` [*⟨model⟩*] {*⟨color⟩*}

The version for use outside a tabular.

5992 `\newcommand{\arrayrulecolor}[2][named]{}`

`\LWR@HTMLarrayrulecolor` [*⟨model⟩*] {*⟨color⟩*}

The version for use inside a tabular.

5993 `\newcommand{\LWR@HTMLarrayrulecolor}[2][named]{\LWR@getmynexttoken}`

`\doublerulesepcolor` [*⟨model⟩*] {*⟨color⟩*}

The version for use outside a tabular.

5994 `\newcommand{\doublerulesepcolor}[2][named]{}`

`\LWR@HTMLdoublerulesepcolor` [*⟨model⟩*] {*⟨color⟩*}

The version for use inside a tabular.

5995 `\newcommand{\LWR@HTMLdoublerulesepcolor}[2][named]{\LWR@getmynexttoken}`

## 64.12 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.

```
5996 \newcommand*{\LWR@maybenewtablerow}
5997 {%
5998 \ifbool{\LWR@startedrow}%
5999 {}% started the row
6000 {% not started the row
```

Remember that now have started the row:

```
6001 \global\booltrue{\LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```
6002 \global\booltrue{\LWR@intabularmetadata}%
6003 \ifbool{\LWR@doinghline}%
6004 {%
6005 \ifdefvoid{\LWR@ruleHTMLcolor}{%
6006 \LWR@htmltag{tr class="hline" }%
6007 }{%
6008 \LWR@htmltag{%
6009 tr class="hline" %
6010 style="border-top: 1px solid \#\LWR@ruleHTMLcolor "%
6011 }%
6012 }%
6013 \LWR@orignewline%
6014 }%
6015 {% not doing hline
6016 \ifbool{\LWR@doingtbrule}%
6017 {%
6018 \ifdefvoid{\LWR@ruleHTMLcolor}{%
6019 \LWR@htmltag{tr class="tbrule"}%
6020 }{%
6021 \LWR@htmltag{%
6022 tr class="tbrule" %
6023 style="border-top: 1px solid \#\LWR@ruleHTMLcolor "%
6024 }%
6025 }%
6026 \LWR@orignewline%
6027 }%
6028 {\LWR@htmltag{tr}\LWR@orignewline}%
6029 }% end of not doing hline
6030 }% end of not started the row
6031 }
```



### 64.13 Printing vertical bar tags

`\LWR@printbartag`  $\{\langle index \rangle\}$

Adds to a tabular data cell an HTML class name for a left/right vertical bar.

```

6032 \newcommand*\LWR@printbartag}[1]{%
6033 \LWR@traceinfo{\LWR@printbartag !#1!}%
6034 \ifboolexpr{bool{\LWR@tabularmutemods} or bool{\LWR@emptyatbang}}%
6035 {}% muting or empty
6036 {% not muting
6037     \edef\LWR@tempone{\LWR@getexparray{\LWR@colbarspec}{#1}}%
6038     \ifdefempty{\LWR@tempone}{\LWR@tempone}%
6039 }% not muting
6040 \LWR@traceinfo{\LWR@printbartag done}%
6041 }
```

### 64.14 Printing at or bang tags

`\LWR@printatbang`  $\{\langle at-or-bang \rangle\} \{\langle index \rangle\}$

```

6042 \newcommand*\LWR@printatbang}[2]{%
```

Fetch the column at or bang spec:

```

6043 \edef\LWR@atbangspec{\LWR@getexparray{\LWR@col#1spec}{#2}}%
6044 \LWR@traceinfo{atbang: #2 !\LWR@atbangspec!}%
```

Only generate if is not empty;

```

6045 \ifdefempty{\LWR@atbangspec}%
6046 {}%
6047 {% not empty
6048     \LWR@htmltag{%
6049         td class="td#1%
6050             \LWR@subaddcmidruletrim{}{}%
6051             \LWR@printbartag{#2}%
6052             "%
6053             \LWR@tdstartstyles%
6054             \LWR@addcmidrulewidth%
6055             \LWR@addtabularrulecolors%
6056             \LWR@tdendstyles%
6057     }%
```

Create an empty cell if muting for the \bottomrule:

```

6058 \ifbool{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}{%
6059   }{%
6060   {\LWR@atbangspec}%
6061 %
6062 \LWR@htmltag{/td}\LWR@orignewline%
6063 \global\booltrue{LWR@tabularcelladded}%
6064 }% not empty
6065 }%

```

\LWR@addleftmostbartag

```

6066 \newcommand*{\LWR@addleftmostbartag}{%
6067 \ifnumcomp{\value{LWR@tablecolindex}}{=}{1}{%
6068   \LWR@printbartag{leftedge}%
6069 }{}%
6070 }

```

\LWR@tabularleftedge

```

6071 \newcommand*{\LWR@tabularleftedge}{%
6072 \ifnumcomp{\value{LWR@tablecolindex}}{=}{1}{%
6073   {%
6074     \LWR@printatbang{at}{leftedge}%
6075     \LWR@printatbang{bang}{leftedge}%
6076   }% left edge
6077   {}% not left edge
6078 }

```

## 64.15 Data opening tag

\LWR@thiscolspec Temporary storage.

```

6079 \newcommand*{\LWR@thiscolspec}{}

```

\LWR@tabledatasinglecolumnntag Print a table data opening tag with style for alignment and color.

```

6080 \newcommand*{\LWR@tabledatasinglecolumnntag}{%
6081   {%
6082 \LWR@traceinfo{LWR@tabledatasinglecolumnntag}%
6083 \LWR@maybenewtablerow%

```

Don't start a new paragraph tag if have already started one:

```
6084 \ifbool{LWR@intabularmetadata}%
6085 {%
```

If have found the end of tabular command, do not create the next data cell:

```
6086 \ifbool{LWR@exitingtabular}{}%
6087 {% not exiting tabular
```

Print the @ and ! contents before first column:

```
6088 \LWR@tabularleftedge%
```

Fetch the current column's alignment character into \LWR@strresult:

```
6089 \StrChar{\LWR@tablecolspec}{\arabic{LWR@tablecolindex}}[\LWR@strresult]%
```

print the start of a new table data cell:

```
6090 \LWR@traceinfo{LWR@tabledatasinglecolumn tag: about to print td tag}%
6091 \LWR@htmltag{td class="td%
```

append this column's spec:

```
6092 \LWR@strresult%
```

If this column has a cmidrule, add "rule" to the end of the HTML class tag. Also add vertical bar tags.

```
6093 \LWR@addcmidruletrim%
6094 \LWR@addleftmostbartag%
6095 \LWR@printbartag{\arabic{LWR@tablecolindex}}%
6096 "%
```

Add styles for rules, alignment:

```
6097 \LWR@tdstartstyles%
6098 \LWR@addcmidrulewidth%
6099 \StrChar{\LWR@tablecolspec}{\arabic{LWR@tablecolindex}}[\LWR@thiscolspec]%
6100 \LWR@addformatwpalignment{\LWR@thiscolspec}%
```

Add styles for cell and rule colors:

```
6101 \LWR@addtabulararrowcolor%
6102 \LWR@addtabularrulecolors%
```

```

6103      \LWR@tdendstyles%
6104      }%
6105      \LWR@traceinfo{LWR@tabledatasinglecolumn: done printing td tag}%

```

If this is a p, m, b, or X column, allow paragraphs:

```

6106      \ifboolexpr{%
6107          test{ \ifdefstring{LWR@strresult}{p} } or
6108          test{ \ifdefstring{LWR@strresult}{m} } or
6109          test{ \ifdefstring{LWR@strresult}{b} } or
6110          test{ \ifdefstring{LWR@strresult}{P} } or
6111          test{ \ifdefstring{LWR@strresult}{M} } or
6112          test{ \ifdefstring{LWR@strresult}{B} } or
6113          test{ \ifdefstring{LWR@strresult}{X} }
6114      }%
6115      {% allow pars
6116          \LWR@traceinfo{LWR@tabledatasinglecolumn: about to LWR@startpars}%
6117          \global\booltrue{LWR@tableparcell}%
6118          \LWR@startpars%
6119          \LWR@traceinfo{LWR@tabledatasinglecolumn: done with LWR@startpars}%
6120      }% allow pars
6121      }% no pars

```

Print the > contents unless muted for the \bottomrule:

```

6122      \ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
6123      {%
6124      {%
6125          \LWR@getexparray{LWR@colbefore:spec}{\arabic{LWR@tablecolindex}}%
6126      }%
6127      \global\boolfalse{LWR@intabularmetadata}%
6128      }% not exiting tabular
6129 }{% in tabular metadata
6130 \LWR@traceinfo{LWR@tabledatasinglecolumn: done}%
6131 }%

```

## 64.16 Midrules

**LWR@midrules** LWR@midrules is a data array (section 37) 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 37) of columns containing 1 if a midrule should be left trimmed for each column.

**LWR@trimrrules** LWR@trimrrules is a data array (section 37) 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.

```
6132 \newcounter{LWR@midrulecounter}
```

Len \LWR@heavyrulewidth The default width of the rule.

```
6133 \newlength{\LWR@heavyrulewidth}
6134 \setlength{\LWR@heavyrulewidth}{.08em}
```

Len \LWR@lightrulewidth The default width of the rule.

```
6135 \newlength{\LWR@lightrulewidth}
6136 \setlength{\LWR@lightrulewidth}{.05em}
```

Len \LWR@cmidrulewidth The default width of the rule.

```
6137 \newlength{\LWR@cmidrulewidth}
6138 \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.

```
6139 \newlength{\LWR@thiscmidrulewidth}
6140 \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.

```
6141 \newcommand*{\LWR@clearmidrules}
6142 {%
6143 \setcounter{LWR@midrulecounter}{1}%
6144 \whileboolexpr{%
6145     not test{%
6146         \ifnumcomp{\value{LWR@midrulecounter}}{>}{\value{LWR@tablecolspecwidth}}%
6147     }%
6148 }%
6149 {%
6150 \LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{Opt}%
6151 \setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
6152 \LWR@setexparray{LWR@trimlrules}{\arabic{LWR@midrulecounter}}{}%
6153 \LWR@setexparray{LWR@trimrrules}{\arabic{LWR@midrulecounter}}{}%
6154 \addtocounter{LWR@midrulecounter}{1}%
6155 }%
6156 }
```

`\LWR@subcmidrule`  $\{\langle width \rangle\} \{\langle trim \rangle\} \{\langle leftcolumn \rangle\} \{\langle rightcolumn \rangle\}$

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.

```

6157 \newcommand*{\LWR@subcmidrule}[4]{%
6158 \setcounter{LWR@midrulecounter}{#3}%
6159 \whileboolexpr{%
6160     not test {%
6161         \ifnumcomp{\value{LWR@midrulecounter}}{>}{#4}%
6162     }%
6163 }%
6164 {%
6165     \LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{#1}%
6166     \addtocounter{LWR@midrulecounter}{1}%
6167 }% whiledo
6168 \IfSubStr{#2}{l}{\LWR@setexparray{LWR@trimlrules}{#3}{l}}{}%
6169 \IfSubStr{#2}{r}{\LWR@setexparray{LWR@trimrrules}{#4}{r}}{}%
6170 \booltrue{LWR@doingcmidrule}%
6171 }
```

`\LWR@docmidrule`  $[\langle width \rangle] \{\langle trim \rangle\} \{\langle leftcolumn-rightcolumn \rangle\}$

Marks `LWR@midrules` array elements to be a non-zero width from left to right columns. Also marks trimming for the L and/or R columns.

```

6172 \NewDocumentCommand{\LWR@docmidrule}{0{\LWR@cmidrulewidth} D(){} >\SplitArgument{1}{-}m}%
6173 {\LWR@subcmidrule{#1}{#2}{#3}}
```

Used to compute margins, tabular trims:

```

6174 \newlength{\LWR@templengthone}%
6175 \newlength{\LWR@templengthtwo}%
6176 \newlength{\LWR@templengththree}%
```

Used to add a style to a table data cell:

```

6177 \newboolean{LWR@tdhavecellstyle}
```

`\LWR@tdstartstyles` Begins possibly adding a table data cell style.

```

6178 \newcommand*{\LWR@tdstartstyles}{\global\boolfalse{LWR@tdhavecellstyle}}
```

`\LWR@tdaddstyle` Starts adding a table data cell style.

```
6179 \newcommand*{\LWR@tdaddstyle}{%
6180 \ifbool{\LWR@tdhavecellstyle}%
6181 {; }%
6182 { style="%
6183 \booltrue{\LWR@tdhavecellstyle}%
6184 }
```

`\LWR@tdendstyles` Finishes possibly adding a table data cell style. Prints the closing quote.

```
6185 \newcommand*{\LWR@tdendstyles}{%
6186 \ifbool{\LWR@tdhavecellstyle}{%
6187     "%
6188     \global\boolfalse{\LWR@tdhavecellstyle}%
6189 }{}}%
6190 }
```

`\LWR@subaddcmidruletrim` `{<lefttrim>}{<righttrim>}` Adds a `\cmidrule` with optional trim.

```
6191 \newcommand*{\LWR@subaddcmidruletrim}[2]{%
6192 \setlength{\LWR@templengthone}{%
6193     \LWR@getexparray{\LWR@midrules}{\arabic{\LWR@tablecolindex}}}%
6194 }%
6195 \ifdimcomp{\LWR@templengthone}{>}{Opt}%
6196 {%
```

Print the class without left and right trim letters appended:

```
6197     \LWR@origtilde tdrule#1#2%
```

Remember the width of the rule:

```
6198     \setlength{\LWR@thiscmidrulewidth}{\LWR@templengthone}%
6199 }%
6200 {%
6201     \setlength{\LWR@thiscmidrulewidth}{Opt}%
6202 }%
6203 }
```

`\LWR@addcmidruletrim` Adds left or right trim to a `\cmidrule`.

```
6204 \newcommand*{\LWR@addcmidruletrim}{%
6205 \LWR@subaddcmidruletrim%
6206 {\LWR@getexparray{\LWR@trimlrules}{\arabic{\LWR@tablecolindex}}}%
6207 {\LWR@getexparray{\LWR@trimrrules}{\arabic{\LWR@tablecolindex}}}%
6208 }
```

`\LWR@addrulewidth`  $\{\langle thiswidth \rangle\} \{\langle defaultwidth \rangle\}$

If not default width, add a custom style with width and color depending on `thiswidth`.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
6209 \newcommand{\LWR@addrulewidth}[2]{%
```

Only add a custom width if `thiswidth` is different than the `defaultwidth`, or if a color is being used:

```
6210 \ifboolexpr{%
6211     test{\ifdimcomp{#1}={}\{0pt\}} or
6212     (
6213         ( test{\ifdimcomp{#1}={}\{#2\}} and not bool{FormatWP} )
6214         and ( test {\ifdefvoid{\LWR@ruleHTMLcolor}} )
6215     )
6216 }%
6217 {}% default width and color
6218 {}% custom width and/or color
```

Ensure that the width is wide enough to display in the browser:

```
6219     \LWR@forceminwidth{#1}%
```

Begin adding another style:

```
6220     \LWR@tdaddstyle%
```

The style itself:

```
6221     border-top:\LWR@printlength{\LWR@atleastonept} solid %
```

If default gray, the darkness of the color depends on the thickness of the rule:

```
6222     \ifdefvoid{\LWR@ruleHTMLcolor}{%
6223         \ifdimcomp{#1}{<}\{\LWR@lightrulewidth}%
6224         {\#AOA0A0}%
6225         {% lightrule or heavier
6226             \ifdimcomp{#1}{<}\{\LWR@heavyrulewidth}%
6227             {\#808080}%
6228             {black}%
6229         }% lightrule or heavier
6230     }{%
6231         \#\LWR@ruleHTMLcolor%
6232     }
6233 }% custom width and/or color
```



6234 }

`\LWR@addcmidrulewidth` Adds a style for the rule width.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
6235 \newcommand{\LWR@addcmidrulewidth}{%
6236 \LWR@addrulewidth{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
6237 }
```

`\LWR@WPcell` `{\langle text-align\rangle}{\langle vertical-align\rangle}`

```
6238 \newcommand*{\LWR@WPcell}[2]{%
6239 \LWR@tdaddstyle%
6240 \LWR@origmbox{text-align:#1}; \LWR@origmbox{vertical-align:#2}%
6241 }
```

`\LWR@addformatwpalignment` If `FormatWP`, adds a style for the alignment.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
6242 \newcommand*{\LWR@addformatwpalignment}[1]{%
6243 \ifbool{FormatWP}{%
6244 \IfSubStr{#1}{l}{\LWR@WPcell{left}{middle}}{}%
6245 \IfSubStr{#1}{c}{\LWR@WPcell{center}{middle}}{}%
6246 \IfSubStr{#1}{r}{\LWR@WPcell{right}{middle}}{}%
6247 \IfSubStr{#1}{p}{\LWR@WPcell{left}{bottom}}{}%
6248 \IfSubStr{#1}{m}{\LWR@WPcell{left}{middle}}{}%
6249 \IfSubStr{#1}{b}{\LWR@WPcell{left}{top}}{}%
6250 \IfSubStr{#1}{P}{\LWR@WPcell{center}{bottom}}{}%
6251 \IfSubStr{#1}{M}{\LWR@WPcell{center}{middle}}{}%
6252 \IfSubStr{#1}{B}{\LWR@WPcell{center}{top}}{}%
6253 }{}%
6254 }
```

## 64.17 Cell colors

`\LWR@addtabulararrowcolor` Adds a cell's row color style, if needed.

No color is added for the final row of empty cells which finishes each tabular.

```
6255 \newcommand*{\LWR@addtabulararrowcolor}{%
6256 \ifbool{LWR@tabularmutemods}{}{}%
6257 \ifdefined{\LWR@rowHTMLcolor}{%
```

```

6258      \ifdefvoid{\LWR@xcolorrowHTMLcolor}{}%
6259      {% xcolor row color
6260          \LWR@tdaddstyle%
6261          background:\#\LWR@xcolorrowHTMLcolor%
6262      }%
6263  }%
6264  {% explicit row color
6265      \LWR@tdaddstyle%
6266      background:\#\LWR@rowHTMLcolor%
6267  }%
6268 }%
6269 }

```

`\LWR@addtabularhrulecolor` Adds a cell's horizontal rule color style, if needed.

```
6270 \newcommand*{\LWR@addtabularhrulecolor}{%
```

If either form of horizontal rule is requested:

```

6271 \ifboolexpr {
6272     bool{\LWR@doinghline} or
6273     bool{\LWR@doingtbrule}
6274 }{%

```

And if there is a custom horizontal color:

```

6275     \ifdefvoid{\LWR@ruleHTMLcolor}{}%
6276     {%
6277         \LWR@tdaddstyle%
6278         border-top: 1px solid \#\LWR@ruleHTMLcolor%
6279     }{}%
6280 }{}%
6281 }

```

`\LWR@addtabularrulecolors` Adds a cell's rule color styles, if needed.

No color is added for the final row of empty cells which finishes each tabular.

```
6282 \newcommand*{\LWR@addtabularrulecolors}{%
```

Custom horizontal rule color:

```
6283 \LWR@addtabularhrulecolor%
```

No vertical rules if finishing the tabular with a row of empty cells:

```
6284 \ifbool{\LWR@tabularmutemods}{}{%
```

If at the leftmost cell, possibly add a leftmost vertical rule:

```
6285 \ifnumequal{\value{LWR@tablecolindex}}{1}{%
```

Fetch the left edge's vertical bar specification:

```
6286 \edef\LWR@tempone{\LWR@getexparray{LWR@colbarspec}{leftedge}}%
```

Add a custom style if a vertical bar was requested:

```
6287 \ifdefstring{\LWR@tempone}{tvertbarl}{%
6288 \LWR@tdaddstyle%
6289 border-left: 1px solid \#\LWR@vertruleHTMLcolor%
6290 }{}%
6291 }{}%
```

Possibly add a right vertical rule for this cell:

```
6292 \edef\LWR@tempone{\LWR@getexparray{LWR@colbarspec}{\arabic{LWR@tablecolindex}}}%
6293 \ifdefstring{\LWR@tempone}{tvertbarr}{%
```

Add a custom style if a vertical bar was requested:

```
6294 \LWR@tdaddstyle%
6295 border-right: 1px solid \#\LWR@vertruleHTMLcolor%
6296 }{}%
6297 }%
6298 }
```

**Ctrl LWR@cellcolordepth** Counts how many cell color <div>s were added to the current tabular data cell.

```
6299 \newcounter{LWR@cellcolordepth}
```

**\LWR@subaddtabularcellcolor** *{\HTML color}*

```
6300 \newcommand*{\LWR@subaddtabularcellcolor}[1]{%
6301 \LWR@htmltag{div class="cellcolor" style="%
6302 background:\#{ }#1 %
6303 " }%
6304 \addtocounter{LWR@cellcolordepth}{1}%
6305 }
```

**\LWR@addtabularcellcolor** Adds a cell color style, if needed.

```
6306 \newcommand*{\LWR@addtabularcellcolor}{%
6307 \ifdefvoid{\LWR@cellHTMLcolor}%
6308 }
```

```

6308 {%
6309     \ifdefvoid{\LWR@rowHTMLcolor}%
6310     {%
6311         \ifdefvoid{\LWR@xcolorrowHTMLcolor}%
6312         {%
6313             \ifdefvoid{\LWR@columnHTMLcolor}%
6314             {%
6315                 {\LWR@subadddtabularcellcolor{\LWR@columnHTMLcolor}}}%
6316             }%
6317             {\LWR@subadddtabularcellcolor{\LWR@xcolorrowHTMLcolor}}}%
6318         }%
6319         {\LWR@subadddtabularcellcolor{\LWR@rowHTMLcolor}}}%
6320     }%
6321     {\LWR@subadddtabularcellcolor{\LWR@cellHTMLcolor}}}%
6322 }

```

## 64.18 Multicolumns

### 64.18.1 Parsing multicolumns

```
6323 \newcounter{\LWR@tablemulticolwidth}
```

Indexes into the multicolumn specification:

```
6324 \newcounter{\LWR@tablemulticolspos}
```

Remembers multicolumn vertical rules if found in the column spec.

```
6325 \newbool{\LWR@mccolvertbarl}
```

```
6326 \newbool{\LWR@mccolvertbarr}
```

`\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.

```

6327 \newcommand*{\LWR@printmccoltype}[1]{%
6328 \LWR@traceinfo{\LWR@printmccoltype -#1-}%

```

Get one token of the column spec:

```
6329 \StrChar{#1}{\arabic{\LWR@tablemulticolspos}}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```

6330 \IfStrEq{\LWR@strresult}{l}{l}{}%
6331 \IfStrEq{\LWR@strresult}{c}{c}{}%
6332 \IfStrEq{\LWR@strresult}{r}{r}{}%
6333 \IfStrEq{\LWR@strresult}{p}{p}{}%
6334 \IfStrEq{\LWR@strresult}{m}{m}{}%
6335 \IfStrEq{\LWR@strresult}{b}{b}{}%
6336 \IfStrEq{\LWR@strresult}{P}{P}{}%
6337 \IfStrEq{\LWR@strresult}{M}{M}{}%
6338 \IfStrEq{\LWR@strresult}{B}{B}{}%
6339 \IfStrEq{\LWR@strresult}{S}{r}{}%
6340 \IfStrEq{\LWR@strresult}{X}{p}{}%

6341 \IfStrEq{\LWR@strresult}{|}{}%
6342   \ifnumcomp{value{\LWR@tablemulticolspos}}{=}{1}% left edge?
6343     {\booltrue{\LWR@mcolvertbarl}}% left edge
6344     {\booltrue{\LWR@mcolvertbarr}}% not left edge
6345 }{}%
6346 \LWR@traceinfo{lwarp@printmccoltype done}%
6347 }

```

`\LWR@multicolpartext` Print the data with paragraph tags:

```

6348 \newcommand*{\LWR@multicolpartext}{%
6349 \LWR@startpars%
6350 \LWR@multicoltext%
6351 \LWR@stoppars%
6352 }

```

`\LWR@multicolother` `{\colspec}` For @, !, >, <, print the next token without paragraph tags:

```

6353 \newcommand*{\LWR@multicolother}[1]{%
6354 \addtocounter{\LWR@tablemulticolspos}{1}%
6355 \StrChar{#1}{\arabic{\LWR@tablemulticolspos}}[\LWR@strresult]%
6356 \LWR@strresult%

```

A valid column data type was found:

```

6357 \booltrue{\LWR@validtablecol}%
6358 }

```

`\LWR@multicolskip` Nothing to print for this column type.

```

6359 \newcommand*{\LWR@multicolskip}{%

```

A valid column data type was found:

```
6360 \booltrue{LWR@validtablecol}%
6361 }
```

\LWR@printmccoldata {<colspec>} Print the data for any valid column type found.

```
6362 \newcommand*{\LWR@printmccoldata}[1]{%
6363 \LWR@traceinfo{lwr@printmccoldata -#1}%
```

Not yet found a valid column type:

```
6364 \boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
6365 \StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
6366 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
6367 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
6368 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
6369 \IfStrEq{\LWR@strresult}{D}{}%
6370 \addtocounter{LWR@tablemulticolspos}{3}% skip parameters
6371 \LWR@multicoltext%
6372 }{}%
6373 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolparttext}{}%
6374 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolparttext}{}%
6375 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolparttext}{}%
6376 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolparttext}{}%
6377 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolparttext}{}%
6378 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolparttext}{}%
6379 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolparttext}{}%
6380 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolparttext}{}%
6381 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
6382 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@multicolother{#1}}{}%
6383 \IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}{}%
6384 \IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}{}%
6385 \IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}{}%
```

If an invalid column type:

```
6386 \ifbool{LWR@validtablecol}{\LWR@multicoltext}%
```

Tracing:

```
6387 \LWR@traceinfo{lwr@printmccoldata done}%
6388 }
```

`\parsemulticolumnalignment`  $\{\langle 1: colspec \rangle\} \{\langle 2: printresults \rangle\}$

Scan the multicolumn specification and execute the printfunction for each entry.

Note that the spec for a `p{spec}` column, or @, >, <, is a token list which will NOT match l, c, r, or p.

```
6389 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
6390 \setcounter{LWR@tablemulticolspos}{1}%
6391 \StrLen{#1}[\LWR@strresult]%
6392 \setcounter{LWR@tablemulticolwidth}{\LWR@strresult}%
```

Scan across the tokens in the column spec:

```
6393 \whileboolexpr{%
6394     not test {%
6395         \ifnumcomp{\value{LWR@tablemulticolspos}}{>}{\value{LWR@tablemulticolwidth}}%
6396     }
6397 }%
6398 {%
```

Execute the assigned print function for each token in the column spec:

```
6399 #2{#1}%
```

Move to the next token in the column spec:

```
6400 \addtocounter{LWR@tablemulticolspos}{1}%
6401 }%
6402 }
```

### 64.18.2 Multicolumn factored code

`\LWR@addmulticolvertrulecolor`

```
6403 \newcommand*{\LWR@addmulticolvertrulecolor}{%
```

No vertical rules if finishing the tabular with a row of empty cells:

```
6404 \ifbool{LWR@tabularmutemods}{}{%
```

Left side:

```
6405     \ifbool{LWR@mcolvertbarl}{%
6406         \LWR@tdaddstyle%
6407         border-left: 1px solid \#\LWR@vertruleHTMLcolor%
6408     }{}%
```

Right side:

```
6409 \ifbool{LWR@mcolvertbarr}{%
6410 \LWR@tdaddstyle%
6411 border-right: 1px solid \#\LWR@vertruleHTMLcolor%
6412 }{}%
6413 }%
6414 }
```

```
6415 \newcommand{\LWR@multicoltext}{}%
```

To find multicolumn right trim:

```
6416 \newcounter{LWR@lastmulticolumn}
```

```
\LWR@domulticolumn [<1: vpos>] [<2: #rows>] [<3: numLaTeXcols>] [<4: numHTMLcols>] [<5: colspec>]
[<6: text>]
```

```
6417 \NewDocumentCommand{\LWR@domulticolumn}{o o m m +m}{%
6418 \LWR@traceinfo{LWR@domulticolumn -#1- -#2- -#4- -#5-}%
```

Remember the text to be inserted, and remember that a valid column type was found:

```
6419 \renewcommand{\LWR@multicoltext}{%
6420 #6%
6421 \booltrue{LWR@validtablecol}%
6422 }%
```

Compute the rightmost column to be included. This is used to create the right trim.

```
6423 \setcounter{LWR@lastmulticolumn}{\value{LWR@tablecolindex}}%
6424 \addtocounter{LWR@lastmulticolumn}{#3}%
6425 \addtocounter{LWR@lastmulticolumn}{-1}%
```

Row processing:

```
6426 \LWR@maybenewtablerow%
```

Begin the opening table data tag:

```
6427 \LWR@htmltag{td colspan="#4" %
6428 \IfValueT{#2}{% rows?
6429 rowspan="#2" %
```



```

6430 \IfValueT{#1}{% vpos?
6431 \ifstrequal{#1}{b}{style="\LWR@origmbox{vertical-align:bottom}" }{}%
6432 \ifstrequal{#1}{t}{style="\LWR@origmbox{vertical-align:top}" }{}%
6433 }% vpos?
6434 }% rows?

```

```

6435 class="td%

```

Print the column type and vertical bars:

```

6436 \boolfalse{LWR@mcolvertbarl}%
6437 \boolfalse{LWR@mcolvertbarr}%
6438 \LWR@parsemulticolumnalignment{#5}{\LWR@printmccoltype}%

```

If this column has a cmidrule, add “rule” to the end of the HTML class tag.

If this position had a “Y” then add “rule” for a horizontal rule:

```

6439 \LWR@subaddcmidruletrim%
6440 {\LWR@gettexpparray{LWR@trimlrules}{\arabic{LWR@tablecolindex}}}%
6441 {\LWR@gettexpparray{LWR@trimrrules}{\arabic{LWR@lastmulticolumn}}}%

```

Also add vertical bar class.

```

6442 \ifbool{LWR@mcolvertbarl}{ tvertbarl}{}%
6443 \ifbool{LWR@mcolvertbarr}{ tvertbarr}{}%

```

Close the class tag’s opening quote:

```

6444 "%

```

```

6445 \LWR@tdstartstyles%

```

```

6446 \LWR@addtabulararrowcolor%

```

```

6447 \LWR@addcmidrulewidth%
6448 \LWR@addtabularhrulecolor%
6449 \LWR@addmulticolvertrulecolor%
6450 \LWR@addformatwppalignment{#5}%
6451 \LWR@tdendstyles%
6452 }% end of the opening table data tag
6453 \global\boolfalse{LWR@intabularmetadata}%
6454 \LWR@parsemulticolumnalignment{#5}{\LWR@printmccoldata}%
6455 }

```

### 64.18.3 Multicolumn

`\LWR@htmlmulticolumn`  $\{\langle numcols \rangle\} \{\langle alignment \rangle\} \{\langle text \rangle\}$

```
6456 \NewDocumentCommand{\LWR@htmlmulticolumn}{m m +m}%
6457 {%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
6458 \LWR@tabularhtmlcolumns{\arabic{\LWR@tablecolindex}}{\#1}
```

Create the multicolumn tag:

```
6459 \LWR@domulticolumn{\#1}{\arabic{\LWR@tabhtmlcoltotal}}{\#2}{\#3}%
```

Move to the next  $\LaTeX$  column:

```
6460 \addtocounter{\LWR@tablecolindex}{\#1}%
6461 \addtocounter{\LWR@tablecolindex}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
6462 \booltrue{\LWR@skipatbang}%
6463 }
```

### 64.18.4 Longtable captions

longtable captions use `\multicolumn`.

`Bool` `LWR@starredlongtable` Per the caption package, step the counter if longtable\*.

```
6464 \newbool{\LWR@starredlongtable}
6465 \boolfalse{\LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
6466 \providecommand*\LTcapttype{table}
```

`\LWR@longtabledatacaptiontag` \*  $[\langle toc entry \rangle] \{\langle caption \rangle\}$

```
6467 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
6468 {%
```

Remember the latest name for `\nameref`:

```

6469 \IfValueTF{#2}{% optional given?
6470   \ifblank{#2}% optional empty?
6471   {\LWR@setlatestname{#3}}% empty
6472   {\LWR@setlatestname{#2}}% given and non-empty
6473 }% optional given
6474 {\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:

```

6475 \LWR@tabularhtmlcolumns{1}{\arabic{LWR@tabletotalcols}}

```

Create the multicolumn tag:

```

6476 \LWR@domulticolumn{\arabic{LWR@tabletotalcols}}{\arabic{LWR@tabhtmlcoltotal}}{P}%
6477 {% \LWR@domulticolumn
6478 \IfBooleanTF{#1}% star?

```

Star version, show a caption but do not make a LOT entry:

```

6479 {% yes star
6480   \LWR@figcaption%
6481   #3%
6482   \endLWR@figcaption%
6483 }%
6484 {% No star:

```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```

6485   \ifbool{LWR@starredlongtable}%
6486   {%
6487     \ifblank{#2}% TOC entry
6488     }%
6489     {%
6490       \refstepcounter{\LTcapttype}%
6491       \protected@edef\@currentlabel{%
6492         \csuse{p@\LTcapttype}\csuse{the\LTcapttype}}%
6493       }%
6494     }{}%

```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```

6495   \LWR@figcaption%
6496   \csuse{fnum@\LTcapttype}\CaptionSeparator#3%
6497   \endLWR@figcaption%

```

See if an optional caption was given:

```
6498 \ifblank{#2}% TOC entry empty
```

if the optional caption was given, but empty, do not form a TOC entry

```
6499 {}%
```

If the optional caption was given, but might only be []:

```
6500 {% TOC entry not empty
6501 \IfNoValueTF{#2}% No TOC entry?
```

The optional caption is []:

```
6502 {% No TOC entry
6503 \addcontentsline%
6504 {\csuse{ext@LTcapttype}}%
6505 {\LTcapttype}%
6506 {%
6507 \protect\numberline%
6508 {\csuse{p@LTcapttype}\csuse{the\LTcapttype}}%
6509 {\ignorespaces #3\protect\relax}%
6510 }%
6511 }% end of No TOC entry
```

The optional caption has text enclosed:

```
6512 {% yes TOC entry
6513 \addcontentsline%
6514 {\csuse{ext@LTcapttype}}%
6515 {\LTcapttype}%
6516 {%
6517 \protect\numberline%
6518 {\csuse{p@LTcapttype}\csuse{the\LTcapttype}}%
6519 {\ignorespaces #2\protect\relax}%
6520 }%
6521 }% end of yes TOC entry
6522 }% end of TOC entry not empty
6523 }% end of no star
```

Skip any trailing @ or ! columns for this cell:

```
6524 \booltrue{LWR@skipatbang}%
6525 }% end of \LWR@domulticolumn
6526
6527 \addtocounter{LWR@tablecolindex}{\arabic{LWR@tabletotalcols}}
6528 \addtocounter{LWR@tablecolindex}{-1}
```

```
6529
6530 }
```

### 64.18.5 Counting HTML tabular columns

The  $\text{\TeX}$  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  $\text{\TeX}$  `\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.

```
6531 \newcounter{LWR@tabhtmlcolindex}
6532 \newcounter{LWR@tabhtmlcolend}
6533 \newcounter{LWR@tabhtmlcoltotal}
```

```
\LWR@subtabularhtmlcolumns {<index>}
```

Factored from `\LWR@tabularhtmlcolumns`, which follows.

```
6534 \newcommand*{\LWR@subtabularhtmlcolumns}[1]{%
```

Temporarily define a macro equal to the `@` specification for this column:

```
6535     \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{#1}}%
```

If the `@` specification is not empty, add to the count:

```
6536     \ifdefempty{\LWR@atbangspec}%
6537         {}%
6538         {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the `!` columns:

```
6539     \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{#1}}%
6540     \ifdefempty{\LWR@atbangspec}%
6541         {}%
6542         {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
6543 }
```

```
\LWR@tabularhtmlcolumns {<starting  $\text{\TeX}$  column>} {<number  $\text{\TeX}$  columns>}
```

Compute the total number of HTML columns being spanned, considering the starting  $\LaTeX$  table column and the number of  $\LaTeX$  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.

```
6544 \newcommand*{\LWR@tabularhtmlcolumns}[2]{%
```

Count the starting index, compute ending index, and begin with the count being the  $\LaTeX$  span, to which additional @ and ! columns may be added:

```
6545 \setcounter{LWR@tabhtmlcolindex}{#1}%
6546 \setcounter{LWR@tabhtmlcoltotal}{#2}%
6547 \setcounter{LWR@tabhtmlcolend}{#1}%
6548 \addtocounter{LWR@tabhtmlcolend}{#2}%
```

If at the left edge, add the at/bang columns for the left edge:

```
6549 \ifnumcomp{\value{LWR@tabhtmlcolindex}}{=}{1}{%
6550     \LWR@subtabularhtmlcolumns{leftedge}%
6551 }{ }%
```

Walk across the  $\LaTeX$  columns looking for @ and ! columns:

```
6552 \whileboolexpr{%
6553     test {%
6554         \ifnumcomp{\value{LWR@tabhtmlcolindex}}{<}{\value{LWR@tabhtmlcolend}}}%
6555     }%
6556 }%
6557 {%
6558     \LWR@subtabularhtmlcolumns{\arabic{LWR@tabhtmlcolindex}}%
6559     \addtocounter{LWR@tabhtmlcolindex}{1}%
6560 }% \whiledo
6561 }

6562 \end{warpHTML}
```

## 64.19 Multicolumnrow

A print-mode version is defined here, and is also used during HTML output while inside a lateximage.

See section 232 for the HTML versions.

**for HTML & PRINT:** 6563 \begin{warpall}

```
\multicolumnrow {<1:cols>} {<2:halign>} [<3:vpos>] {<4:numrows>} [<5:bigstruts>] {<6:width>} [<7:fixup>]
{<8:text>}
```

For discussion of the use of `\DeclareExpandableDocumentCommand`, see:  
[https://tex.stackexchange.com/questions/168434/](https://tex.stackexchange.com/questions/168434/problem-with-abbreviation-of-multirow-and-multicolumn-latex)  
[problem-with-abbreviation-of-multirow-and-multicolumn-latex](https://tex.stackexchange.com/questions/168434/problem-with-abbreviation-of-multirow-and-multicolumn-latex)

After the user may have

```
6564 \AtBeginDocument{
```

`\@ifundefined{@xmultirow}` determines if multirow was never loaded.

```
6565 \@ifundefined{@xmultirow}
6566 {}% no version of multirow was loaded
6567 {% \@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.

```
6568 \@ifpackageloaded{multirow}{% v2.0 or newer
6569 \@ifpackagelater{multirow}{2016/09/01}% 2016/09/27 for v2.0
6570 {% v2.0+:
6571 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
6572   {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
6573 {\multicolumn{#1}{#2}{\@xmultirow[#3]{#4}[#5]{#6}[#7]{#8}}}%
6574 }
6575 {% loaded but older, probably not executed:
6576 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
6577   {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
6578 {\multicolumn{#1}{#2}{\@xmultirow{#4}[#5]{#6}[#7]{#8}}}%
6579 }
6580 }% packageloaded{multirow}
```

If not `\@ifpackageloaded{multirow}` but `\@xmultirow` is defined, then this must be v1.6 or earlier, which did not `\ProvidesPackage{multirow}`, and did not have the `vposn` option.

```
6581 {% v1.6 or older did not \ProvidePackage
6582 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
6583   {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
6584 {\multicolumn{#1}{#2}{\@xmultirow{#4}[#5]{#6}[#7]{#8}}}%
6585 }
```

The user-level interface. This is provided if the HTML version was not already given.

```
6586 \providecommand*{\multicolumnrow}{\LWR@origmulticolumnrow}
```

```

6587 }% \@xmultirow defined, so multirow was loaded
6588 }% AtBeginDocument

6589 \end{warpall}

```

## 64.20 Utility macros inside a table

**for HTML output:** 6590 \begin{warpHTML}

Used to prevent opening a tabular data cell if the following token is one which does not create tabular data:

```
6591 \newcommand*{\LWR@donothing}{}

```

In case bigdelim is not loaded:

```

6592 \newcommand*{\ldelim}{}
6593 \newcommand*{\rdelim}{}

```

```
6594 \end{warpHTML}

```

## 64.21 Special-case tabular markers

**for HTML & PRINT:** 6595 \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 8.9.


```
6596 \newcommand*{\TabularMacro}{}

```

```
6597 \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:** 6598 \begin{warpHTML}

```

6599 \newcommand*{\ResumeTabular}{%
6600 \global\boolfalse{LWR@exitingtabular}%
6601 \global\boolfalse{LWR@tabularmutemods}%
6602 \LWR@getmynexttoken%
6603 }

```

```

6604 \end{warpHTML}

```

**for PRINT output:** 6605 \begin{warpprint}

```

6606 \newcommand*{\ResumeTabular}{}
6607 \end{warpprint}

```

## 64.22 Checking for a new table cell

**for HTML output:** 6608 \begin{warpHTML}

Bool LWR@exitingtabular When \end is found, turns off the next opening data tag.

```

6609 \newbool{LWR@exitingtabular}

```

Bool LWR@tabularmutemods Mutes HTML output for @, !, < and >.

This is used while printing the final row to generate \bottomrules.

```

6610 \newbool{LWR@tabularmutemods}

```

`\LWR@tabledatacolumnntag` Open a new HTML table cell unless the next token is for a macro which does not create data, such as `\hline`, `\toprule`, etc:

```
6611 \newcommand*{\LWR@tabledatacolumnntag}%
6612 {%
6613 \LWR@traceinfo{\LWR@tabledatacolumnntag}%
```

`\show\LWR@mynexttoken` to see what tokens to look for

If not any of the below, start a new table cell:

```
6614 \let\mynext\LWR@tabledatasinglecolumnntag%
```

If exiting the tabular:

```
6615 \ifdefequal{\LWR@mynexttoken}{\end}%
6616     {\global\booltrue{\LWR@exitingtabular}}{}%
```

`longtable` can have a caption in a cell

```
6617 \ifdefequal{\LWR@mynexttoken}{\caption}%
6618     {\let\mynext\LWR@donothing}{}%
```

Look for other things which would not start a table cell:

```
6619 \ifdefequal{\LWR@mynexttoken}{\multicolumn}%
6620     {\let\mynext\LWR@donothing}{}%
6621 \ifdefequal{\LWR@mynexttoken}{\multirow}%
6622     {\let\mynext\LWR@donothing}{}%
6623 \ifdefequal{\LWR@mynexttoken}{\multicolumnrow}%
6624     {\let\mynext\LWR@donothing}{}%
6625 \ifdefequal{\LWR@mynexttoken}{\noalign}%
6626     {\let\mynext\LWR@donothing}{}%
```

If an `\mrowcell`, this is a cell to be skipped over:

```
6627 \ifdefequal{\LWR@mynexttoken}{\mrowcell}%
6628     {\let\mynext\LWR@donothing}{}%
```

If an `\mcolrowcell`, this is a cell to be skipped over:

```
6629 \ifdefequal{\LWR@mynexttoken}{\mcolrowcell}%
6630     {\let\mynext\LWR@donothing}{}%
6631 %
6632 \ifdefequal{\LWR@mynexttoken}{\TabularMacro}%
6633     {\let\mynext\LWR@donothing}{}%
```

```
6634 %
6635 \ifdefequal{\LWR@mynexttoken}{\hline}%
6636     {\let\mynext\LWR@donothing}{}%
6637 %
6638 \ifdefequal{\LWR@mynexttoken}{\firsthline}%
6639     {\let\mynext\LWR@donothing}{}%
6640 %
6641 \ifdefequal{\LWR@mynexttoken}{\lasthline}%
6642     {\let\mynext\LWR@donothing}{}%
6643 %
6644 \ifdefequal{\LWR@mynexttoken}{\toprule}%
6645     {\let\mynext\LWR@donothing}{}%
6646 %
6647 \ifdefequal{\LWR@mynexttoken}{\midrule}%
6648     {\let\mynext\LWR@donothing}{}%
6649 %
6650 \ifdefequal{\LWR@mynexttoken}{\cmidrule}%
6651     {\let\mynext\LWR@donothing}{}%
6652 %
6653 \ifdefequal{\LWR@mynexttoken}{\specialrule}%
6654     {\let\mynext\LWR@donothing}{}%
6655 %
6656 \ifdefequal{\LWR@mynexttoken}{\cline}%
6657     {\let\mynext\LWR@donothing}{}%
6658 %
6659 \ifdefequal{\LWR@mynexttoken}{\bottomrule}%
6660     {\let\mynext\LWR@donothing}{}%
6661 %
6662 \ifdefequal{\LWR@mynexttoken}{\rowcolor}%
6663     {\let\mynext\LWR@donothing}{}%
6664 %
6665 \ifdefequal{\LWR@mynexttoken}{\arrayrulecolor}%
6666     {\let\mynext\LWR@donothing}{}%
6667 %
6668 \ifdefequal{\LWR@mynexttoken}{\doublerulesepcolor}%
6669     {\let\mynext\LWR@donothing}{}%
6670 %
6671 \ifdefequal{\LWR@mynexttoken}{\warpprintonly}%
6672     {\let\mynext\LWR@donothing}{}%
6673 %
6674 \ifdefequal{\LWR@mynexttoken}{\warpHTMLonly}%
6675     {\let\mynext\LWR@donothing}{}%
6676 %
6677 \ifdefequal{\LWR@mynexttoken}{\ldelim}%
6678     {\let\mynext\LWR@donothing}{}%
6679 %
6680 \ifdefequal{\LWR@mynexttoken}{\rdelim}%
6681     {\let\mynext\LWR@donothing}{}%
```

Ignore an empty line between rows:

```
6682 \ifdefequal{\LWR@mynexttoken}{\par}%
6683     {\let\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:

```
6684 \LWR@traceinfo{\LWR@tabledatacolumnstag: about to do mynext}%
6685 \mynext%
6686 \LWR@traceinfo{\LWR@tabledatacolumnstag: done}%
6687 }
```

```
6688 \end{warpHTML}
```

### 64.23 `\mrowcell`

**for HTML & PRINT:** 6689 `\begin{warpall}`


`\mrowcell` The user must insert `\mrowcell` into any `\multirow` cells which must be skipped.  This command has no action during print output.

```
6690 \newcommand*{\mrowcell}{}%
```

```
6691 \end{warpall}
```

### 64.24 `\mcolrowcell`

**for HTML & PRINT:** 6692 `\begin{warpall}`

`\mcolrowcell` The user must insert `\mcolrowcell` into any `\multicolumnrow` cells which must be skipped.  This command has no action during print output.

```
6693 \newcommand*{\mcolrowcell}{}%
```

```
6694 \end{warpall}
```

## 64.25 New tabular environment

for HTML output: 6695 \begin{warpHTML}

These are default definitions in case booktabs is not loaded, and are not expected to used, but must exist as placeholders.

```
6696 \newcommand*{\LWR@origtoprule}[1] [] {\hline}
6697 \newcommand*{\LWR@origmidrule}[1] [] {\hline}
6698 \LetLtxMacro\LWR@origcmidrule\cline
6699 \newcommand*{\LWR@origbottomrule}[1] [] {\hline}
6700 \newcommand*{\LWR@origaddlinespace}[1] [] {}
6701 \newcommand*{\LWR@origmorecmidrules}{}
6702 \newcommand*{\LWR@origspecialrule}[3] {\hline}
```

\noalign {<text>} Redefined for use inside tabular.

```
6703 \LetLtxMacro\LWR@orignoalign\noalign
6704
6705 \newcommand{\LWR@tabularnoalign}[1] {%
6706 \begingroup%
6707 \global\advance\rownum\m@ne%
6708 \renewcommand*{\LWR@xcolorrowHTMLcolor}{}%
6709 \multicolumn{\value{\LWR@tabletotalcols}}{1}{#1} \\
6710 \endgroup%
6711 % \@rowcolors%
6712 \LWR@getmynexttoken%
6713 }
```

\LWR@HTMLhline The definition of \hline depends on whether tabs has been loaded. If so, optional space below the line may be specified, but will be ignored.

```
6714 \AtBeginDocument{
6715 \ifpackageloaded{lwarp-tabs}
6716 {
6717 \newcommand*{\LWR@HTMLhline}[1] [] {%
6718 \ifbool{FormatWP}%
6719 {\LWR@docmidrule{1-\arabic{\LWR@tabletotalcols}}}%
6720 {\booltrue{\LWR@doinghline}}%
6721 \LWR@getmynexttoken%
6722 }
6723 {
6724 \newcommand*{\LWR@HTMLhline} {%
6725 \ifbool{FormatWP}%
6726 {\LWR@docmidrule{1-\arabic{\LWR@tabletotalcols}}}%
6727 {\booltrue{\LWR@doinghline}}%
6728 \LWR@getmynexttoken%
6729 }
```

```
6729 }
6730 }% AtBeginDocument
```

`\LWR@HTMLcline`  $\{\langle columns \rangle\}$

```
6731 \NewDocumentCommand{\LWR@HTMLcline}{m}%
6732 {\LWR@docmidrule{#1}\LWR@getmynexttoken}%
```

`\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.

```
6733 \AtBeginDocument{
6734 \ifundefined{frenchbsetup}%
6735 {% no babel-french
6736   \newcommand*\LWR@nullifyNoAutoSpacing{}
6737 }% no babel-french
6738 {% yes babel-french
6739   \newcommand*\LWR@nullifyNoAutoSpacing{%
6740     \NoAutoSpacing%
6741     \renewcommand*\NoAutoSpacing{}%
6742     \renewcommand*\LWR@FBcancel{}%
6743   }
6744 }% yes babel-french
6745 }% AtBeginDocument
```

Env `LWR@tabular`  $[\langle vertposition \rangle] \{\langle colspecs \rangle\}$

The new tabular environment will be `\let` in `\LWR@LwarpStart`, since `siunitx` might redefine `tabular` in the user's document.

```
6746 \StartDefiningTabulars
6747
6748 \newenvironment*LWR@tabular}[2] []
6749 {%
6750 \LWR@traceinfo{LWR@tabular started}%
6751 \addtocounter{LWR@tabulardepth}{1}%
```

Not yet started a table row:

```
6752 \global\boolfalse{LWR@startedrow}%
```

Not yet doing any rules:

```
6753 \global\boolfalse{LWR@doinghline}%
6754 \global\boolfalse{LWR@doingtbrule}%
6755 \global\boolfalse{LWR@doingcmidrule}%
```

For babel-french, turn off auto spacing one time, then nullify the autospacing commands since were not compatible with the tabular parsing code.

```
6756 \LWR@nullifyNoAutoSpacing%
```

Have not yet found the end of tabular command. Unmute the @ and ! columns.

```
6757 \global\boolfalse{LWR@exitingtabular}%
6758 \global\boolfalse{LWR@tabularmutemods}%
```

Create the table tag:

```
6759 \global\booltrue{LWR@intabularmetadata}%
6760 \LWR@forcenewpage
6761 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
6762 \LWR@parsetablecols{#2}%
```

Table col spec is: \LWR@tablecolspec which is a string of llccrr, etc.

Do not place the table inside a paragraph:

```
6763 \LWR@stoppars%
```

Track column #:

```
6764 \setcounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
6765 \boolfalse{LWR@tabularcelladded}%
```

Start looking for midrules:

```
6766 \LWR@clearmidrules%
```

\\ becomes a macro to end the table row:

```
6767 \LetLtxMacro{\\}{\LWR@tabularendoffline}%
```

The following adjust for colortbl:

```
6768 \LetLtxMacro\columncolor\LWR@HTMLcolumncolor%
6769 \LetLtxMacro\rowcolor\LWR@HTMLrowcolor%
6770 \LetLtxMacro\cellcolor\LWR@HTMLcellcolor%
```

```

6771 \LetLtxMacro\arrayrulecolor\LWR@HTMLarrayrulecolor%
6772 \LetLtxMacro\doublerulesepcolor\LWR@HTMLdoublerulesepcolor%
6773 \renewcommand*{\LWR@columnHTMLcolor}{}%
6774 \renewcommand*{\LWR@rowHTMLcolor}{}%
6775 \renewcommand*{\LWR@cellHTMLcolor}{}%
6776 \@rowcolors%

```

The vertical rules are set to the color active at the start of the tabular. `\arrayrulecolor` will then affect horizontal rules inside the tabular, but not the vertical rules.

```

6777 \edef\LWR@vertruleHTMLcolor{\LWR@ruleHTMLcolor}%

```

Tracking the depth of cell color <div>s:

```

6778 \setcounter{LWR@cellcolordepth}{0}%

```

The following may appear before a data cell is created, so after doing their actions, we look ahead with `\LWR@getmynexttoken` to see if the next token might create a new data cell:

The optional parameter for `\hline` supports the `tabs` package.

```

6779 \LWR@traceinfo{LWR@tabular: redefining macros}%
6780 \LetLtxMacro\noalign\LWR@tabularnoalign%
6781 \LetLtxMacro\hline\LWR@HTMLhline%
6782 \LetLtxMacro\cline\LWR@HTMLcline%

6783 \DeclareDocumentCommand{\toprule}{o d()}%
6784   {%
6785     \IfValueTF{##1}%
6786       {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
6787       {%
6788         \ifbool{FormatWP}%
6789         {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
6790         {\booltrue{LWR@doingtbrule}}}%
6791       }%
6792   \LWR@getmynexttoken}%
6793 %
6794 \DeclareDocumentCommand{\midrule}{o d()}%
6795   {%
6796     \IfValueTF{##1}%
6797       {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
6798       {%
6799         \ifbool{FormatWP}%
6800         {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
6801         {\booltrue{LWR@doinghline}}}%
6802       }%
6803   \LWR@getmynexttoken}%

```



```

6804 %
6805 \DeclareDocumentCommand{\cmidrule}{0{\LWR@cmidrulewidth} d() m}%
6806 {\LWR@docmidrule[##1](##2){##3}\LWR@getmynexttoken}%
6807 %
6808 \DeclareDocumentCommand{\bottomrule}{o d()}%
6809 {%
6810     \IfValueTF{##1}%
6811     {\LWR@docmidrule[##1](){1-\arabic{\LWR@tabletotalcols}}}%
6812     {%
6813         \ifbool{FormatWP}%
6814         {\LWR@docmidrule[##1](){1-\arabic{\LWR@tabletotalcols}}}%
6815         {\booltrue{\LWR@doingtbrule}}}%
6816     }%
6817     \LWR@getmynexttoken}%
6818 %
6819 \DeclareDocumentCommand{\addlinespace}{o}{}%
6820 \DeclareDocumentCommand{\morecmidrules}{l}{}%
6821 \DeclareDocumentCommand{\specialrule}{m m m d()}%
6822 {\LWR@docmidrule[##1](){1-\arabic{\LWR@tabletotalcols}}\LWR@getmynexttoken}%

```

The following create data cells and will have no more data in this cell, so we do not want to look ahead for a possible data cell, so do not want to use \LWR@getmynexttoken.

```

6823 \renewcommand{\multicolumn}{\LWR@htmlmulticolumn}%
6824 \renewcommand*{\mrowcell}{%
6825     \LWR@maybenewtablerow%
6826     \LWR@tabularleftedge%
6827     \global\booltrue{\LWR@skippingmrowcell}%
6828 }%
6829 \renewcommand*{\mcolrowcell}{%
6830     \LWR@maybenewtablerow%
6831     \global\booltrue{\LWR@skippingmcolrowcell}%
6832 }%
6833 \LetLtxMacro\caption\LWR@longtabledatacaptiontag%

```

Reset for new processing:

```

6834 \global\boolfalse{\LWR@tableparcell}%
6835 \global\boolfalse{\LWR@skippingmrowcell}%
6836 \global\boolfalse{\LWR@skippingmcolrowcell}%
6837 \global\boolfalse{\LWR@skipatbang}%
6838 \global\boolfalse{\LWR@emptyatbang}%

```

Set & for its special meaning inside the tabular:

```

6839 \StartDefiningTabulars%
6840 \protected\gdef&{\LWR@tabularampersand}%

```

Nest one level deeper of tabular paragraph handling:

```
6841 \addtocounter{LWR@tabularpardepth}{1}%
```

Look ahead for a possible table data cell:

```
6842 \LWR@traceinfo{LWR@tabular: about to LWR@getmynexttoken}%
6843 \LWR@getmynexttoken%
6844 }%
```

Ending the environment:

```
6845 {%
6846 \LWR@traceinfo{LWR@tabular ending}%
```

Unnest one level of tabular paragraph handling:

```
6847 \addtocounter{LWR@tabularpardepth}{-1}%
6848 \ifboolexpr{%
6849   test {%
6850     \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}
6851   } or %
6852   (%
6853     bool{LWR@intabularmetadata} and%
6854     not bool{LWR@tabularcelladded} and%
6855     test {%
6856       \ifnumcomp{\value{LWR@tablecolindex}}{=}{\value{LWR@tabletotalcols}}%
6857     }%
6858   )%
6859 }%
6860 {%
6861   \LWR@tabularfinishrow%
6862 }%
6863 {%
6864   \LWR@closetabledatacell%
6865 }%
6866 \LWR@htmlblocktag{/tr}%
```

xcolor row color support:

```
6867 \@rowcolor%

6868 \LWR@htmlblocktag{/table}%
6869 \global\boolfalse{LWR@intabularmetadata}%
```

Unnest one level of tabular:

```
6870 \addtocounter{LWR@tabulardepth}{-1}%
```

Restore & to its usual meaning:

```
6871 \protected\gdef&{\LWR@origampmacro}%
6872 \EndDefiningTabulars%
6873 \LWR@traceinfo{\LWR@tabular finished ending}%
6874 }
6875
6876 \EndDefiningTabulars

6877 \end{warpHTML}
```

## 65 Cross-references

Sectioning commands have been emulated from scratch, so the cross-referencing commands are custom-written for them. Emulating both avoids several layers of patches.

The zref package is used to remember section name, file, and lateximage depth and number for each label.

Table 10 shows the data structures related to cross-referencing.

for HTML output: 6878 \begin{warpHTML}

### 65.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

```
6879 \providecommand*{\@currentlabelname}{}%
```

`\LWR@stripperperiod` `{\langle text \rangle} [(\.)]`

Removes a trailing period.

```
6880 \def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%
```

`\LWR@setlatestname` `{\langle object name \rangle}`

Removes `\label`, strips any final period, and remembers the result.

```
6881 \newcommand*{\LWR@setlatestname}[1]{%
```

Table 10: Cross-referencing data structures

<b>Original <math>\LaTeX</math>:</b>	(print and HTML)
<code>\refstepcounter:</code> Steps the counter and sets <code>\@currentlabel</code> . <code>\@currentlabel:</code> <code>\p@&lt;ctr&gt;\the&lt;ctr&gt;</code> Updated by <code>\refstepcounter</code> . <code>\label:</code> Writes to the .aux file: <code>\newlabel{&lt;label&gt;}{\@currentlabel}{\thepage}}</code> <code>\newlabel:</code> When the .aux file is read, sets <code>\r@&lt;label&gt;</code> . <code>\r@&lt;label&gt;:</code> Set to: <code>{\@currentlabel}{\thepage}}</code> <code>\ref:</code> Returns the first part of <code>\r@&lt;label&gt;</code> . <code>\pageref:</code> Returns the second part of <code>\r@&lt;label&gt;</code> .	
<b>Added by lwarp:</b>	(HTML only)
<code>\label:</code> Adds HTML tags (section 65.3), plus <code>\splabel</code> data (section 65.2): <code>zLWR@name:</code> The section name for this label. <code>zLWR@htmlfilenumber:</code> The file number or name for this label. <code>zLWR@lateximagedepth:</code> The <code>lateximagedepth</code> for this label. <code>zLWR@lateximagenumber:</code> The <code>lateximagenumber</code> for this label. <code>\nameref:</code> Emulated from <code>hyperref</code> for <code>lwarp</code> . See section 65.4. <code>\ref</code> and <code>\nameref:</code> Adds HTML tags. See section 65.4.	
<b>Added by amsmath:</b>	(print and HTML)
<code>\label:</code> Execution is delayed until the math environment is completed. <code>\ltx@label:</code> $\LaTeX$ <code>\label</code> , (HTML: patched by <code>lwarp</code> ), later patched by <code>cleveref</code> .	
<b>Added by cleveref:</b>	(print and HTML)
<code>\refstepcounter:</code> Added: sets <code>\cref@currentlabel</code> . <code>\cref@currentlabel:</code> ( <code>&lt;type&gt;=&lt;ctr&gt;</code> unless an alias is used): <code>[&lt;type&gt;][\arabic{&lt;ctr&gt;}][&lt;parent&gt; &lt;trs&gt;]{\p@&lt;ctr&gt;\the&lt;ctr&gt;}</code> Also see section 51.4 for use with footnotes. <code>\label:</code> Writes to the .aux file: <code>\newlabel{&lt;label&gt;@cref}{\cref@currentlabel}{\thepage}}</code> <code>\newlabel:</code> (Unchanged.) When the .aux file is read, sets <code>\r@&lt;label&gt;@cref</code> . <code>\r@&lt;label&gt;@cref:</code> Set to: <code>{\cref@currentlabel}{\thepage}}</code> <b>Utility functions:</b> See <code>\cref@getlabel</code> , <code>\cref@gettype</code> , <code>\cref@getcounter</code> , <code>\cref@getprefix</code> . <b>Cross-referencing names:</b> <code>\crefname</code> and <code>\Crefname</code> assign human-readable names for references to this counter type.	
<b>Additionally patched by lwarp:</b>	(HTML only)
<code>\cref, etc.:</code> Modified for <code>lwarp</code> . See section 78. <code>\label</code> inside math: See section 71.5.1.	
<b>Footnotes:</b> See <code>\noteentry</code> in section 51.4.	

Remove `\label` and other commands from the name, the strip any final period. See `zref-titleref` and `getttitlestring`.

```
6882 \GetTitleStringExpand{#1}%
6883 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
6884 \edef\@currentlabelname{%
6885 \expandafter\LWR@stripperperiod\@currentlabelname%
6886 \ltx@empty.\ltx@empty\@nil%
6887 }%
6888 }
```

## 65.2 Zref setup

See:

[http://tex.stackexchange.com/questions/57194/  
extract-section-number-from-equation-reference](http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference)

Create a new property list called `special`:

```
6889 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
6890 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
6891 \zref@newprop{zLWR@htmlfilenumber}{%
6892 \ifbool{FileSectionNames}{\LWR@thisfilename}{\arabic{LWR@htmlfilenumber}}%
6893 }%
```

Additional properties for `lateximages`:

```
6894 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
6895 \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`:

```
6896 \zref@addprop{special}{zLWR@name}
6897 \zref@addprop{special}{zLWR@htmlfilenumber}
6898 \zref@addprop{special}{zLWR@lateximagedepth}
6899 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
6900 \newcommand*{\LWR@spre{#1}{#2}{??}}{#2}
6901 \zref@extractdefault{#1}{#2}{??}%
6902 }
```

`\LWR@nameref` `{\label}` Returns the section name for this label:

```
6903 \newcommand*{\LWR@nameref}[1]{%
6904 \LWR@spre{#1}{\LWR@name}%
6905 }
```

`\LWR@htmlfileref` `{\label}` Returns the file number or name for this label:

```
6906 \newcommand*{\LWR@htmlfileref}[1]{%
```

DO NOT USE `\LWR@traceinfo` HERE! Will be expanded.

```
6907 \LWR@spre{#1}{\LWR@htmlfilenumber}%
6908 }
```

`\LWR@lateximagedepthref` `{\label}` Returns the `lateximagedepth` for this label:

```
6909 \newcommand*{\LWR@lateximagedepthref}[1]{%
6910 \LWR@spre{#1}{\LWR@lateximagedepth}%
6911 }
```

`\LWR@lateximagenumberref` `{\label}` Returns the `lateximagenumber` for this label:

```
6912 \newcommand*{\LWR@lateximagenumberref}[1]{%
6913 \LWR@spre{#1}{\LWR@lateximagenumber}%
6914 }
```

`\LWR@splabel` `{\label}` Sanitize the name and then creates the label:

```
6915 \newcommand*{\LWR@splabel}[1]{%
6916 \LWR@traceinfo{\LWR@splabel !#1!}%
6917 \LWR@setlatestname{\@currentlabelname}%
6918 \zref@labelbylist{#1}{special}%
6919 }
```

## 65.3 Labels

`\LWR@sublabel` `{\label}` Creates an HTML id tag.

`\detokenize` is used to allow underscores in the labels.

```
6920 \newcommand*{\LWR@sublabel}[1]{%
6921 \LWR@traceinfo{\LWR@sublabel !#1!}%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```
6922 \ifnumcomp{\value{\LWR@lateximagedepth}}{>}{0}%
6923 }{%
6924 {% not lateximage
```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```
6925     \LWR@sanitize{#1}%
6926     \ifbool{\LWR@doingstartpars}%
6927     {% pars allowed
6928         \ifbool{\LWR@doingapar}%
6929         {% par started
6930             \LWR@htmltag{a \LWR@origmbox{id="\LWR@sanitized"}}\LWR@htmltag{/a}%
6931         }% par started
6932         {% par not started
6933             \LWR@stoppars%
6934             \LWR@htmltag{a \LWR@origmbox{id="\LWR@sanitized"}}\LWR@htmltag{/a}%
6935             \LWR@startpars%
6936         }% par not started
6937     }% pars allowed
6938     {% pars not allowed
6939         \LWR@htmltag{a \LWR@origmbox{id="\LWR@sanitized"}}\LWR@htmltag{/a}%
6940     }% pars not allowed
6941 }% not lateximage
6942 }
```

`\LWR@newlabel` (*<bookmark>*) [*<label>*] [*<type>*]

`\label` during HTML output when not in SVG math mode, removing extra spaces around the label, as done by regular  $\LaTeX$  `\label`.

`cleveref` later encases this to add its own cross-referencing.

The optional *<bookmark>* is per the memoir class, and is ignored.

The optional *<type>* is per the ntheorem package, and is ignored.

```
6943 \NewDocumentCommand{\LWR@newlabel}{d() m o}{%
6944 \LWR@traceinfo{\LWR@newlabel: starting}%
6945 \LWR@traceinfo{\LWR@newlabel: !#2!}%
6946 % \@bsphack%
```

Create a traditional  $\LaTeX$  label, as modified by cleveref:

```
6947 \LWR@origlabel{#2}%
```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```
6948 \LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
6949 \LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
6950 \LWR@traceinfo{LWR@newlabel: LWR@htmlfilenumber is \arabic{LWR@htmlfilenumber}}%
6951 \LWR@splabel{#2}%
6952 \LWR@sublabel{#2}%
6953 % \@esphack%
6954 \LWR@traceinfo{LWR@newlabel: done}%
6955 }
```

## 65.4 References

`\LWR@startref` `{\label}` (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, # character, and a label.

```
6956 \newcommand*{\LWR@startref}[1]
6957 {%
6958 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
6959 \LWR@sanitize{#1}%
6960 \LWR@traceinfo{LWR@startref A: !#1!}%
```

Create the filename part of the link:

```
6961 \LWR@htmltag{a href="%
6962 \LWR@traceinfo{LWR@startref B}%
6963 \LWR@origmbox{\LWR@htmlrefsectionfilename{#1}}%
6964 \LWR@traceinfo{LWR@startref C}%
6965 \#%
```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```
6966 \LWR@traceinfo{LWR@startref D: !#1!}%
6967 \ifthenelse{\equal{\LWR@lidref}{??}}%
```

“??” if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:



```

6968 {%
6969   \LWR@traceinfo{LWR@startref D0: ??}%
6970   ??%
6971 }%

```

If LWR@lateximagedepth is known. Use a lateximage if the depth is greater than zero, or a regular link otherwise:

```

6972 {%
6973   \LWR@traceinfo{LWR@startref D1: \LWR@lidref}%
6974   \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}%
6975   {%
6976     \LWR@traceinfo{LWR@startref D2: \LWR@lidref}%
6977     lateximage\LWR@lateximagenumberref{#1}%
6978   }%
6979   {%
6980     \LWR@traceinfo{LWR@startref D3}%

```

\detokenize is used to allow underscores in the labels:

```

6981   \LWR@origmbox{\LWR@sanitized}%
6982   }%
6983 }%
6984 \LWR@traceinfo{LWR@startref E}%

```

Closing quote:

```

6985 "%
6986 \LWR@traceinfo{LWR@startref F}%
6987 }

```

`\LWR@subnewref`  $\{ \langle label \rangle \} \{ \langle label \text{ or } sub@label \rangle \}$

Factored for the subfig package. Uses the original label for the hyper-reference, but prints its own text, such as “1(b)”.

```

6988 \NewDocumentCommand{\LWR@subnewref}{m m}{%
6989 \LWR@traceinfo{LWR@subnewref #1 #2}%
6990 \LWR@startref{#1}%
6991 \LWR@origref{#2}%
6992 \LWR@htmltag{/a}%
6993 }

```

`\ref` \*  $\{ \langle label \rangle \}$  \ref is \let to \LWR@newref

`\LWR@newref` \*  $\langle label \rangle$  Create an internal document reference link, or without a link if starred per hyperref.

```
6994 \NewDocumentCommand{\LWR@newref}{s m}{%
6995 \LWR@traceinfo{\LWR@newref !#2!}%
6996 \IfBooleanTF{#1}%
6997 {\LWR@origref{#2}}%
6998 {\LWR@subnewref{#2}{#2}}%
6999 }
```

`\pagerefPageFor` Text for page references.

```
7000 \newcommand*{\pagerefPageFor}{see }
```

`\pageref` \*  $\langle label \rangle$  Create an internal document reference, or just the unlinked number if starred, per hyperref.

```
7001 \NewDocumentCommand{\LWR@newpageref}{s m}{%
7002 \IfBooleanTF{#1}%
7003 {(\pagerefPageFor\LWR@origref{#2})}%
7004 {(\cpageref{#2})}%
7005 }
```

`\nameref`  $\langle label \rangle$

```
7006 \newrobustcmd*{\nameref}[1]{%
7007 \LWR@traceinfo{nameref}%
7008 \LWR@startref{#1}%
7009 \LWR@traceinfo{nameref B}%
7010 \LWR@nameref{#1}%
7011 \LWR@traceinfo{nameref C}%
7012 \LWR@htmltag{/a}%
7013 \LWR@traceinfo{nameref: done}%
7014 }
```

`\Nameref`  $\langle label \rangle$  In print, adds the page number. In HTML, does not.

```
7015 \LetLtxMacro\Nameref\nameref
```

## 65.5 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.

```
7016 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
7017 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Emulates `hyperref`:

`\@currentHref` Added to support backref.

```
7018 \AtBeginDocument{
7019 \def\@currentHref{%
7020 autopage-\theLWR@currentautosec%
7021 }
7022 }
```

Create a link with a text name:

`\LWR@subhyperref` `{\langle URL \rangle}{\langle text \rangle}`

```
7023 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
7024 \LWR@traceinfo{\LWR@subhyperref !#1!}%
7025 \LWR@sanitize{#1}%
7026 \LWR@htmltag{%
7027   a href="\LWR@sanitized" %
7028 %   a href="%
7029 %   \begingroup\@sanitize#1\endgroup%
7030 %   " %
7031   target="\_{}blank"\LWR@orignewline%
7032 }%
7033 #2%
7034 \LWR@htmltag{/a}%
7035 \LWR@ensuredoingapar%
7036 }
```

`\LWR@subhyperrefclass` `{\langle URL \rangle}{\langle text \rangle}{\langle htmlclass \rangle}`

```
7037 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
```

```

7038 % \LWR@sanitize{#1}%
7039 \LWR@htmltag{%
7040 %     a href="\LWR@sanitized"
7041 %     a href="%
7042 %         \begingroup\@sanitize#1\endgroup%
7043 %     " %
7044 %     class="#3"\LWR@orignewline%
7045 }%
7046 #2%
7047 \LWR@htmltag{/a}%
7048 \LWR@ensuredoingapar%
7049 }

```

`\href` [*<options>*] {<URL>} {<text>}

Create a link with accompanying text:

```

7050 \DeclareDocumentCommand{\href}{0+{ m +m}{%
7051 \LWR@ensuredoingapar%
7052 \LWR@subhyperref{#2}{#3}%
7053 }

```

`\nolinkurl` {<URL>}

Print the name of the link without creating the link:

```

7054 \newcommand*{\nolinkurl}[1]{%
7055 \LWR@ensuredoingapar%
7056 \def\LWR@templink{#1}%
7057 \@onelevel@sanitize\LWR@templink%
7058 \LWR@templink%
7059 }

```

`\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`.

```

7060 \DeclareDocumentCommand{\url}{m}{%
7061 \LWR@ensuredoingapar%
7062 \def\LWR@templink{#1}%
7063 \@onelevel@sanitize\LWR@templink%
7064 \href{\LWR@templink}{\LWR@templink}%
7065 }

```

`\LWR@subinlineimage` [*<alttag>*] {<class>} {<filename>} {<extension>} {<style>}

```
7066 \newcommand*{\LWR@subinlineimage}[5][ ]{%  
7067 \ifblank{#1}%  
7068 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"}}%  
7069 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"}}%  
7070 }  
  
7071 \end{warpHTML}
```

Table 11: Float data structures

---

For each `<type>` of float (figure, table, etc.) there exists the following:

---

- counter `<type>`:** A counter called `<type>`, such as figure, table.
  - `\<type>name`:** Name. `\figurename` prints “Figure”, etc.
  - `\ext@<type>`:** File extension. `\ext@figure` prints “lof”, etc.
  - `\fps@<type>`:** Placement.
  - `\the<type>`:** Number. `\thetable` prints the number of the table, etc.
  - `\p@<type>`:** Parent’s number. Prints the number of the [within] figure, etc.
  - `\fnum@<type>`:** Prints the figure number for the caption.  
`\<type>name \the<type>`, “Figure 123”.
  - `\<type>`:** Starts the float environment. `\figure` or `\begin{figure}`
  - `\end<type>`:** Ends the float environment. `\endfigure` or `\end{figure}`
  - `\tf@<ext>`:** The  $\TeX$  file identifier for the output file.
  - `LWR@have<type>`:** A boolean remembering whether a `\listof` was requested for a float of this type.
  - File with extension `lo<f,t,a-z>`:** An output file containing the commands to build the `\listof<type>` “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.
- 

## 66 Floats

Floats are supported, although partially through emulation.

Table 11 shows the data structure associated with each `<type>` of float.

`\@makecaption` is redefined to print the float number and caption text, separated by `\CaptionSeparator`, which works with the `babel` package to adjust the caption separator according to the language. French, for example, uses an en-dash instead of a colon: “Figure 123 – Caption text”.

## 66.1 Float captions

for HTML output: 7072 \begin{warpHTML}

\LWR@floatbegin {<type>} [<placement>]

Begins a \newfloat environment.

```
7073 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
7074 \ifbool{FormatWP}{\newline}{}%
7075 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
7076 \addtocounter{LWR@thisautoid}{1}%
7077 \booltrue{LWR@freezethisautoid}%
```

```
7078 \begingroup%
```

Settings while inside the environment:

```
7079 \LWR@origraggedright%
```

Open an HTML figure tag:

```
7080 \LWR@htmltag{figure id="\LWR@origmbox{autoid-\arabic{LWR@thisautoid}}" class="#1"}%
7081 \ifbool{FormatWP}{%
7082   \LWR@orignewline%
7083   \LWR@BlockClassWP}{\}{wp#1}%
7084 }{}%
```

```
7085 \renewcommand*{\@capttype}{#1}%
7086 \caption@settype{#1}%
7087 \LWR@startpars%
7088 \ifbool{expr{bool{FormatWP} and bool{WPMarkFloats}}}{%
7089
7090   === begin #1 ===
7091
7092 }{}%
7093 }
```

\@float Support packages which create floats directly.  
 \@dblfloat

```
7094 \let\@float\LWR@floatbegin
7095 \let\@dblfloat\LWR@floatbegin
```

`\LWR@floatend` Ends a `\newfloat` environment.

```
7096 \newcommand*{\LWR@floatend}{%
7097 \ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%
7098
7099 === end ===
7100
7101 }{}%
7102 \LWR@stoppars%
```

Close an HTML figure tag:

```
7103 \ifbool{FormatWP}{\endLWR@BlockClassWP}{}%
7104 \LWR@htmlElementend{figure}%
7105 \endgroup%
7106 \boolfalse{LWR@freezethisautoid}%
7107 \LWR@startpars%
7108 \ifbool{FormatWP}{\newline}{}%
7109 }
```

`\end@float` Support packages which create floats directly.  
`\end@dblfloat`

```
7110 \let\end@float\LWR@floatend
7111 \let\end@dblfloat\LWR@floatend
```

**Ctr** `LWR@thisautoid` 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.

```
7112 \newcounter{LWR@thisautoid}
```

**Ctr** `LWR@thisautoidWP` A sequential counter for all word processor conversion `<div>`s. This is used to convince LIBREOFFICE to form a frame around this element.

```
7113 \newcounter{LWR@thisautoidWP}
```

**Bool** `LWR@freezethisautoid` Prevents multiple increments of `\LWR@thisautoid` inside a float.

```
7114 \newbool{LWR@freezethisautoid}
7115 \boolfalse{LWR@freezethisautoid}
```

`\LWR@newautoidanchor` Adds a new `<autoid>` anchor.

```
7116 \newcommand*{\LWR@newautoidanchor}{%
7117 \ifbool{LWR@freezethisautoid}{}%
7118 \addtocounter{LWR@thisautoid}{1}%
7119 \LWR@htmltag{a id="\LWR@origmbbox{autoid-\arabic{LWR@thisautoid}}"}\LWR@htmltag{/a}%
```



```

7120 }%
7121 }

```

`\@capttype` Remembers which float type is in use.

```

7122 \newcommand*\@capttype{}

```

### 66.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```

7123 \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.

```

7124 \AtBeginDocument{\renewcommand*\@makecaption}[2]{%
7125   \LWR@traceinfo{\@makecaption}%
7126   #1\CaptionSeparator#2%
7127   \LWR@traceinfo{\@makecaption: done}%
7128 }%
7129 }

```

### 66.1.2 Caption and LOF linking and tracking

When a new HTML file is marked in the  $\text{\LaTeX}$  PDF file, the  $\text{\LaTeX}$  page number at that point is stored in `LWR@latestautopage`, (and the associated filename is remembered by the special  $\text{\LaTeX}$  labels). This page number is used to generate an `autopage` 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 `autoid <id>` at the start of the float itself in the HTML file. The `autopage` and `autoid` 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.

Ctrl `LWR@nextautoid` Tracks autoid for floats. Tracks autopage for floats.

Ctrl `LWR@nextautopage` These are updated per float as the `.lof`, `.lot` file is read.

```

7130 \newcounter{LWR@nextautoid}
7131 \newcounter{LWR@nextautopage}

```

`\LWRsetnextfloat`  $\{\langle autopage \rangle\} \{\langle autoid \rangle\}$

This is written to the .lof, .lot file just before each float's usual entry. The autopage and autoid are remembered for \l@figure to use when creating the HTML links.

```
7132 \newcommand*\LWRsetnextfloat}[2]{%
7133 \setcounter{LWR@nextautopage}{#1}%
7134 \setcounter{LWR@nextautoid}{#2}%
7135 }
```

**Ctr** LWR@latestautopage Updated each time a new HTML file is begun. \LWRsetnextfloat is written with this and the autoid by the modified \addcontentsline just before each float's entry.

```
7136 \newcounter{LWR@latestautopage}
7137 \setcounter{LWR@latestautopage}{1}
```

**Env** LWR@figcaption Encapsulates a caption inside <figcaption>, and if FormatWP then also a <div> with an italic style.

```
7138 \newenvironment*LWR@figcaption
7139 {%
7140 \LWR@traceinfo{LWR@figcaption env start}%
7141 \LWR@htmlblocktag{figcaption}%
7142 \ifbool{FormatWP}{%
7143 \begin{BlockClass}[font-style:italic]{italic}
7144 \LWR@origvspace*\baselineskip
7145 }{}%
7146 \LWR@traceinfo{LWR@figcaption env start: done}%
7147 }
7148 {%
7149 \LWR@traceinfo{LWR@figcaption env end}%
7150 \ifbool{FormatWP}{\end{BlockClass}}{}%
7151 \LWR@htmlblocktag{/figcaption}%
7152 \LWR@traceinfo{LWR@figcaption env end: done}%
7153 }
```

After packages have loaded, remember the print-mode version of the following:

```
7154 \AtBeginDocument{
7155 \LetLtxMacro\LWR@origcaption@begin\caption@begin
7156 \LetLtxMacro\LWR@origcaption@end\caption@end
7157 }
```

**\LWR@caption@begin** Low-level patches to create HTML tags for captions.

```
7158 \newcommand*\LWR@caption@begin}[1]
7159 {
7160 \LWR@traceinfo{LWR@caption@begin}%
```

Keep par and minipage changes local:

```
7161 \begingroup%
```

The caption code was not allowing the closing par tag:

```
7162 \@setpar{\LWR@closeparagraph\@@par}%
```

No need for a minipage or \parbox inside the caption:

```
7163 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
```

```
7164 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
7165 \LWR@figcaption%
```

```
7166 \LWR@traceinfo{\LWR@caption@begin: about to LWR@origcaption@begin}%
```

```
7167 \LWR@origcaption@begin{#1}%
```

```
7168 \LWR@traceinfo{\LWR@caption@begin: done}%
```

```
7169 }
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
7170 \newcommand{\LWR@caption@end}
```

```
7171 {%
```

```
7172 \LWR@traceinfo{\LWR@caption@end}%
```

```
7173 \LWR@origcaption@end%
```

Closing tag:

```
7174 \endLWR@figcaption%
```

```
7175 \endgroup%
```

```
7176 % \leavevmode% avoid bad space factor (0) error
```

```
7177 \LWR@traceinfo{\LWR@caption@end: done}%
```

```
7178 }
```

`\caption@begin` Low-level patches to create HTML tags for captions.

`\caption@end`

```
7179 \AtBeginDocument{
```

```
7180 \let\caption@begin\LWR@caption@begin
```

```
7181 \let\caption@end\LWR@caption@end
```

```
7182 }
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```

7183 \let\LWR@origcaptionlistentry\captionlistentry
7184
7185 \renewcommand*{\captionlistentry}{%
7186 \LWR@ensuredoingapar%
7187 \LWR@origcaptionlistentry%
7188 }
7189
7190 \def\LWR@LTcaptionlistentry{%
7191 \LWR@ensuredoingapar%
7192 \LWR@htmltag{a id="\LWR@origmbox{autoid-\arabic{LWR@thisautoid}}"}\LWR@htmltag{/a}%
7193 \bgroup
7194 \@ifstar{\egroup\LWR@LTcaptionlistentry}% gobble *
7195 {\egroup\LWR@LTcaptionlistentry}}%
7196 \def\LWR@LTcaptionlistentry#1{%
7197 \caption@listentry\@firstoftwo[\LTcaptype]{#1}}%

```

`\addcontentsline` Patched to write the autopage and autoid 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.

```

7198 \let\LWR@origaddcontentsline\addcontentsline
7199
7200 \renewcommand*{\addcontentsline}[3]{%
7201 \ifstrequal{#1}{toc}{}% not TOC
7202 \LWR@newautoidanchor%
7203 \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{}%
7204 \addtocontents{\@nameuse{ext@#2}}{%
7205 \protect\LWRsetnextfloat%
7206 {\arabic{LWR@latestautopage}}%
7207 {\arabic{LWR@thisautoid}}%
7208 }%
7209 }% not TOC
7210 \LWR@origaddcontentsline{#1}{#2}{#3}%
7211 }

```

Pkg `capt-of` Either package provides `\captionof`, which is later patched at the beginning of the document.

Pkg `caption`

`\captionof` Patched to handle paragraph tags.

```

7212 \AtBeginDocument{
7213 \let\LWR@origcaptionof\captionof
7214
7215 \renewcommand*{\captionof}{%
7216 \LWR@stoppars
7217 \LWR@origcaptionof%
7218 }
7219 }

```

```
7220 \end{warpHTML}
```

## 67 Table of Contents, LOF, LOT

This section controls the generation of the TOC, LOF, and LOT.

The .toc, .lof, and .lot files are named by the source code \jobname.

In HTML, the printed tables are placed inside a <div> of class toc, lof, or lot.

A “sidetoc” is provided which prints a subset of the toc on the side of each page other than the homepage.

The regular  $\TeX$  infrastructure is used for TOC, along with some patches to generate HTML output.

**for HTML output:** 7221 \begin{warpHTML}

### 67.1 Reading and printing the TOC

```
\LWR@myshorttoc {<toc/lof/lot/sidetoc>}
```

Reads in and prints the TOC/LOF/LOT at the current position. While doing so, makes the @ character into a normal letter to allow formatting commands in the section names.

Unlike in regular  $\TeX$ , the file is not reset after being read, since the sideroc may be referred to again in each HTML page.

```
7222 \newcommand*{\LWR@myshorttoc}[1]{%
7223 \LWR@traceinfo{\LWR@myshorttoc: #1}%
7224 \LWR@ensuredoingapar%
```

Only if the file exists:

```
7225 \IfFileExists{\jobname.#1}{%
7226 \LWR@traceinfo{\LWR@myshorttoc: loading}%
```



Make @ a regular letter. Many of the commands in the file will have @ characters in them, so @ must be made a regular letter.



**disabled**

For pdf<sub>l</sub>atex, 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.

```

7227 \begingroup%
7228 % \ifxetexorluatex%
7229 % \else
7230 % \inputencoding{latin1}% currently disabled
7231 % \fi
7232 \makeatletter%

```

Read in the TOC file:

```

7233 \@input{\jobname.#1}%
7234 % \makeatother
7235 \endgroup%
7236 }%
7237 {}%
7238 \LWR@traceinfo{LWR@myshorttoc: done}%
7239 }

```

`\LWR@subtableofcontents`  $\{\langle toc/lof/lot \rangle\} \{\langle sectionstarname \rangle\}$

Places a TOC/LOF/LOT at the current position.

```
7240 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```

7241 \@ifundefined{chapter}
7242 {\LWR@closeprevious{\LWR@depthsection}}
7243 {\LWR@closeprevious{\LWR@depthchapter}}

```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
7244 \LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
7245 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}
```

Create a new HTML nav containing the TOC/LOF/LOT:

```
7246 \LWR@htmlelementclass{nav}{#1}
```

Create the actual list:

```
7247 \LWR@myshorttoc{#1}
```

Close the nav:

```
7248 \LWR@html elementclassend{nav}{#1}
7249 }
```

`\@starttoc` `{\langle ext\rangle}`

Patch `\@starttoc` to encapsulate the toc inside HTML tags:

```
7250 \let\LWR@orig@starttoc\@starttoc
7251
7252 \renewcommand{\@starttoc}[1]{
7253 \LWR@html elementclass{nav}{#1}
7254 \LWR@orig@starttoc{#1}
7255 \LWR@html elementclassend{nav}{#1}
7256 }
```

Bool `LWR@copiedsidetoc` Used to only copy the toc file to the sidetoc a single time.

(listings and perhaps other packages would re-use `\tableofcontents` for their own purposes, causing the sidetoc to be copied more than once, and thus end up empty.)

```
7257 \newbool{LWR@copiedsidetoc}
7258 \boolfalse{LWR@copiedsidetoc}
```

`\tableofcontents` Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.

```
7259 \AtBeginDocument{
7260 \let\LWR@origtableofcontents\tableofcontents
7261
7262 \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:

```
7263 \ifboolexpr{bool{FormatWP} and bool{WPMarkTOC}}{
7264
7265 === table of contents ===
7266
7267 }
7268 {
```

Copy the `.toc` file to `.sidetoc` for printing the sideroc. The original `.toc` file is renewed when `\tableofcontents` is finished.

```
7269 \ifbool{LWR@copiedsidetoc}{\{%
7270 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
```

```

7271      \booltrue{LWR@copiedsidetoc}%
7272    }%
7273    \LWR@printpendingfootnotes
7274    \LWR@origtableofcontents
7275  }
7276 }% \tableofcontents
7277 }% AtBeginDocument

```

### `\listoffigures`

```

7278 \let\LWR@origlistoffigures\listoffigures
7279
7280 \renewcommand*{\listoffigures}{
7281 \ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{
7282
7283 === list of figures ===
7284
7285 }
7286 {
7287   \LWR@printpendingfootnotes
7288   \LWR@origlistoffigures
7289 }
7290 }

```

### `\listoftables`

```

7291 \let\LWR@origlistoftables\listoftables
7292
7293 \renewcommand*{\listoftables}{
7294 \ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{
7295
7296 === list of tables ===
7297
7298 }
7299 {
7300   \LWR@printpendingfootnotes
7301   \LWR@origlistoftables
7302 }
7303 }

```

## 67.2 High-level TOC commands

`\listof`  $\{\langle type \rangle\} \{\langle title \rangle\}$



Emulate the `\listof` command from the float package (section 167). Used to create lists of custom float types. Also used to redefine the standard  $\TeX$  `\listoffigures` and `\listoftables` commands.

```

7304 \NewDocumentCommand{\listof}{m +m}{%
7305 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
7306 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
7307 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
7308     \jobname.\csuse{ext@#1}\relax
7309 }
```

### 67.3 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 sideroc.

**div.sidetoctitle:** The title.

**div.sidetoccontents:** The table of contents.

---

```

7310 \end{warpHTML}
```

**for HTML & PRINT:** 7311 `\begin{warpall}`

**Ctrl SideTOCDepth** Controls how deep the side-TOC gets. Use a standard  $\TeX$  section level similar to `tocdepth`.

```

7312 \newcounter{SideTOCDepth}
7313 \setcounter{SideTOCDepth}{1}
```

`\sidetocname` Holds the default name for the sideroc.

```

7314 \newcommand{\sidetocname}{Contents}
```

```

7315 \end{warpall}
```

for HTML output: 7316 \begin{warpHTML}

\LWR@sidetoc Creates the actual side-TOC.

```
7317 \newcommand*{\LWR@sidetoc}{
7318 \LWR@forcenewpage
7319 \LWR@stoppars
7320
```

The entire siderOC is placed into a nav of class sidetoc.

```
7321 \LWR@htmlclass{nav}{sidetoc}
7322
7323 \setcounter{tocdepth}{\value{SideTOCDepth}}
7324
```

The title is placed into a <div> of class sidetoc title, and may contain paragraphs.

```
7325 \begin{BlockClass}{sidetoc title}
7326 \sidetocname
7327 \end{BlockClass}
```

The table of contents is placed into a <div> of class sidetoc contents.

```
7328 \begin{BlockClass}{sidetoc contents}
7329 \LinkHome
7330
7331 \LWR@myshorttoc{sidetoc}
7332 \end{BlockClass}
7333 \LWR@htmlclassend{nav}{sidetoc}
7334 }
```

## 67.4 Low-level TOC line formatting

\numberline {<number>}

(Called from each line in the .aux, .lof files.)

Record this section number for further use:

```
7335 \newcommand*{\LWR@numberline}[1]{%
7336 \LWR@sectionnumber{#1}\quad%
7337 }
7338
7339 \LetLtxMacro\numberline\LWR@numberline
```

`\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 opens.

**#1** is depth

**#2** is section, subsection, etc.

**#3** the text of the caption

**#4** page number

```
7340 \NewDocumentCommand{\hypertoc}{m m +m m}{%
7341 \LWR@traceinfo{hypertoc !#1!#2!#3!#4!}%
```

Respond to `tocdepth`:

```
7342 \ifthenelse{\cnttest{#1}{<=}}{\value{tocdepth}}{%
7343 \LWR@startpars%
```

Create an HTML link to `filename#autosec-(page)`, with text of the caption, of the given HTML class.

```
7344 \LWR@subhyperrefclass{%
7345 \LWR@htmlrefsectionfilename{autopage-#4}\#\LWR@origmbox{autosec-#4}%
7346 }{#3}{toc#2}%
7347 \LWR@stoppars%
7348 }
7349 {}
7350 \LWR@traceinfo{hypertoc done}%
7351 }
```

**Ctr** `lofdepth` TOC depth for figures.

```
7352 \@ifclassloaded{memoir}{}{
7353 \newcounter{lofdepth}
7354 \setcounter{lofdepth}{1}
7355 }
```

**Ctr** `lotdepth` TOC depth for tables.

```
7356 \@ifclassloaded{memoir}{}{
7357 \newcounter{lotdepth}
7358 \setcounter{lotdepth}{1}
7359 }
```

`\hypertocfloat`  $\{\langle 1: depth \rangle\} \{\langle 2: type \rangle\} \{\langle 3: ext of parent \rangle\} \{\langle 4: caption \rangle\} \{\langle 5: page \rangle\}$

**#1** is depth

**#2** is figure, table, etc.

**#3** is lof, lot, of the parent.

**#4** the text of the caption

**#5** page number

```
7360 \newcommand{\hypertocfloat}[5]{%
7361 \LWR@startpars
```

If some float-creation package has not yet defined the float type's lofdepth counter, etc, define it here:

```
7362 \@ifundefined{c@#3depth}{%
7363 \newcounter{#3depth}%
7364 \setcounter{#3depth}{1}%
7365 }{}%
```

Respond to lofdepth, etc.:

```
7366 \LWR@traceinfo{\hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
7367 \ifthenelse{\cnttest{#1}{<=}}{\arabic{#3depth}}{}%
7368 \LWR@startpars%
```

Create an HTML link to filename#autoid-(float number), with text of the caption, of the given HTML class.

```
7369 \LWR@subhyperrefclass{%
7370 \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
7371 \#\LWR@origmbox{autoid-\arabic{LWR@nextautoid}}}%
7372 {#4}{toc#2}%
7373 \LWR@stoppars%
7374 }{}%
7375 }
```

Automatically called by `\contentsline`:

`\l@part`  $\{\langle name \rangle\} \{\langle page \rangle\}$

Uses `\DeclareDocumentCommand` in case the class does not happen to have a `\part`.

```
7376 \DeclareDocumentCommand{\l@part}{m m}{\hypertoc{-1}{part}{#1}{#2}}
```

`\l@chapter`  $\{\langle name \rangle\} \{\langle page \rangle\}$

Uses `\DeclareDocumentCommand` in case the class does not happen to have a `\chapter`.

```
7377 \DeclareDocumentCommand{\l@chapter}{m m}
7378   {\hypertoc{0}{chapter}{#1}{#2}}
```

`\l@section`  $\{\langle name \rangle\} \{\langle page \rangle\}$

```
7379 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
```

`\l@subsection`  $\{\langle name \rangle\} \{\langle page \rangle\}$

```
7380 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
```

`\l@subsubsection`  $\{\langle name \rangle\} \{\langle page \rangle\}$

```
7381 \renewcommand{\l@subsubsection}[2]{\hypertoc{3}{subsubsection}{#1}{#2}}
```

`\l@paragraph`  $\{\langle name \rangle\} \{\langle page \rangle\}$

```
7382 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
```

`\l@subparagraph`  $\{\langle name \rangle\} \{\langle page \rangle\}$

```
7383 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
```

`\l@figure`  $\{\langle name \rangle\} \{\langle page \rangle\}$

```
7384 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
```

`\l@table`  $\{\langle name \rangle\} \{\langle page \rangle\}$

```
7385 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}
```

```
7386 \end{warpHTML}
```

## 68 Index and glossary

See:

<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: 7387 `\begin{warpHTML}`

```
7388 \newcounter{LWR@autoindex}
7389 \setcounter{LWR@autoindex}{0}
7390
7391 \newcounter{LWR@autoglossary}
7392 \setcounter{LWR@autoglossary}{0}
```

`\printindex`

```
7393 \let\LWR@origprintindex\printindex
7394
7395 \renewcommand*\printindex{
7396 {
7397 \LWR@startpars
7398 \LWR@origprintindex
7399 }
```

Env `theindex`

```
7400 \@ifundefined{chapter}
7401 {\newcommand*\LWR@indexsection}[1]{\section*{#1}}
7402 {\newcommand*\LWR@indexsection}[1]{\chapter*{#1}}
7403
7404 \renewenvironment*{theindex}{%
7405 \LWR@indexsection{\indexname}%
7406 \let\item\LWR@indexitem%
7407 \let\subitem\LWR@indexsubitem%
7408 \let\subsubitem\LWR@indexsubsubitem%
7409 }{}
```

`\LWR@indexitem`

```
7410 \newcommand*\LWR@indexitem){
7411
7412 \InlineClass{indexitem}{
7413 }
```

`\LWR@indexitem`

```
7414 \newcommand*\LWR@indexsubitem){
```

```

7415
7416 \InlineClass{indexsubitem}{ }
7417 }

```

\LWR@indexitem

```

7418 \newcommand{\LWR@indexsubsubitem}{
7419
7420 \InlineClass{indexsubsubitem}{ }
7421 }

```

\@wrindex     $\langle term \rangle$     Redefined to write the LWR@autoindex counter instead of page

```

7422 \def\LWR@wrindex#1{%
7423 \addtocounter{LWR@autoindex}{1}%
7424 \LWR@newlabel{LWRindex-\arabic{LWR@autoindex}}%
7425 \protected@write\@indexfile{%
7426 {\string\indexentry{#1}{\arabic{LWR@autoindex}}}%
7427 \endgroup
7428 \@esphack}
7429
7430 \let\@wrindex\LWR@wrindex

```

\@wrglossary     $\langle term \rangle$     Redefined to write the LWR@latestautopage counter instead of page

```

7431 \def\@wrglossary#1{%
7432 \addtocounter{LWR@autoglossary}{1}%
7433 \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
7434 \protected@write\@glossaryfile{%
7435 {\string\glossaryentry{#1}{\theLWR@autoglossary}}%
7436 \endgroup
7437 \@esphack}

```

\hyperindexref     $\langle autosecnumber \rangle$

\hyperindexref{web address} is inserted into \*.ind by the xindy style file lwarp.xdy

```

7438 \newcommand*\hyperindexref[1]{\nameref{LWRindex-#1}}

7439 \end{warpHTML}

```

**for PRINT output:**    A null command for print mode, in case hyperref was not used:

```

7440 \begin{warpprint}
7441 \newcommand{\hyperindexref}[1]{#1}
7442 \end{warpprint}

```

**for HTML & PRINT:** For the glossaries package, try to prevent an error where `\glo@name` was not found:

```
7443 \begin{warpall}
7444 \providecommand{\glo@name}{}
7445 \end{warpall}
```

## 69 Bibliography presentation

**for HTML output:** 7446 `\begin{warpHTML}`

`\bibliography` `{\filenames)}`

Modified to use the base jobname instead of the `_html` jobname.

```
7447 \def\bibliography#1{%
7448   \if@filesw
7449     \immediate\write\@auxout{\string\bibdata{#1}}%
7450   \fi
7451   \@input@{\jobname.bbl}% original
7452   \@input@{\BaseJobname.bbl}% lwarp
7453 }
```

`\@biblabel` `{\text-refnumber)}`

Modified to use the base jobname instead of the `_html` jobname.

```
7454 \renewcommand{\@biblabel}[1]{[#1]\quad}
```

Env `thebibliography` To emphasize document titles in the bibliography, the following redefines `\em` inside `thebibliography` to gather everything until the next closing brace, then display these tokens with `\textit`.

*Adapted from `embracedef.sty`, which is by TAKAYUKI YATO:*

<https://gist.github.com/zr-tex8r/b72555e3e7ad2f0a37f1>

```
7455 \AtBeginDocument{
7456 \AtBeginEnvironment{thebibliography}{
7457 \providecommand*\LWR@newem}[1]{\textit{#1}}
7458
7459 \renewrobustcmd{\em}{%
7460   \begingroup
7461     \gdef\LWR@em@after{\LWR@em@finish\LWR@newem}%
7462     \afterassignment\LWR@em@after
7463     \toks@\bgroup
```



```

7464 }
7465
7466 \def\LWR@em@finish#1{%
7467     \xdef\LWR@em@after{\noexpand#1{\the\toks@}}%
7468     \endgroup
7469     \LWR@em@after\egroup
7470 }
7471 }% \AtBeginEnvironment{thebibliography}
7472 }% \AtBeginDocument

7473 \end{warpHTML}

```

## 70 Restoring original formatting

for HTML output: 7474 \begin{warpHTML}

`\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`.

```

7475 \newcommand*{\LWR@restoreorigformatting}{%
7476 \LWR@traceinfo{\LWR@restoreorigformatting}%
7477 \linespread{1}%

7478 \LetLtxMacro\caption@begin\LWR@origcaption@begin%
7479 \LetLtxMacro\caption@end\LWR@origcaption@end%
7480 \let\par\LWR@origpar%

7481 \LetLtxMacro\ref\LWR@origref%{ } syntax highlighting

7482 \let\normalsize\LWR@orignormalsize%
7483 \let\small\LWR@origsmall%
7484 \let\footnotesize\LWR@origfootnotesize%
7485 \let\scriptsize\LWR@origscriptsize%
7486 \let\tiny\LWR@origtiny%
7487 \let\large\LWR@origlarge%
7488 \let\Large\LWR@origLarge%
7489 \let\LARGE\LWR@origLARGE%
7490 \let\huge\LWR@orighuge%
7491 \let\Huge\LWR@origHuge%

```

```

7492 \RenewDocumentCommand{\InlineClass}{o m +m}{##3}%
7493 \RenewDocumentEnvironment{BlockClass}{o m}{-}{-}%
7494 \renewcommand{\BlockClassSingle}[2]{##2}%
7495 \LetLtxMacro{\hspace}{\LWR@orighspace}%

7496 \LetLtxMacro\hfill\LWR@orighfill%
7497 \LetLtxMacro\hfil\LWR@orighfil%
7498 \LetLtxMacro\rule\LWR@origrule%
7499 \LetLtxMacro\hrulefill\LWR@orighrulefill%
7500 \LetLtxMacro\dotfill\LWR@origdotfill%
7501 \let\vspace\LWR@origvspace%
7502 \let\hss\LWR@orighss%
7503 \let\llap\LWR@origllap%
7504 \let\rlap\LWR@origrlap%
7505 \let\hfilneg\LWR@orighfilneg%

7506 \let\raggedright\LWR@origraggedright
7507 \let\raggedleft\LWR@origraggedleft
7508 \let\centering\LWR@origcentering

7509 \let\,\LWR@origcomma% disable HTML short unbreakable space
7510 \let\textellipsis\LWR@origtextellipsis%
7511 \let\textless\LWR@origtextless%
7512 \let\textgreater\LWR@origtextgreater%
7513 \LetLtxMacro{\textrm}{\LWR@origtextrm}%
7514 \LetLtxMacro{\textsf}{\LWR@origtextsf}%
7515 \LetLtxMacro{\texttt}{\LWR@origtexttt}%
7516 \LetLtxMacro{\textbf}{\LWR@origtextbf}%
7517 \LetLtxMacro{\textmd}{\LWR@origtextmd}%
7518 \LetLtxMacro{\textit}{\LWR@origtextit}%
7519 \LetLtxMacro{\textsl}{\LWR@origtextsl}%
7520 \LetLtxMacro{\textsc}{\LWR@origtextsc}%
7521 \LetLtxMacro{\textup}{\LWR@origtextup}%
7522 \LetLtxMacro{\textnormal}{\LWR@origtextnormal}%
7523 \LetLtxMacro{\emph}{\LWR@origemph}%
7524 \LetLtxMacro{\rmfamily}{\LWR@origrmfamily}%
7525 \LetLtxMacro{\sffamily}{\LWR@origsffamily}%
7526 \LetLtxMacro{\ttfamily}{\LWR@origttfamily}%
7527 \LetLtxMacro{\bfseries}{\LWR@origbfseries}%
7528 \LetLtxMacro{\mdseries}{\LWR@origmdseries}%
7529 \LetLtxMacro{\upshape}{\LWR@origupshape}%
7530 \LetLtxMacro{\slshape}{\LWR@origslshape}%
7531 \LetLtxMacro{\scshape}{\LWR@origscshape}%
7532 \LetLtxMacro{\itshape}{\LWR@origitshape}%
7533 \LetLtxMacro{\em}{\LWR@origem}%
7534 \LetLtxMacro{\normalfont}{\LWR@orignormalfont}%
7535 \let\sp\LWR@origsp%
7536 \let\sb\LWR@origsb%

```

---

```

7537 \LetLtxMacro\textsuperscript\LWR@origtextsuperscript%
7538 \LetLtxMacro\@textsuperscript\LWR@orig@textsuperscript%
7539 \LetLtxMacro\textsubscript\LWR@origtextsubscript%
7540 \LetLtxMacro\@textsubscript\LWR@orig@textsubscript%
7541 \LetLtxMacro\underline\LWR@origunderline%
7542 \let~\LWR@origtilde%
7543 \let\enskip\LWR@origenskip%
7544 \let\quad\LWR@origquad%
7545 \let\qquad\LWR@origqquad%
7546 \LetLtxMacro\tabular\LWR@origtabular%
7547 \LetLtxMacro\endtabular\LWR@origendtabular%
7548 \LetLtxMacro\noalign\LWR@orignoalign%
7549 \LetLtxMacro\hline\LWR@orighline%
7550 \LetLtxMacro\toprule\LWR@origtoprule%
7551 \LetLtxMacro\midrule\LWR@origmidrule%
7552 \LetLtxMacro\cmidrule\LWR@origcmidrule%
7553 \LetLtxMacro\bottomrule\LWR@origbottomrule%
7554 \LetLtxMacro\addlinespace\LWR@origaddlinespace%
7555 \LetLtxMacro\morecmidrules\LWR@origmorecmidrules%
7556 \LetLtxMacro\specialrule\LWR@origspecialrule%
7557 \let\newline\LWR@orignewline%
7558 \LetLtxMacro{\raisebox}{\LWR@origraisebox}%
7559 \LetLtxMacro{\includegraphics}\LWR@origincludegraphics%
7560 \LetLtxMacro{\scalebox}{\LWR@origscalebox}%
7561 \LetLtxMacro{\rotatebox}{\LWR@origrotatebox}%
7562 \let\reflectbox\LWR@origreflectbox%
7563 \LetLtxMacro\resizebox\LWR@origresizebox%
7564 \let\framebox\LWR@origframebox%

7565 \LetLtxMacro\mbox\LWR@origmbox%

7566 \let\makebox\LWR@origmakebox%
7567 \let\fbbox\LWRprint@fbbox%
7568 \let\fbboxBlock\LWRprint@fbbox%
7569 \LetLtxMacro{\fminipage}{\LWRprint@fminipage}%
7570 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}%
7571 \LetLtxMacro{\minipage}{\LWR@origminipage}%
7572 \let\endminipage\LWR@origendminipage%
7573 \LetLtxMacro{\parbox}{\LWR@origparbox}%
7574 \let\TeX\LWR@origTeX%
7575 \let\LaTeX\LWR@origLaTeX%
7576 \let\LaTeXe\LWR@origLaTeXe%
7577 \renewcommand*{\Xe}{X\textsubscript{E}}%

7578 \LetLtxMacro\@ensuredmath\LWR@origensuredmath%

7579 \csletcs{equation*}{LWR@origequationstar}%
7580 \csletcs{endequation*}{LWR@origendequationstar}%

```

```

7581 %
7582 \LWR@restoreorigaccents%
7583 \LWR@restoreoriglists%
7584 %
7585 \LWR@FBcancel%
7586 }

7587 \end{warpHTML}

```

## 71 Math

### 71.1 Limitations

#### 71.1.1 Rendering tradeoffs

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

**SVG files** Rendering math as images creates a new SVG file for each expression, except that an MD5 hash is used to combine identical duplicates of the same inline math expression into a single file, which must be converted to SVG only once. Display math is still handled as individual files, since it may contain labels or references which are likely to change.

**SVG inline** The SVG images are currently stored separately, but they could be encoded in-line directly into the HTML document. 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  $\text{\LaTeX}$ -to-HTML 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.

#### 71.1.2 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>15</sup>, then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by  $\text{\LaTeX}$  with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML `alt` attribute carries the  $\text{\LaTeX}$  code which generated the math, allowing copy/paste of the  $\text{\LaTeX}$  math expression into other documents.

**SVG image font size** The size of the math and text used in the SVG image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, for ex:

```
\renewcommand{\LateximageFontSizeName}{large}
```

**SVG math copy/paste** For SVG math, text copy/paste from the HTML `<alt>` tags lists the equation number or tag for single equations, along with the  $\text{\LaTeX}$  code for the math expression. For  $\mathcal{AMS}$  environments with multiple numbers in the same environment, only the first and last is copy/pasted, as a range. No tags are listed inside a starred  $\mathcal{AMS}$  environment, although the `\tag` macro will still appear inside the  $\text{\LaTeX}$  math expression.

 **SVG math in  $\text{\TeX}$  boxes** SVG math does not work inside  $\text{\TeX}$  boxes, since a `\newpage` is required before and after each image.

### 71.1.3 MATHJAX option

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

Prog MathJax

When MATHJAX is enabled, math is rendered twice:

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

**MATHJAX limitations** Limitations when using MATHJAX include:

Prog MathJax

**chapter numbers**

- In document classes which have chapters, `\tagged` equations have the chapter number prepended in HTML output, unlike  $\text{\LaTeX}$ . `\tag*` equations (correctly) do not. This may be improved with future versions of the MATHJAX support script.

<https://groups.google.com/forum/#!topic/mathjax-users/jUtewUcE2bY>

<sup>15</sup>See section 332 regarding fonts and fractions.

- subequations

- MATHJAX itself does not support subequations. This may be improved by parsing the  $\TeX$  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.

 siunitx inside an equation

<https://github.com/burnpanck/MathJax-siunitx>

Also see section 8.6.5.
- tabbing

- A tabbing environment is emulated using an HTML `<pre>`. While MATHJAX is enabled inside tabbing, the browser may not correctly render the horizontal alignment of the math and text following after on the same line.
-  other macros and packages

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

- MATHJAX does not automatically support custom  $\TeX$  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}

```

## 71.2 Inline and display math

for HTML output: 7588 `\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.

```
7589 \let\LWR@origtextdollar\$
7590
7591 \renewcommand*{\$}{%
7592 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
7593 {\LWR@origtextdollar}%
7594 {\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
7595 }
```

Ctr `LWR@externalfilecnt` Counter for the external files which are generated and then referenced from the HTML:

```
7596 \newcounter{LWR@externalfilecnt}

7597 \LetLtxMacro\LWR@origdollar$
7598 \LetLtxMacro\LWR@secondorigdollar$% balance for editor syntax highlighting

7599 \LetLtxMacro\LWR@origopenparen\(  
7600 \LetLtxMacro\LWR@origcloseparen\  
7601 \LetLtxMacro\LWR@origopenbracket\[  
7602 \LetLtxMacro\LWR@origclosebracket\]
```

`$` Redefine the dollar sign to place math inside a `lateximage`, or use MATHJAX:  
`$$`

```
7603 \begingroup
7604 \catcode'\$=\active%
7605 \protected\gdef$ {\ifnextchar$\LWR@doubledollar\LWR@singledollar}%
```

`\LWR@doubledollar` Redefine the double dollar sign to place math inside a `lateximage`, or use MATHJAX:

```
7606 \protected\gdef\LWR@doubledollar$#1$${
```

If MATHJAX or formatting for a word processor, print the  $\LaTeX$  expression:

```
7607 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{%
```

For MATHJAX, print the math between `\[` and `\]`:

```
7608 {
7609
7610 \textbackslash[%
7611 \LWR@HTMLsanitize{#1}%
7612 \textbackslash]
```

```

7613
7614 }% mathjax

```

For SVG, print the math inside a `lateximage`, with an `<alt>` tag of the  $\text{\LaTeX}$  code:

```

7615 {% not mathjax
7616
7617     \begin{BlockClass}{displaymath}
7618     \begin{lateximage}%
7619     [%
7620         \textbackslash{[] % extra space
7621         \LWR@HTMLsanitize{#1} % extra space
7622         \textbackslash{[]}%
7623     ]%
7624     \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
7625     \end{lateximage}%
7626     \end{BlockClass}
7627
7628 }% not mathjax
7629 }%

```

`\LWR@subsingledollar` `{\langle alt text \rangle}{\langle math expression \rangle}` Redefine the single dollar sign to place math inside a `lateximage`, or use `MATHJAX`: If the alt text is empty, the math expression is used for the alt text.

```

7630 \newlength{\LWR@singledollarwidth}
7631 \newlength{\LWR@singledollarheight}
7632 \newlength{\LWR@singledollardepth}
7633
7634 \protected\gdef\LWR@subsingledollar#1#2{%
7635 \begingroup%

```

MathJax cannot parse the often complicated  $\text{\TeX}$  expressions which appear in the various uses of `\ensuredmath`. `\ensuremath` forces the alt tag to “math image”. If this is the case, force the use of a `lateximage` even if MathJax. Likewise for `siunitx` if `parse-numbers=false`.

If MathJax, or if formatting math for a word processor, print the math expression.

```

7636 \ifboolexpr{%
7637     (
7638         bool{mathjax} or
7639         ( bool{FormatWP} and bool{WPMarkMath} )
7640     ) and
7641     ( not test { \ifstrequal {#1} {math image} } )
7642 }%

```



For MATHJAX, print the math between `\(` and `\)`:

```
7643 {%
7644     {\textbackslash(\LWR@HTMLsanitize{#2}\textbackslash)}%
7645 }% mathjax
```

For SVG, print the math inside a `lateximage`, with an `<alt>` tag of the  $\LaTeX$  code, and a CSS style to control the baseline adjustment.

```
7646 {% not mathjax
```

Measure the depth, width, and height of the math image:

```
7647     \LWR@restoreorigformatting%
7648     \LWR@orignormalsize%
7649     \global\setlength{\LWR@singledollardepth}{\depthof{%
7650         \LWR@origdollar#2\LWR@origdollar%
7651     }*\real{.8}}%
7652     \global\setlength{\LWR@singledollarwidth}{\widthof{%
7653         \LWR@origdollar#2\LWR@origdollar%
7654     }*\real{.8}}%
7655     \global\setlength{\LWR@singledollarheight}{\totalheightof{%
7656         \LWR@origdollar#2\LWR@origdollar%
7657     }*\real{.8}}%

7658     \LWR@origscriptsize%
```

Set a style for the the height or width. The `em` unit is used so that the math scales according to the user's selected font size. Start with the larger of width or height:

```
7659     \ifdimgreater{\LWR@singledollarwidth}{\LWR@singledollarheight}{%
7660         \def\LWR@singledollarstyle{%
7661             width:\LWR@convertto{em}{\the\LWR@singledollarwidth} em%
7662         }%
7663     }{%
7664         \def\LWR@singledollarstyle{%
7665             height:\LWR@convertto{em}{\the\LWR@singledollarheight }em%
7666         }%
7667     }%
```

If narrow width, use the height. Single-letter variables look best if they all are scaled according to height.

```
7668     \ifdimless{\LWR@singledollarwidth}{.8em}%
7669     {%
7670         \def\LWR@singledollarstyle{%
7671             height:\LWR@convertto{em}{\the\LWR@singledollarheight }em%
7672         }%
```

```
7673 }%
7674 {}%
```

If extremely wide and short, use the width:

```
7675 \ifdimless{\LWR@singledollarheight}{.3em}%
7676 {%
7677     \def\LWR@singledollarstyle{%
7678         width:\LWR@convertto{em}{\the\LWR@singledollarwidth} em%
7679     }%
7680 }%
7681 {}%
```

If there is significant text depth, add the depth to the style.

```
7682 \ifdimgreater{\LWR@singledollardepth}{0.01em}{%
7683     \def\LWR@singledollardepthstyle{%
7684         \ ; % extra space
7685         \LWR@origmbox{%
7686             vertical-align:-\LWR@convertto{em}{\the\LWR@singledollardepth} em%
7687         } % extra space
7688     }%
7689 }{%
7690     \def\LWR@singledollardepthstyle{%
7691     }%
```

Create the `lateximage` using the alternate tag and the computed size and depth.  
The star causes `lateximage` to use an MD5 hash as the filename.

```
7692 \ifblank{#1}{%
7693     \begin{lateximage}*%
7694     [%
7695         \textbackslash( %
7696         \LWR@HTMLsanitize{#2} % extra space
7697         \textbackslash)%
7698     ]%
7699     [\LWR@singledollarstyle \LWR@singledollardepthstyle]%
7700 }{%
7701     \begin{lateximage}%
7702     [#1]%
7703     [\LWR@singledollarstyle \LWR@singledollardepthstyle]%
7704 }%
7705 \LWR@origdollar#2\LWR@origdollar%
7706 \end{lateximage}%
7707 }%
7708 \endgroup%
7709 }
```

`\LWR@singledollar`  $\{\langle alt\ text\rangle\}\{\langle math\ expression\rangle\}$

```
7710 \protected\gdef\LWR@singledollar#1${%
7711 \LWR@subsingledollar}\{#1}%
7712 }
```

`\(` Redefine to the above dollar macros.

`\)`

```
7713 \protected\gdef\(#1\){$#1$}
7714 \protected\gdef\[#1\]{$$#1$$}
7715
7716 \endgroup
```

`\@ensuredmath`  $\{\langle expression\rangle\}$

```
7717 \LetLtxMacro\LWR@origensuredmath\@ensuredmath
7718
7719 \renewcommand{\@ensuredmath}[1]{%
7720 \ifbool{mathjax}{%
```

If MathJax, a `lateximage` is used, since `\ensuremath` is often used for complex  $\TeX$  expressions which MathJax may not render.

```
7721 % \begin{lateximage}[math image]
7722 % \LWR@origensuredmath{#1}
7723 % \end{lateximage}
7724 \LWR@subsingledollar{math image}\{\relax#1}%
7725 }\}% SVG math
```

If already inside a `lateximage` in math mode, continue as-is.

```
7726 \ifmmode%
7727 \LWR@origensuredmath{#1}%
7728 \else%
```

Create an inline math `lateximage`, but don't use the expression as the alt tag, since the expression may be complex internal code used to draw a math expression.

```
7729 \LWR@subsingledollar{math image}\{\relax#1}%
7730 \fi%
7731 }\}%
7732 }
```

Remove the old `math` and `displaymath` environments:

```
7733 \let\math\relax
```

```

7734 \let\endmath\relax
7735 \let\displaymath\relax
7736 \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.

```

7737 \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.

```

7738 \NewEnviron{displaymath}{\expandafter\[\BODY\]\@ignoretrue}

```

### 71.3 MATHJAX support

Ctrl **LWR@nextequation** Used to add one to compute the next equation number.

```

7739 \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  $\LaTeX$ .

```

7740 \newcommand*{\LWR@syncmathjax}{%

```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```

7741 \ifcsdef{thechapter}{
7742 \InlineClass{hidden}{
7743 \textbackslash(
7744 \textbackslash{seteqsection \thechapter\}
7745 \textbackslash)}
7746 }
7747 }
7748 {}% not using chapters

```

MATHJAX doesn't allow setting the equation number to 1:

```

7749 \ifthenelse{\cnttest{\value{equation}}>0}
7750 {

```

Tell MATHJAX that the next set of equations begins with the current  $\LaTeX$  equation number, plus one.

```
7751 \setcounter{LWR@nextequation}{\value{equation}}
7752 \addtocounter{LWR@nextequation}{1}
```

Place the MATHJAX command inside “\(\” and “\)” characters, to be printed to HTML, not interpreted by  $\LaTeX$ .

```
7753 \InlineClass{hidden}{
7754   \textbackslash(
7755   \textbackslash{seteqnumber \{\arabic{LWR@nextequation}\}}
7756   \textbackslash)
7757 }
7758 }{}% not eq > 0
7759 }
```

`\LWR@hidelatexequation`  $\{\langle environment \rangle\} \{\langle contents \rangle\}$

Creates the  $\LaTeX$  version of the equation inside an HTML comment.

```
7760 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```
7761 \LWR@stoppars
7762 \LWR@htmlopencomment
7763
```

Start the  $\LaTeX$  math environment inside the HTML comment:

```
7764 \begingroup
7765 \csuse{LWR@orig#1}
```

While in the math environment, restore various commands to their  $\LaTeX$  meanings.

```
7766 \LWR@restoreorigformatting
```

See `\LWR@htmlmathlabel` in section 71.5.1.

Print the contents of the equation:

```
7767 #2
```

End the  $\LaTeX$  math environment inside the HTML comment:

```
7768 \csuse{LWR@origend#1}
```

```
7769 \endgroup
7770
```

Close the HTML comment and resume HTML paragraph handling:

```
7771 \LWR@htmlclosecomment
7772 \LWR@startpars
7773 }
```

```
\LWR@addmathjax {<environment>} {<contents>}
```

Given the name of a math environment and its contents, create a MATHJAX instance. The contents are printed to HTML output, not interpreted by  $\text{\LaTeX}$ .

```
7774 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
```

Enclose the MATHJAX environment inside printed “\(" and “\)” characters.

```
7775 \LWR@origtilde\LWR@orignewline
7776 \textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
7777 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MATHJAX environment:

```
7778 \textbackslash{}end\{#1\}
7779 \LWR@orignewline
7780 }
```

## 71.4 Equation environment

Remember existing equation environment:

```
7781 \let\LWR@origequation\equation
7782 \let\LWR@origendequation\endequation
```

Remove existing equation environment:

```
7783 \let\equation\relax
7784 \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  $\TeX$  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.

```
7785 \NewEnviron{equation}{%
7786
```

If `mathjax` or `FormatWP`, print the  $\TeX$  expression:

```
7787 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{%
```

MATHJAX output:

```
7788 {
```

Print commands to synchronize MATHJAX's equation number and format to the current  $\TeX$  chapter/section and equation number:

```
7789 \LWR@syncmathjax
```

Print the  $\TeX$  math inside an HTML comment:

```
7790 \LWR@hidelatexequation{equation}{\BODY}
7791 }
```

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`.

```
7792 {% not mathjax
```

Begin the `lateximage` with an `<alt>` tag containing the math source:

```
7793 \begin{BlockClass}{displaymathnumbered}
7794 \begin{lateximage}[%
7795 (\LWR@equationtag) \textbackslash{begin\{equation\}} % extra space
7796 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}} % extra space
7797 \textbackslash{end\{equation\}}}%
7798 ]% alt tag
```

Create the actual  $\TeX$ -formatted equation inside the `lateximage` using the contents of the environment.

```

7799 \LWR@origequation
7800 \BODY% contents collected by NewEnviron
7801 \LWR@origendequation
7802 \end{lateximage}%
7803 \end{BlockClass}
7804 }% not mathjax
7805

```

After the environment, if MATHJAX, print the math to the HTML output for MATHJAX processing:

```

7806 }[%
7807 \ifbool{expr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }}%
7808 {%
7809 \LWR@addmathjax{equation}{\BODY}%
7810 }{}%
7811
7812 ]

```

Env **equation\***

```

7813 \csletcs{LWR@origequationstar}{equation*}
7814 \csletcs{LWR@origendequationstar}{endequation*}
7815 \renewenvironment*{equation*}
7816 {\displaymath}
7817 {\enddisplaymath}

```

## 71.5 AMS Math environments

### 71.5.1 Support macros

Bool **LWR@amsmultline** True if processing a multiline environment.

To compensate for multiline-specific code, **LWR@amsmultline** is used to add extra horizontal space in **\LWR@htmlmathlabel** if is used in an **amsmath** environment which is not a multiline environment and not an equation.

```

7818 \newbool{LWR@amsmultline}
7819 \boolfalse{LWR@amsmultline}

```

**\LWR@htmlmathlabel** **{⟨label⟩}**

**lwarp** points **\ltx@label** here. This is used by **\label** when inside a  $\TeX$  AMS math environment's math display environment.



`\LWR@origltx@label` points to the  $\LaTeX$  original, modified by lwarp, then by amsmath, then by cleveref.

```
7820 \newcommand*{\LWR@htmlmathlabel}[1]{%
7821 \LWR@traceinfo{\LWR@htmlmathlabelb #1}%
```

If mathjax or FormatWP, print the  $\LaTeX$  expression:

```
7822 \ifbool{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{%
7823 {%
```

The combined  $\LaTeX$  & HTML label is printed in a `\text` field:

```
7824 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
7825 \ifbool{\LWR@amsmultline}{\hspace*{\totwidth@}}
```

Temporarily end the HTML comment, insert the  $\LaTeX$  & HTML label, then resume the HTML comment. `\@firstofone` is required to remove extra braces introduced by the amsmath package.)

```
7826 \LWR@htmlclosecomment%
7827 \LWR@origltx@label{#1}%
7828 \LWR@htmlopencomment%
7829 }% text
7830 }% mathjax
7831 {%
7832 \LWR@origltx@label{#1}%
7833 }%
7834 }
```

`\LWR@beginhideamsmath` Starts hiding  $\LaTeX$  math inside an HTML comment.

```
7835 \newcommand*{\LWR@beginhideamsmath}{
7836 \LWR@stoppars
7837 \LWR@origtilde\LWR@orignewline
7838 \LWR@htmlopencomment
7839
7840 \begingroup
7841 \LWR@restoreorigformatting
7842 }
```

`\LWR@endhideamsmath` Ends hiding  $\LaTeX$  math inside an HTML comment.

```

7843 \newcommand*{\LWR@endhideamsmath}{
7844 \endgroup
7845
7846 \LWR@htmlclosecomment
7847 \LWR@orignewline
7848 \LWR@startpars
7849 }

```

### 71.5.2 Environment patches

The following amsmath environments already collect their contents in `\@envbody` for further processing. `eqnarray` is not an  $\mathcal{AMS}$  package, and thus requires special handling.

For `svg math`: Each environment is encapsulated inside a `lateximage` environment, along with a special 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  $\mathcal{AMS}$  environment.

For `MATHJAX`: Each environment is syched with  $\text{\LaTeX}$ 's equation numbers, typeset with  $\text{\LaTeX}$  inside an HTML comment, then printed to HTML output for `MATHJAX` to process.

Env `eqnarray` This environment is not an  $\mathcal{AMS}$  environment and thus its body is not automatically captured, so the `environ` package is used to capture the environment into `\BODY`.

```

7850 \let\LWR@origeqnarray\eqnarray
7851 \let\LWR@origendeqnarray\endeqnarray

```

To remember whether the starred environment was used, and thus whether to number the equations:

```

7852 \newbool{\LWR@numbereqnarray}
7853 \booltrue{\LWR@numbereqnarray}

```

Common code used by `eqnarray` and `Beqnarray` (from `fancybox`):

```

7854 \newcommand{\LWR@eqnarrayfactor}{%

```

If `mathjax` or `FormatWP`, print the  $\text{\LaTeX}$  expression:

```

7855 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7856 {%

```

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:

```
7857 \LWR@syncmathjax
7858 \boolfalse{LWR@amsmultline}
7859 \ifbool{LWR@numbreqnarray}
7860 {
```

If numbering the equations, execute a copy inside an HTML comment block:

```
7861 \LWR@beginhideamsmath
7862 \LWR@origeqnarray
7863 \BODY
7864 \LWR@origendeqnarray
7865 \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MATHJAX to interpret:

```
7866 \LWR@addmathjax{eqnarray}{\BODY}
7867 }%
7868 {% not LWR@numbreqnarray
```

If not numbering equations, just create the contents for MATHJAX:

```
7869 \LWR@addmathjax{eqnarray*}{\BODY}
7870 }% LWR@numbreqnarray
7871 }% mathjax
7872 {% not mathjax
7873 \ifbool{LWR@numbreqnarray}
7874 {
```

For numbered svg equations, first create a `lateximage` with an `alt` attribute containing sanitized copy of the source code:

```
7875 \begin{BlockClass}{displaymathnumbered}
7876 \begin{lateximage}[(\LWR@startingequationtag--\LWR@equationtag)
7877 \LWR@addmathjax{eqnarray}{\BODY}]
```

Then create the image contents using an actual `eqnarray`:

```
7878 \LWR@origeqnarray
7879 \BODY
7880 \LWR@origendeqnarray
7881 \end{lateximage}
7882 \end{BlockClass}
7883 }%
7884 {% not LWR@numbreqnarray
```

If not numbered, do the same, but an extra `\nonumber` seems to be required:

```

7885      \begin{BlockClass}{displaymath}
7886      \begin{lateximage}[\LWR@addmathjax{eqnarray*}]{\BODY}]
7887      \LWR@origeqnarray
7888      \BODY
7889      \nonumber
7890      \LWR@origendeqnarray
7891      \end{lateximage}
7892      \end{BlockClass}
7893  }% \LWR@numbereqnarray
7894 }% not mathjax

```

Default to number equations in the future:

```

7895 \booltrue{\LWR@numbereqnarray}
7896 }

```

`eqnarray` itself is made with a blank line before and after to force it to be on its own line:

```

7897 \RenewEnviron{eqnarray}
7898 {%
7899
7900 \LWR@eqnarrayfactor
7901
7902 }

```

The starred version is patched to turn off the numbering:

```

7903 \csgpreto{eqnarray*}{\boolfalse{\LWR@numbereqnarray}}

```

The following  $\mathcal{A}\mathcal{M}\mathcal{S}$  environments are more easily patched in-place:

Env `multline`

```

7904 \BeforeBeginEnvironment{multline}{
7905
7906 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7907 {
7908   \LWR@syncmathjax
7909   \booltrue{\LWR@amsmultline}
7910   \LWR@beginhideamsmath
7911 }
7912 {
7913   \begin{BlockClass}{displaymathnumbered}
7914   \begin{lateximage}[\LWR@amsmathbodynumbered{multline}]

```

```

7915 }
7916 }
7917
7918 \AfterEndEnvironment{multline}{
7919
7920 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7921 {
7922     \LWR@endhideamsmath
7923     \boolfalse{LWR@amsmultline}
7924     \LWR@addmathjax{multline}{\the\@envbody}
7925 }
7926 {\end{lateximage}\end{BlockClass}}
7927
7928 }

```

Env **multline\***

```

7929 \BeforeBeginEnvironment{multline*}{
7930
7931 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7932 {
7933     \LWR@syncmathjax
7934     \booltrue{LWR@amsmultline}
7935     \LWR@beginhideamsmath
7936 }
7937 {
7938     \begin{BlockClass}{displaymath}
7939     \begin{lateximage}[\LWR@amsmathbody{multline*}]
7940 }
7941 }
7942
7943 \AfterEndEnvironment{multline*}{
7944
7945 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7946 {
7947     \LWR@endhideamsmath
7948     \boolfalse{LWR@amsmultline}
7949     \LWR@addmathjax{multline*}{\the\@envbody}
7950 }
7951 {\end{lateximage}\end{BlockClass}}
7952
7953 }
7954

```

Env **gather**

```

7955 \BeforeBeginEnvironment{gather}{
7956

```

```

7957 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7958 {
7959     \LWR@syncmathjax
7960     \boolfalse{LWR@amsmultline}
7961     \LWR@beginhideamsmath
7962 }
7963 {
7964     \begin{BlockClass}{displaymathnumbered}
7965     \begin{lateximage}[\LWR@amsmathbodynumbered{gather}]
7966 }
7967 }
7968
7969 \AfterEndEnvironment{gather}{
7970
7971 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7972 {
7973     \LWR@endhideamsmath
7974     \LWR@addmathjax{gather}{\the\@envbody}
7975 }
7976 {\end{lateximage}\end{BlockClass}}
7977
7978 }

```

Env **gather\***

```

7979 \BeforeBeginEnvironment{gather*}{
7980
7981 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7982 {
7983     \LWR@syncmathjax
7984     \boolfalse{LWR@amsmultline}
7985     \LWR@beginhideamsmath
7986 }
7987 {
7988     \begin{BlockClass}{displaymath}
7989     \begin{lateximage}[\LWR@amsmathbody{gather*}]
7990 }
7991 }
7992
7993 \AfterEndEnvironment{gather*}{
7994
7995 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7996 {
7997     \LWR@endhideamsmath
7998     \LWR@addmathjax{gather*}{\the\@envbody}
7999 }
8000 {\end{lateximage}\end{BlockClass}}
8001
8002 }

```

Env `align`

```

8003 \BeforeBeginEnvironment{align}{
8004
8005 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8006 {
8007     \LWR@syncmathjax
8008     \boolfalse{LWR@amsmultline}
8009     \LWR@beginhideamsmath
8010 }
8011 {
8012     \begin{BlockClass}{displaymathnumbered}
8013     \begin{lateximage}[\LWR@amsmathbodynumbered{align}]
8014 }
8015 }
8016
8017 \AfterEndEnvironment{align}{
8018
8019 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8020 {
8021     \LWR@endhideamsmath
8022     \LWR@addmathjax{align}{\the\@envbody}
8023 }
8024 {\end{lateximage}\end{BlockClass}}
8025
8026 }
```

Env `align*`

```

8027 \BeforeBeginEnvironment{align*}{
8028
8029 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8030 {
8031     \LWR@syncmathjax
8032     \boolfalse{LWR@amsmultline}
8033     \LWR@beginhideamsmath
8034 }
8035 {
8036     \begin{BlockClass}{displaymath}
8037     \begin{lateximage}[\LWR@amsmathbody{align*}]
8038 }
8039 }
8040
8041 \AfterEndEnvironment{align*}{
8042
8043 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8044 {
8045     \LWR@endhideamsmath
8046     \LWR@addmathjax{align*}{\the\@envbody}
```

```

8047 }
8048 {\end{lateximage}\end{BlockClass}}
8049
8050 }

```

Env `flalign`

```

8051 \BeforeBeginEnvironment{flalign}{
8052
8053 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8054 {
8055     \LWR@syncmathjax
8056     \boolfalse{LWR@amsmultline}
8057     \LWR@beginhideamsmath
8058 }
8059 {
8060     \begin{BlockClass}{displaymathnumbered}
8061     \begin{lateximage}[\LWR@amsmathbodynumbered{flalign}]
8062 }
8063 }
8064
8065 \AfterEndEnvironment{flalign}{
8066
8067 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8068 {
8069     \LWR@endhideamsmath
8070     \LWR@addmathjax{flalign}{\the\@envbody}
8071 }
8072 {\end{lateximage}\end{BlockClass}}
8073
8074 }

```

Env `flalign*`

```

8075 \BeforeBeginEnvironment{flalign*}{
8076
8077 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8078 {
8079     \LWR@syncmathjax
8080     \boolfalse{LWR@amsmultline}
8081     \LWR@beginhideamsmath
8082 }
8083 {
8084     \begin{BlockClass}{displaymath}
8085     \begin{lateximage}[\LWR@amsmathbody{flalign*}]
8086 }
8087 }
8088

```



```

8089 \AfterEndEnvironment{flalign*}{
8090
8091 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8092 {
8093     \LWR@endhideamsmath
8094     \LWR@addmathjax{flalign*}{\the\@envbody}
8095 }
8096 {\end{lateximage}\end{BlockClass}}
8097
8098 }

8099 \end{warpHTML}

```

## 72 Lateximages

### 72.1 Description

**Env** `lateximage` A `lateximage` is a piece of the document which is typeset in  $\text{\LaTeX}$  then included in the HTML output as an image. This is used for math if `svg math` is chosen, and also for the `picture`, `tikzpicture`, and other environments.

Before typesetting the `lateximage` a large number of formatting, graphics, and symbols-related macros are temporarily restored to their print-mode meaning by `\LWR@restoreorigformatting`. (See section 70.)

A `lateximage` is typeset on its own PDF page inside an HTML comment which starts on the preceeding page and ends on following page, and instructions are written to `lateximage.txt` for `lwarpmk` to extract the `lateximage` from the page of the PDF file then generate an accompanying `.svg` file image file. Meanwhile, instructions to show this image are placed into the HTML file after the comment.

An HTML `<span>` is created to hold both the HTML comment, which will have the `pdftotext` conversion, and also the link to the final `.svg` image.

A  $\text{\LaTeX}$  label is used to remember which PDF page has the image. A label is used because footnotes, endnotes, and pagenotes may cause the image to appear at a later time. The label is declared along with the image, and so it correctly remembers where the image finally ended up.

**SVG image font size** The size of the math and text used in the `svg` image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, for ex:  
`\renewcommand{\LateximageFontSizeName}{large}`

## 72.2 Support counters and macros

**for HTML output:** 8100 `\begin{warpHTML}`

Ctr `LWR@lateximagenumber` Sequence the images.

```
8101 \newcounter{LWR@lateximagenumber}
8102 \setcounter{LWR@lateximagenumber}{0}
```

Ctr `LWR@lateximagedepth` Do not create `\lateximage` inside of `\lateximage`.

```
8103 \newcounter{LWR@lateximagedepth}
8104 \setcounter{LWR@lateximagedepth}{0}
```

A few utility macros to write special characters:

```
8105 \edef\LWR@hashmark{\string#} % for use in \write
8106 \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`.


```
8107 \newcounter{LWR@Lipage}
```

```
8108 \end{warpHTML}
```

## 72.3 Font size

**for HTML & PRINT:** 8109 `\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.*

```
8110 \newcommand*{\LateximageFontSizeName}{large}
```

```
8111 \end{warpall}
```

## 72.4 Sanitizing math expressions for HTML

**for HTML output:** 8112 `\begin{warpHTML}`

`\LWR@HTMLsanitize`  $\{\langle text \rangle\}$

Math expressions are converted to lateximages, and some math environments may contain “&”, “<”, or “>”, which should not be allowed inside an HTML <alt> tag, so must convert them to HTML entities.

Two versions follow, depending on expansion needs. There may be a better way...

```

8113 \newcommand{\LWR@HTMLsanitize}[1]{%
8114 \begingroup%
8115 \LWR@FBcancel%
8116 \protect\StrSubstitute{\detokenize{#1}}%
8117 {\detokenize{&}}%
8118 {\detokenize{&}}[\LWR@strresult]%
8119 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
8120 {\detokenize{<}}%
8121 {\detokenize{<}}[\LWR@strresult]%
8122 [\LWR@strresult]%
8123 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
8124 {\detokenize{>}}%
8125 {\detokenize{>}}[\LWR@strresult]%
8126 [\LWR@strresult]%
8127 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
8128 {\detokenize{##}}%
8129 {\#}%
8130 [\LWR@strresult]%
8131 \LWR@strresult%
8132 \endgroup%
8133 }

```

`\LWR@HTMLsanitizeexpand`  $\{\langle text \rangle\}$

This version expands the argument before sanitizing it.

```

8134 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
8135 \begingroup%
8136 \LWR@FBcancel%
8137 \protect\StrSubstitute{\detokenize\expandafter{#1}}%
8138 {\detokenize{&}}%
8139 {\detokenize{&}}[\LWR@strresult]%
8140 [\LWR@strresult]%
8141 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
8142 {\detokenize{<}}%
8143 {\detokenize{<}}[\LWR@strresult]%
8144 [\LWR@strresult]%
8145 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
8146 {\detokenize{>}}%
8147 {\detokenize{>}}[\LWR@strresult]%
8148 [\LWR@strresult]%

```

```

8149 \LWR@strresult%
8150 \endgroup%
8151 }

```

## 72.5 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.

```

8152 \newcounter{LWR@startingequation}
8153
8154 \@ifundefined{chapter}
8155 {
8156 \renewcommand{\theLWR@startingequation}{%
8157 \arabic{LWR@startingequation}%
8158 }
8159 }
8160 {% chapter defined
8161 \renewcommand{\theLWR@startingequation}{%
8162 \ifnumcomp{\value{chapter}}{>}{0}{\arabic{chapter}.}{}%
8163 \arabic{LWR@startingequation}%
8164 }
8165 }

```

**Bool** **LWR@isstartingequation** True for the first equation tag, false for later tags in the same environment.

```

8166 \newbool{LWR@isstartingequation}

```

**\LWR@startingequationtag** Prints the starting equation number or tag.

```

8167 \let\LWR@startingequationtag\theLWR@startingequation

```

**\LWR@equationtag** Prints the ending equation number or tag.

```

8168 \let\LWR@equationtag\theequation

```

Only if svg math, patch \tag after packages have loaded, in case someone else modified \tag.

```

8169 \AtBeginDocument{
8170
8171 \ifbool{mathjax}{\tag}{% not mathjax

```

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

For use inside the math environments while using svg math. Sets `\theLWR@startingequation` and `\theequation` to the given tag.

```

8172 \NewDocumentCommand{\LWR@remembertag}{m}{%
8173 \ifbool{LWR@isstartingequation}%
8174 {%
8175   \global\boolfalse{LWR@isstartingequation}%
8176   \xdef\LWR@startingequationtag{#1}%
8177 }%
8178 {}%
8179 \xdef\LWR@equationtag{#1}%
8180 }%
```

Patches for  $\mathcal{AMS}$  math `\tag` macro to remember the first tag:

```

8181 \LetLtxMacro\LWR@origmake@df@tag@@\make@df@tag@@
8182 \LetLtxMacro\LWR@origmake@df@tag@@@\make@df@tag@@@
8183
8184 \renewcommand*{\make@df@tag@@}[1]{%
8185 \LWR@remembertag{#1}%
8186 \LWR@origmake@df@tag@@{#1}%
8187 }
8188
8189 \renewcommand*{\make@df@tag@@@}[1]{%
8190 \LWR@remembertag{#1}%
8191 \LWR@origmake@df@tag@@@{#1}%
8192 }
8193
8194 }% not mathjax
8195 }% AtBeginDocument
```

## 72.6 HTML `<alt>` tags

`\LWR@amsmathbody`  $\{ \langle envname \rangle \}$  For use inside the optional argument to a `lateximage` to add the contents of a AMS math environment to the `<alt>` tag.

```

8196 \newcommand*{\LWR@amsmathbody}[1]
8197 {%
8198 \textbackslash\{begin\}\{#1\} % extra space
8199 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}%
8200 \textbackslash\{end\}\{#1\}%
8201 }
```

`\LWR@amsmathbodynumbered`  $\{ \langle envname \rangle \}$  For use inside the optional argument to a `lateximage` to add the contents of a AMS math environment to the `<alt>` tag, prefixed by the equation numbers.

```
8202 \newcommand*{\LWR@amsmathbodynumbered}[1]
8203 {%
8204 \ifnumcomp{\value{LWR@startingequation}}{=}{\value{equation}}%
8205 {(\LWR@equationtag)}%
8206 {(\LWR@startingequationtag--\LWR@equationtag)} % extra space
8207 \LWR@amsmathbody{#1} % extra space
8208 }
```

## 72.7 lateximage

Env `lateximage` \* [`<alt> tag`] [`<CSS style>`]

Typesets the contents and then renders the result as an svg file.

The optional `<alt>` tag is included in the HTML code for use with copy/paste.

The starred version also uses the `<alt>` tag to create an MD5 hash which is used as the image's filename. This is used for svg math expressions, and may also be of some use for the document creator. The use of the hash for the filename means that identical expressions will all share a single file, and the file must be converted to svg only a single time.

```
8209 \catcode'\$=\active%
8210
8211 \NewDocumentEnvironment{lateximage}{s O{image} O{}}
8212 {%
8213 \LWR@traceinfo{lateximage: starting on \jobname.pdf page \arabic{page}}%
```

Nested `lateximages` remain one large `lateximage`:

```
8214 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
```

If nesting inside an already-existing `lateximage`, simply record one more level:

```
8215 {%
8216 \addtocounter{LWR@lateximagedepth}{1}%
8217 }%
```

Otherwise, this is the outer-most `lateximage`:

```
8218 {% start of outer-most lateximage
```

Remember the next equation number to be allocated, in case it must be printed in a multi-equation environment:

```
8219 \setcounter{LWR@startingequation}{\value{equation}}%
8220 \addtocounter{LWR@startingequation}{1}%
8221 \booltrue{LWR@isstartingequation}%
8222 \let\LWR@startingequationtag\theLWR@startingequation%
8223 \let\LWR@equationtag\theequation%
```

Starting a new lateximage:

```
8224 \addtocounter{LWR@lateximagenumber}{1}%
8225 \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%
```

While inside a lateximage, locally do not use mathjax:

```
8226 \boolfalse{mathjax}%
```

Be sure that are doing a paragraph:

```
8227 \LWR@ensuredoingapar%
```

Next file:

```
8228 \addtocounter{LWR@externalfilecnt}{1}%
8229 \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%
```

Figure out what the next page number will be. `\setcounterpageref` assigns `LWR@LIpage` to the page number for the reference `LWRlateximageXXX`:

```
8230 \setcounterpageref{LWR@LIpage}{LWRlateximage\arabic{LWR@lateximagenumber}}%
8231 \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:

```
8232 \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
8233 class="lateximagesource"}%
```

Write instructions to the `lateximages.txt` file:

```
8234 \LWR@traceinfo{lateximage: about to write to lateximages.txt}%
8235 \IfBooleanTF{#1}% starred
8236 {% hash
8237 \LWR@traceinfo{lateximage: hash true}%
8238 \edef\LWR@hashedname{\LWR@mdfive{\detokenize\expandafter{#2}}}%
8239 \LWR@traceinfo{lateximage: hash is \LWR@hashedname}%
```

```

8240      \immediate\write\LWR@lateximagesfile{%
8241          |\arabic{LWR@LPage}|true|\LWR@hashedname|
8242      }%
8243  }% hash
8244  {% no hash
8245      \LWR@traceinfo{lateximage: hash false}%
8246      \immediate\write\LWR@lateximagesfile{%
8247          |\arabic{LWR@LPage}|false|lateximage-\arabic{LWR@externalfilecnt}|
8248      }%
8249  }% no hash

```

Place an open comment tag. This will hide any traces of the lateximage PDF page which were picked up by pdftotext.

```

8250  \LWR@traceinfo{lateximage: about to create open comment}%
8251  \LWR@htmlopencomment%

```

One level deeper:

```

8252  \addtocounter{LWR@lateximagedepth}{1}%

```

Start the new PDF page:

```

8253  \LWR@traceinfo{lateximage: about to create a new page}%
8254  \LWR@orignewpage%

```

Typeset the image in a “standard” width page and font size:

```

8255  \LWR@traceinfo{lateximage: about to create minipage}%
8256  \LWR@originipage{6in}%
8257  \csuse{LWR@orig\LateximageFontSizeName}%

```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for \hspace, etc. inside a lateximage.

```

8258  \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
8259  \LWR@restoreorigformatting%

```

Use full-page footnotes instead of minipage footnotes. These become HTML footnotes.

```

8260  \def\@mpfn{footnote}%
8261  \def\thempfn{\thefootnote}%
8262  \let\@footnotetext\LWR@footnotetext%

```

Create the LWRlateximage<number> label:

```

8263  \LWR@traceinfo{lateximage: about to create label}%

```



```

8264 \LWR@origlabel{LWRlateximage\arabic{LWR@lateximagenumber}}}%
8265 \LWR@traceinfo{lateximage: finished creating the label}%

```

Enable print-mode math functions:

```

8266 \LetLtxMacro$\LWR@origdollar%
8267 \catcode'\$=3% math shift
8268 \LetLtxMacro\(\LWR@origopenparen%
8269 \LetLtxMacro\)\LWR@origcloseparen%
8270 \LetLtxMacro\[ \LWR@origopenbracket%
8271 \LetLtxMacro\] \LWR@origclosebracket%
8272 }% end of outer-most lateximage
8273 \LWR@traceinfo{lateximage: finished start of environment}%
8274 }% end of \begin{lateximage}

```

`\endlateximage` When the environment closes:

```

8275 {% start of \end{lateximage}
8276 \LWR@traceinfo{lateximage: starting end of environment}%

```

Nested more than one deep?

```

8277 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%

```

If nesting inside an already-existing lateximage, simply record one more level:

```

8278 {\addtocounter{LWR@lateximagedepth}{-1}}%

```

If this is the outer-most lateximage:

```

8279 {% end of outer-most lateximage

```

Finish the lateximage minipage and start a new PDF page:

```

8280 \LWR@origendminipage%
8281 \LWR@orignewpage%
8282 \LWR@origscriptsize%

```

Close the HTML comment which encapsulated any traces of the lateximage picked up by pdftotext:

```

8283 \LWR@origvspace*{.5\baselineskip}%
8284 \LWR@htmlclosecomment%
8285 \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}%

```

Create a link to the lateximage, allowing its natural height:

```

8286 \IfBooleanTF{#1}% starred
8287 {% hash
8288 \LWR@subinlineimage[#2]{lateximage}%
8289 {%
8290 lateximages\OSPathSymbol%
8291 \LWR@origmbox{\LWR@hashedname}%
8292 }{svg}{#3}%
8293 }% hash
8294 {% no hash
8295 \LWR@subinlineimage[#2]{lateximage}%
8296 {%
8297 lateximages\OSPathSymbol%
8298 \LWR@origmbox{lateximage-\theLWR@externalfilecnt}%
8299 }{svg}{#3}%
8300 }% no hash

```

Be sure that are doing a paragraph:

```
8301 \LWR@ensuredoingapar%
```

Close the HTML span which has the pdftotext comment and also the link to the .svg image:

```

8302 \LWR@htmltag{/span}%
8303 \ifbool{HTMLDebugComments}{%
8304 \LWR@htmlcomment{End of lateximage}%
8305 }{}%
8306 % \LWR@orignewline% Removed to prevent extra space.

```

Undo one lateximage level:

```

8307 \addtocounter{LWR@lateximagedepth}{-1}%
8308 }% end of outer-most lateximage
8309 \LWR@traceinfo{lateximage: done}%
8310 }%
8311 \catcode'\$=3% math shift
8312 \end{warpHTML}

```

**for PRINT output:** 8313 \begin{warpprint}

Env lateximage [ $\langle alt \rangle tag$ ] [ $\langle CSS style \rangle$ ]

varwidth is used to create a box of the natural width of its contents.

```

8314 \NewDocumentEnvironment{lateximage}{s o o}
8315 {\begin{varwidth}[b]{\linewidth}}
8316 {\end{varwidth}}

```

```
8317 \end{warpprint}
```

## 73 center, flushleft, flushright

for HTML output: 8318 \begin{warpHTML}

Env center Replace center functionality with CSS tags:

```
8319 \renewenvironment*{center}
8320 {
8321 \LWR@forcenewpage
8322 \ifbool{FormatWP}
8323 {\BlockClass[\LWR@origmbox{text-align:center}]{center}}
8324 {\BlockClass{center}}
8325 }
8326 {\endBlockClass}
```

Env flushright

```
8327 \renewenvironment*{flushright}
8328 {
8329 \LWR@forcenewpage
8330 \ifbool{FormatWP}
8331 {\BlockClass[\LWR@origmbox{text-align:right}]{flushright}}
8332 {\BlockClass{flushright}}
8333 }
8334 {\endBlockClass}
```

Env flushleft

```
8335 \renewenvironment*{flushleft}
8336 {
8337 \LWR@forcenewpage
8338 \ifbool{FormatWP}
8339 {\BlockClass[\LWR@origmbox{text-align:left}]{flushleft}}
8340 {\BlockClass{flushleft}}
8341 }
8342 {\endBlockClass}
```

\leftline {\text{}}

```
8343 \renewcommand{\leftline}[1]{\begin{flushleft}#1\end{flushleft}}
```

```

\centerline  {\langle text\rangle}

8344 \renewcommand{\centerline}[1]{\begin{center}#1\end{center}}

\rightline  {\langle text\rangle}

8345 \renewcommand{\rightline}[1]{\begin{flushright}#1\end{flushright}}

8346 \end{warpHTML}

```

## 74 Pre-loaded packages

**for HTML output:** 8347 \begin{warpHTML}

If textcomp was loaded before lwarp, perhaps as part of the font-related packages, explicitly load the lwarp patches now:

```

8348 \@ifpackageloaded{textcomp}
8349 {
8350 \LWR@origRequirePackage{lwarp-textcomp}
8351 }
8352 {}

```

If graphics or graphicx were loaded before lwarp, perhaps by xunicode, explicitly load the lwarp patches now:

```

8353 \@ifpackageloaded{graphics}
8354 {
8355 \LWR@origRequirePackage{lwarp-graphics}
8356 }
8357 {}


8358 \end{warpHTML}

```

## 75 Siunitx

**Pkg** siunitx The lwarp core passes a few options to siunitx.

**fractions** Due to pdftolatex limitations, fraction output is replaced by symbol output for per-mode and quotient-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.

for HTML output: 8359 \begin{warpHTML}

Options for siunitx:

```

8360 \newrobustcmd{\LWR@siunitx@textcelsius}{\HTMLentity{deg}C}
8361 \newrobustcmd{\LWR@siunitx@textdegree}{\HTMLentity{deg}}
8362 \newrobustcmd{\LWR@siunitx@textprime}{\HTMLunicode{2032}}
8363 \newrobustcmd{\LWR@siunitx@textdblprime}{\HTMLunicode{2033}}
8364 \newrobustcmd{\LWR@siunitx@textplanckbar}{\text{\HTMLunicode{0127}}}
8365
8366 \appto\LWR@restoreorigformatting{
8367 \renewrobustcmd{\LWR@siunitx@textcelsius}{\text{\ensuremath{^\circ}C}}
8368 \renewrobustcmd{\LWR@siunitx@textdegree}{\text{\ensuremath{^\circ}}}
8369 \renewrobustcmd{\LWR@siunitx@textprime}{\text{\ensuremath{^\prime}}}
8370 \renewrobustcmd{\LWR@siunitx@textdblprime}{\text{\ensuremath{^\prime\prime}}}
8371 \renewrobustcmd{\LWR@siunitx@textplanckbar}{\text{\ensuremath{\hbar}}}
8372 }
8373
8374 \PassOptionsToPackage{
8375     detect-mode=true,
8376     per-mode=symbol,% fraction is not seen by pdftotext
8377     text-celsius = {\LWR@siunitx@textcelsius},
8378     text-degree = {\LWR@siunitx@textdegree},
8379     text-arcminute = {\LWR@siunitx@textprime} ,
8380     text-arcsecond = {\LWR@siunitx@textdblprime} ,
8381 }{siunitx}

8382 \end{warpHTML}

```

## 76 Graphics print-mode modifications

### 76.1 General limitations

- ⚠ .pdf image files For \includegraphics with .pdf files, the user should provide a .pdf image file, and also a .svg, .png, or .jpg version of the same image. **These should be referred to without a file extension:**
- ⚠ no file extension

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

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

Prog **pdftocairo** To convert a PDF image to SVG, use the utility `pdftocairo`:

Enter  $\Rightarrow$  `pdftocairo -svg filename.pdf`

If a `.pdf` file is 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

Pkg **epstopdf** For `.eps` files, use `epstopdf` to provide a PDF version, and also provide a SVG version as well.

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

## 76.2 Print-mode modifications

for PRINT output: For print output, accept and then discard the new class key:

```
8383 \begin{warpprint}
8384 \define@key{Gin}{class}{}

```

Print-mode additions for the overpic package. See section 245 for the HTML version.

```

8385 \AtBeginDocument{
8386 \ifpackageloaded{overpic}{
8387 \newcommand*{\overpicfontsize}{12}
8388 \newcommand*{\overpicfontskip}{14}
8389 }{}
8390 }
8391 \end{warpprint}

```

## 77 Xcolor boxes

**Pkg** **xcolor** A few new definitions are provided for enhanced HTML colored boxes, and `\fcolorbox` is slightly modified. Print-mode version are also provided.

Print-mode versions of new xcolor defintions. These are defined inside `warppall` because they are also used for HTML while inside a `lateximage`. They are defined `\AtBeginDocument` so that the xcolor originals may first be loaded and saved for reuse.

The framed versions are modified to allow a background color of none, in which case only the frame is drawn, allowing the background page color to show.

**for HTML & PRINT:** 8392 `\begin{warppall}`

After `xparse` may have been loaded ...

```

8393 \AtBeginDocument{

```

... and *only* if xcolor was loaded:

```

8394 \ifpackageloaded{xcolor}{
8395 \LWR@traceinfo{patching xcolor}

```

`\colorboxBlock` `\colorboxBlock` is the same as `\colorbox`:

```

8396 \LetLtxMacro\colorboxBlock\colorbox

```

In HTML mode, the following is done when xcolor is loaded. Following is the print-mode action:

```

8397 \warpprintonly{
8398 \LetLtxMacro\LWRprint@colorboxBlock\colorbox
8399 \LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox

```

```
8400 \LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox
8401 }
```

`\fcolorbox` [*framemodel*] [*framecolor*] [*boxmodel*] [*boxcolor*] [*text*]

In print mode, `\fcolorbox` is modified to accept a background color of none.

(`\fcolorbox` is particular about its optional arguments, thus the elaborate combinations of `\ifthenelse`.)

```
8402 \newsavebox{\LWR@colorminipagebox}
8403
8404 \DeclareDocumentCommand{\LWRprint@fcolorbox}{o m o m +m}{%
8405 \LWR@traceinfo{\LWRprint@fcolorbox #2 #4}%
```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
8406 \begin{lrbox}{\LWR@colorminipagebox}%
8407 #5%
8408 \end{lrbox}%
```

Sort out the various optional arguments and the background color of none. In each case, the LRbox is placed inside a `\fcolorbox`.

The current color is remembered, then set to the frame, then the current color is used for the contents.

```
8409 \ifthenelse{\equal{#4}{none}}{%
8410 {% #4 none
8411   \LWR@traceinfo{background is none}%
8412   {% scope the \colorlet
8413     \colorlet{\LWR@currentcolor}{.}%
8414     \color{#2}%
8415     \fbox{%
8416       \color{\LWR@currentcolor}%
8417       \usebox{\LWR@colorminipagebox}%
8418     }% fbox
8419   }% colorlet
8420 }% #4 none
8421 {% #4 not none
8422 \LWR@traceinfo{background not none}%
8423 \IfValueTF{#1}%
8424 {%
8425   \IfValueTF{#3}%
8426   {\LWRorigprint@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
8427   {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
8428 }%
8429 {% no value #1
8430   \IfValueTF{#3}%
```



```

8431    {\LWRorigprint@fcolorbox{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
8432    {\LWRorigprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
8433 }% no value #1
8434 }% #4 not none
8435 \LWR@traceinfo{\LWRprint@fcolorbox done}%
8436 }

```

`\fcolorboxBlock` [*framemodel*] [*framecolor*] [*boxmodel*] [*boxcolor*] [*text*]

In print mode, `\fcolorboxBlock` is the same as `\fcolorbox`.

```

8437 \LetLtxMacro\LWRprint@fcolorboxBlock\LWRprint@fcolorbox

```

Env `fcolorminipage` [*1:framemodel*] [*2:framecolor*] [*3:boxmodel*] [*4:boxcolor*] [*5:align*] [*6:height*] [*7:inner-align*] [*8:width*]

In print mode, becomes a `\fcolorbox` containing a minipage:

```

8438 \NewDocumentEnvironment{\LWRprint@fcolorminipage}{o m o m O{c} O{ } o m}
8439 {%
8440 \LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%

```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```

8441 \begin{lrbox}{\LWR@colorminipagebox}%

```

If inner alignment is not given, use the outer alignment instead:

```

8442 \IfValueTF{#7}%
8443 {\begin{minipage}[#5][#6][#7]{#8}}%
8444 {\begin{minipage}[#5][#6][#5]{#8}}%
8445 }%
8446 {%
8447 \end{minipage}%
8448 \end{lrbox}%
8449 \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`.

The current color is remembered, then set to the frame, then the current color is used for the contents.

```

8450 \ifthenelse{\equal{#4}{none}}%
8451 {% #4 none
8452   {\scope the \colorlet
8453     \colorlet{\LWR@currentcolor}{.}%

```

```

8454         \color{#2}%
8455         \fbox{%
8456             \color{LWR@currentcolor}%
8457             \usebox{\LWR@colorminipagebox}%
8458         }% fbox
8459     }% colorlet
8460 }% #4 none
8461 {% #4 not none
8462     \IfValueTF{#1}%
8463     {%
8464         \IfValueTF{#3}%
8465         {\LWRorigprint@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
8466         {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
8467     }%
8468     {% no value #1
8469         \IfValueTF{#3}%
8470         {\LWRorigprint@fcolorbox{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
8471         {\LWRorigprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
8472     }% no value #1
8473 }% #4 not none
8474 \LWR@traceinfo{*** finished end fcolorminipage}%
8475 }

```

`\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`.

```

8476 \newcommand*{\LWR@restoreorigprintxcolor}{%
8477 \LWR@traceinfo{\LWR@restoreorigprintxcolor}%
8478 \LetLtxMacro\colorboxBlock\LWRprint@colorboxBlock%
8479 \LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
8480 \LetLtxMacro\fcolorboxBlock\LWRprint@fcolorboxBlock%
8481 \LetLtxMacro\fcolorminipage\LWRprint@fcolorminipage%
8482 \LetLtxMacro\endfcolorminipage\endLWRprint@fcolorminipage%
8483 }
8484
8485 \appto{\LWR@restoreorigformatting}{%
8486 \LWR@restoreorigprintxcolor%
8487 }

```

If print mode, immediately activate the print-mode enhancements for xcolor:

```

8488 \warpprintonly{\LWR@restoreorigprintxcolor}
8489
8490 \LWR@traceinfo{xcolor patches done}
8491 }{% xcolor loaded
8492 }% AtBeginDocument

8493 \end{warppall}

```

## 78 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 section 78 to redefine the message which is printed for page number references.

loading order cleveref and the following associated macro patches are automatically preloaded at the end of the preamble via `\AtEndPreamble` and `\AfterEndPreamble`. This is done because the HTML conversion requires cleveref. The user’s document may not require cleveref, thus the user may never explicitly load it, so during HTML output lwarp loads it last. If the user’s document preamble uses cleveref options, or functions such as `\crefname`, then cleveref may be loaded in the user’s preamble near the end, and lwarp’s additional loading of cleveref will have no effect.

Table 10 on page 380 shows the data structure of the label/reference system as revised by lwarp and cleveref.

A few patches allow cleveref to work as-is:

for HTML output: 8494 `\begin{warpHTML}`

`\AtEndPreamble` forces cleveref to be loaded last:

```
8495 \AtEndPreamble{
8496 \RequirePackage{cleveref}
8497 }
```

The following patches are applied after cleveref has loaded, and after `\AtBeginDocument`. Print-mode versions are not required since they all come down to `\ref` eventually, and `\ref` has a print-mode version.

```
8498 \AfterEndPreamble{
8499 \LWR@traceinfo{Patching cleveref.}
```

```
\@@@setcref {\<kindofref>} {\<label>}
```

\@templabel becomes the section number.

```

8500 \def\LWR@orig@@@setcref#1#2{\cref@getlabel{#2}{\@templabel}#1{\@templabel}{-}{-}}%
8501
8502 \ifdefequal{\@@setcref}{\LWR@orig@@@setcref}{% before v0.21
8503   \renewcommand*{\@@setcref}[2]{#1{\ref{#2}}{-}{-}}
8504 }{
8505   \ifdefequal{\@@@setcref}{\LWR@orig@@@setcref}{% as of v0.21
8506     \renewcommand*{\@@@setcref}[2]{#1{\ref{#2}}{-}{-}}
8507   }{
8508     \PackageWarning{lwarp-cleveref}{
8509       Unknown verison of cleveref.
8510       \protect\cref\space will fail.
8511     }%
8512   }
8513 }
```

\@@@setcrefrange {<text>} {<label>} {<label>}

```

8514 \def\LWR@orig@@@setcrefrange#1#2#3{%
8515   \cref@getlabel{#2}{\@labela}%
8516   \cref@getlabel{#3}{\@labelb}%
8517   #1{\@labela}{\@labelb}{-}{-}{-}}%
8518
8519 \ifdefequal{\@@setcrefrange}{\LWR@orig@@@setcrefrange}{
8520   \renewcommand{\@@setcrefrange}[3]{%
8521     #1{\ref{#2}}{\ref{#3}}{-}{-}{-}}%
8522 }
8523 }{
8524   \ifdefequal{\@@@setcrefrange}{\LWR@orig@@@setcrefrange}{
8525     \renewcommand{\@@@setcrefrange}[3]{%
8526       #1{\ref{#2}}{\ref{#3}}{-}{-}{-}}%
8527   }
8528 }{
8529   \PackageWarning{lwarp-cleveref}{
8530     Unknown verison of cleveref.
8531     \protect\crefrange\space will fail.
8532   }
8533 }
8534 }
8535
```

\cpagerefFor Redefinable word between “page(s)” and the page numbers.

```

8536 \newcommand*{\cpagerefFor}{for}
```

`\@@@setcpageref {<typeofref>}{<label>}`, where `typeofref` is “page” or “pages”

```

8537 \def\LWR@orig@@setcpageref#1#2{% before v0.21
8538   \cref@getpageref{#2}{\@temppage}#1{\@temppage}{-}{-}%
8539
8540 \def\LWR@orig@@@setcpageref#1#2{% as of v0.21
8541   \cpageref@getlabel{#2}{\@temppage}#1{\@temppage}{-}{-}%
8542
8543 \ifdefequal{\@@setcpageref}{\LWR@orig@@setcpageref}{
8544   \renewcommand*{\@@setcpageref}[2]{%
8545     #1{\cpagerefFor\ \cref{#2}}{-}{-}%
8546   }
8547 }{
8548   \ifdefequal{\@@@setcpageref}{\LWR@orig@@@setcpageref}{
8549     \renewcommand*{\@@@setcpageref}[2]{%
8550       #1{\cpagerefFor\ \cref{#2}}{-}{-}%
8551     }
8552   }
8553   {
8554     \PackageWarning{lwarp-cleveref}{
8555       Unknown verison of cleveref.
8556       \protect\cpageref\space will fail.
8557     }
8558   }
8559 }

8560 \def\LWR@orig@@setcpagerefrange#1#2#3{% before v0.21
8561   \cref@getpageref{#2}{\@pagea}%
8562   \cref@getpageref{#3}{\@pageb}%
8563   #1{\@pagea}{\@pageb}{-}{-}{-}%
8564
8565 \def\LWR@orig@@@setcpagerefrange#1#2#3{% as of v0.21
8566   \cpageref@getlabel{#2}{\@pagea}%
8567   \cpageref@getlabel{#3}{\@pageb}%
8568   #1{\@pagea}{\@pageb}{-}{-}{-}%
8569
8570 \ifdefequal{\@@setcpagerefrange}{\LWR@orig@@setcpagerefrange}{
8571   \renewcommand*{\@@setcpagerefrange}[3]{%
8572     #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{-}{-}{-}%
8573   }
8574 }{
8575   \ifdefequal{\@@@setcpagerefrange}{\LWR@orig@@@setcpagerefrange}{
8576     \renewcommand*{\@@@setcpagerefrange}[3]{%
8577       #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{-}{-}{-}%
8578     }
8579   }
8580   {
8581     \PackageWarning{lwarp-cleveref}{
8582       Unknown verison of cleveref.

```

```

8583         \protect\cpagerefrange\space will fail.
8584     }
8585 }
8586 }
8587
8588 }% AfterEndPreamble

```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

```

8589 \LetLtxMacro\LWR@origlabel\label
8590 \RenewDocumentCommand{\label}{-}{\LWR@newlabel}
8591
8592 \LetLtxMacro\LWR@origref\ref
8593 \RenewDocumentCommand{\ref}{-}{\LWR@newref}%
8594
8595 \LetLtxMacro\LWR@origpageref\pageref
8596 \RenewDocumentCommand{\pageref}{-}{\LWR@newpageref}
8597
8598 \end{warpHTML}

```

## 79 Picture

Env `picture` The `picture` environment is enclosed inside a `\lateximage`.

for HTML output: 8599 `\begin{warpHTML}`

Env `picture`

```

8600 \BeforeBeginEnvironment{picture}{\begin{lateximage}}
8601
8602 \AfterEndEnvironment{picture}{\end{lateximage}}
8603
8603 \end{warpHTML}

```

## 80 Boxes and Minipages

A CSS flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

 **inline** A line of text with an inline minipage or parbox will have the minipage or parbox

placed onto its own line, because a paragraph is a block element and cannot be made inline-block.

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


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

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

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

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

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

 **text alignment** Nested minipages adopt their parent’s text alignment in HTML, whereas in regular  $\TeX$  PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

**for HTML output:** 8604 `\begin{warpHTML}`

## 80.1 Counters and lengths

**Ctr** `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
8605 \newcounter{LWR@minipagedepth}
8606 \setcounter{LWR@minipagedepth}{0}
```

**Len** `\WR@minipagewidth` Used to convert the width into printable units.

```
8607 \newlength{\WR@minipagewidth}
```

**Len** `\WR@minipageheight` Used to convert the height into printable units.

```
8608 \newlength{\WR@minipageheight}
```

## 80.2 Footnote handling

Also see section 51 for other forms of footnotes. Minipage footnotes are gathered in section 51.5, and then placed into the document in section 80.3.

## 80.3 Minipage handling

Bool LWR@minipagefullwidth Should the next minipage have no HTML width?

```
8609 \newbool{LWR@minipagefullwidth}
8610 \boolfalse{LWR@minipagefullwidth}
```

\minipagefullwidth Requests that the next minipage have no width tag in HTML:

**for HTML output:** 8611 \newcommand\*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}  
8612 \end{warppHTML}

**for PRINT output:** 8613 \begin{warpprint}  
8614 \newcommand\*{\minipagefullwidth}{}  
8615 \end{warpprint}

**for HTML output:** 8616 \begin{warppHTML}

Bool LWR@minipagethispar Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
8617 \newbool{LWR@minipagethispar}
8618 \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.

```
8619 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}
8620 {%
```

Temporarily open a group, in which width and height is computed based on a virtual page size instead of the extra-large PDF page used during HTML tag generation.

The following used to be an actual  $\text{\LaTeX}$  minipage.

```
8621 \begingroup
```



Compute width, adjusted for frames:

```
8622 \setlength{\LWR@minipagewidth}{#4}%
8623 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
```

Only create a new page if not yet nested:

```
8624 \LWR@orignewpage%
```

Adjust virtual page size:

```
8625 \addtolength{\LWR@minipagewidth}{3em}% room for frames
8626 \setlength{\linewidth}{6in}%
8627 \setlength{\textwidth}{6in}%
8628 \setlength{\textheight}{9in}%
8629 }{}%
8630 \LWR@traceinfo{computed width is \LWR@printlength{\LWR@minipagewidth}}%
```

Compute height:

```
8631 \setlength{\LWR@minipageheight}{\textheight}% default unless specified
8632 \IfValueT{#2}{\setlength{\LWR@minipageheight}{#2}}%
```

Track nesting depth:

```
8633 \addtocounter{LWR@minipagedepth}{1}%
```

$\LaTeX$  wants to start a paragraph for the virtual minipage, then start a paragraph again for the contents of the minipage, so cancel the paragraph tag handling until the minipage has begun.

```
8634 \ifbool{FormatWP}{\newline}{}%
8635 \LWR@stoppars%
```

If FormatWP, add a text frame:

```
8636 \ifbool{FormatWP}{%
8637
8638 \addtocounter{LWR@thisautoidWP}{1}%
8639 \LWR@htmltag{div id="\LWR@origmbox{autoidWP-\arabic{LWR@thisautoidWP}}" class="wpminipage"}%
8640
8641 }{}%
```

Create the <div> tag with optional alignment style:

```
8642 \LWR@traceinfo{minipage: creating div class}%
8643 \LWR@htmltag{div class="minipage" style="%
8644 \ifthenelse{\equal{#1}{t}}{\LWR@origmbox{vertical-align:bottom}} ; }{}%
```

```

8645 \ifthenelse{\equal{#1}{c}}{\LWR@origmbox{vertical-align:middle} ; }{}%
8646 \ifthenelse{\equal{#1}{b}}{\LWR@origmbox{vertical-align:top} ; }{}%
8647 \ifthenelse{\equal{#3}{t}}{\LWR@origmbox{justify-content:flex-start} ; }{}%
8648 \ifthenelse{\equal{#3}{c}}{\LWR@origmbox{justify-content:center} ; }{}%
8649 \ifthenelse{\equal{#3}{b}}{\LWR@origmbox{justify-content:flex-end} ; }{}%
8650 \ifthenelse{\equal{#3}{s}}{\LWR@origmbox{justify-content:space-between} ; }{}%

```

Print the width and optional height styles:

```

8651 \LWR@traceinfo{minipage: about to print the width of \LWR@printlength{\LWR@minipagewidth}}%
8652 \ifbool{\LWR@minipagefullwidth}%
8653 {\boolfalse{\LWR@minipagefullwidth}}%
8654 {%
8655     \ifthenelse{\lengthtest{#4}=\linewidth}%
8656     {}%
8657     {width:\LWR@printlength{\LWR@minipagewidth} ; }%
8658 }%
8659 \LWR@traceinfo{minipage: about to print the height}%
8660 \IfValueT{#2}{height:\LWR@printlength{\LWR@minipageheight} ; }%
8661 "%

```

Finish with an empty line to start the contents on a new line.

```

8662
8663 % The preceding empty line is required.

```

Set the user-accessible line and text width and height values inside the virtual minipage. These do not affect the actual size of the PDF output, but are used by any reference to `\linewidth`, etc. inside the virtual minipage being created here.

```

8664 \setlength{\linewidth}{#4}% the original width
8665 \setlength{\textwidth}{6in}%
8666 \setlength{\textheight}{9in}%

```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```

8667 \LWR@origraggedright%

```

Set minipage footnotes:

```

8668 \def\@mpfn{mpfootnote}%
8669 \def\thempfn{\thempfootnote}\c@mpfootnote\z@%
8670 \let\@footnotetext\@mpfootnotetext%

```

Resume paragraph tag handling for the contents of the minipage:

```

8671 \LWR@startpars%
8672 \ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%

```

```

8673
8674 === begin minipage ===
8675
8676 }{}%
8677 \LWR@traceinfo{minipage: finished starting the minipage}%
8678 }% finished \minipage
8679 {% \endminipage

```

Print pending minipage footnotes:

```

8680 \LWR@printpendingmpfootnotes%

```

End the environment with closing tag:

```

8681 \ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%
8682
8683 === end minipage ===
8684
8685 }{}%
8686 \LWR@stoppars%

```

The following used to be an actual  $\text{\LaTeX}$  minipage.

```

8687 \endgroup%
8688
8689 \ifbool{FormatWP}{%
8690
8691 \LWR@htmlelementend{div}%
8692
8693 }{}%
8694 \LWR@htmldivclassend{minipage}%
8695
8696 \addtocounter{LWR@minipagedepth}{-1}%
8697 \LWR@startpars%
8698 \ifbool{FormatWP}{\newline}{}%

```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```

8699 \global\booltrue{LWR@minipagethispar}%
8700 }

```

## 80.4 Parbox, mbox, makebox, framebox, fbox, raisebox

for HTML output:  
 $\backslash\text{parbox}$  [ $\langle pos \rangle$ ] [ $\langle height \rangle$ ] [ $\langle inner-pos \rangle$ ] { $\langle width \rangle$ } { $\langle text \rangle$ }

A parbox uses the minipage code:

```
8701 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}
8702 {
8703 \LWR@traceinfo{parbox of width #4}%
8704 \begin{minipage}[#1][#2][#3]{#4}%
8705 #5
8706 \end{minipage}%
8707 }
```

`\mbox`     $\langle text \rangle$       Nullified for HTML.

```
8708 \renewcommand*{\mbox}[1]{#1}
```

`\makebox`     $\langle \langle \rangle \rangle posn$      $[\langle width \rangle]$      $[\langle pos \rangle]$      $\langle text \rangle$

```
8709 \RenewDocumentCommand{\makebox}{d() o o m}{%
```

Check for the optional width:

```
8710 \IfValueTF{#2}%
8711 {%
```

Check for the horizontal text alignment. For stretched, the best HTML can do is justified alignment.

```
8712   {% scope
8713   \def\LWR@align{center}%
8714   \ifstrequal{#3}{l}{\def\LWR@align{left}}{}%
8715   \ifstrequal{#3}{r}{\def\LWR@align{right}}{}%
8716   \ifstrequal{#3}{s}{\def\LWR@align{justify}}{}%
```

To print the width argument:

```
8717   \setlength{\LWR@tempwidth}{#2}%
```

inline-block allows width and text-alignment to be used in a `<span>`.

```
8718   \InlineClass[%
8719     \LWR@origmbox{display:inline-block} ; %
8720     \LWR@origmbox{text-align}:\LWR@align\ ; %
8721     width:\LWR@printlength{\LWR@tempwidth}%
8722   ]%
8723   {makebox}%

8724   {#4}%
```

```
8725     }% scope
8726 }%
```

Without a width argument, the text is simply used inline:

```
8727 {#4}% no width
8728 }
```

`\framebox` [*<width>*] [*<pos>*] {*<text>*}

```
8729 \LetLtxMacro\LWR@origframebox\framebox
8730
8731 \RenewDocumentCommand{\framebox}{o o m}{%
8732   \fbox{\makebox[#1] [#2] {#3}}}%
8733 }
```

`\LWR@forceminwidth` {*<legth>*}

Sets `\LWR@atleastonept` to be at least 1pt.

```
8734 \newlength{\LWR@atleastonept}
8735
8736 \newcommand*{\LWR@forceminwidth}[1]{%
8737   \setlength{\LWR@atleastonept}{#1}%
8738   \ifthenelse{%
8739     \lengthtest{\LWR@atleastonept>0pt}\AND%
8740     \lengthtest{\LWR@atleastonept<1pt}}%
8741   }%
8742   {\setlength{\LWR@atleastonept}{1pt}}%
8743   {}%
8744 }
```

`\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.

```
8745 \newcommand*{\LWR@blackborderpadding}{%
8746   border:\LWR@printlength{\LWR@atleastonept} solid black ; %
8747   padding:\LWR@printlength{\fboxsep}%
8748 }
```

`\fbox` {*<text>*}

Creates a framed inline span enclosing the text.

Remember the print-mode version:

```
8749 \let\LWRprint@fbox\fbox
```

Create a new HTML version, but don't use it until after xcolor may have loaded:

```
8750 \newcommand{\LWRhtml@fbox}[1]{%
8751 \LWR@traceinfo{HTML fbox}%
8752 \LWR@forceminwidth{\fboxrule}%
8753 \InlineClass[%
8754 \LWR@blackborderpadding%
8755 ]{\fbox}{#1}
8756 }
```

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:

```
8757 \AtBeginDocument{\let\fbox\LWRhtml@fbox}
```

`\fboxBlock` `{\langle text \rangle}` Creates a framed HTML <div> of the text.

A print-output version is also supplied below.

```
8758 \newcommand{\fboxBlock}[1]{%
8759 \LWR@forceminwidth{\fboxrule}%
8760 \begin{BlockClass}[%
8761 \LWR@blackborderpadding%
8762 ]{\fboxBlock}
8763 #1
8764 \end{BlockClass}
8765 }
```

Env `fminipage` `[\langle align \rangle] [\langle height \rangle] [\langle align \rangle] {\langle width \rangle}`

Creates a framed HTML <div> around its contents.

A print-output version is also supplied below.

```
8766 \NewDocumentEnvironment{fminipage}{0{t} o 0{t} m}
8767 {%
8768 \LWR@traceinfo{fminipage #1 #2 #3 #4}%
8769 \LWR@forceminwidth{\fboxrule}%
8770 \setlength{\LWR@tempwidth}{#4}%
8771 \IfValueT{#2}{\setlength{\LWR@tempheight}{#2}}%
8772 \begin{BlockClass}[%
8773 \LWR@blackborderpadding ; %
8774 \IfValueT{#2}{height:\LWR@printlength{\LWR@tempheight} ; }%
8775 width:\LWR@printlength{\LWR@tempwidth}%
8776 ]{\fminipage}%
```

```

8777 }
8778 {%
8779 \end{BlockClass}%
8780 \LWR@traceinfo{fminipage done}%
8781 }

```

`\raisebox`  $\{ \langle raiselen \rangle \} [ \langle height \rangle ] [ \langle depth \rangle ] \{ \langle text \rangle \}$

```

8782 \LetLtxMacro{\LWR@origraisebox}{\raisebox}
8783
8784 \RenewDocumentCommand{\raisebox}{m o o m}{%
8785 #4%
8786 }

```

```

8787 \end{warpHTML}

```

**for HTML & PRINT:** 8788 \begin{warpall}

`LWRprint@fminipage` is defined inside `warpall`. For print output, it is `\let` to `fminipage`. For HTML output, the HTML version of `fminipage` is used instead, but the print version is still available for use inside a `lateximage`.

Env `LWRprint@fminipage`  $[ \langle 1:align \rangle ] [ \langle 2:height \rangle ] [ \langle 3:inner-align \rangle ] \{ \langle 4:width \rangle \}$

Creates a frame around its contents.

```

8789 \newsavebox{\LWR@fminipagebox}
8790
8791 \NewDocumentEnvironment{LWRprint@fminipage}{0{t} o 0{t} m}
8792 {%

```

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:

```

8793 \IfValueTF{#3}%
8794 {\def\LWR@thisalign{#3}}
8795 {\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.

```

8796 \IfValueTF{#2}%
8797 {\minipage[#1][#2+2\fboxsep+2\fboxrule][\LWR@thisalign]{#4+2\fboxsep+2\fboxrule}}%
8798 {\minipage[#1]{#4+2\fboxsep+2\fboxrule}}%

```

Capture the contents of the environment:

```
8799 \begin{lrbox}{\LWR@fminipagebox}%
```

Nest the contents inside an inner minipage of the desired size:

```
8800 \IfValueTF{#2}%
8801 {\minipage[#1][#2][\LWR@thisalign]{#4}}%
8802 {\minipage[#1]{#4}}%
8803 }
8804 {%
```

Close the inner minipage and the LR box with the contents:

```
8805 \endminipage%
8806 \end{lrbox}%
```

Create a frame around the contents of the environment:

```
8807 \fbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```
8808 \endminipage%
8809 }
```

```
8810 \end{warppall}
```

**for PRINT output:** 8811 \begin{warpprint}

For print output, the following are \let to become active.

```
\fboxBlock {\langle text \rangle}
```

Creates a framed HTML <div> around the text.

```
8812 \let\fboxBlock\fbox
```

Env fminipage [\langle align \rangle] [\langle height \rangle] [\langle align \rangle] {\langle width \rangle}

Creates a frame around its contents.

```
8813 \LetLtxMacro{fminipage}{\LWRprint@fminipage}
8814 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}

8815 \end{warpprint}
```



## 81 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 45.8.

**for HTML output:** 8816 \begin{warpHTML}

\LWR@HTMLtextstyle {<FormatWP style>} {<class>} {<text>}

If FormatWP, adds an explicit style to the text span class. This is used by LIBREOFFICE to mark its imported text using the given style.

```
8817 \DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{%
8818 \ifbool{FormatWP}%
8819 {\LWR@htmlspanclass[#1]{#2}{#3}}%
8820 {\LWR@htmlspanclass{#2}{#3}}%
8821 }
```

\emph {<text>}

```
8822 \DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{#1}}
8823 \DeclareRobustCommand{\LWR@nullemph}[1]{#1}
8824 \LetLtxMacro{\emph}{\LWR@HTMLemph}
```

\textmd {<text>}

```
8825 \DeclareRobustCommand{\LWR@HTMLtextmd}[1]{%
8826 \LWR@HTMLtextstyle{font-weight:normal}{textmd}{#1}%
8827 }
8828 \DeclareRobustCommand{\LWR@nulltextmd}[1]{#1}
8829
8830 \LetLtxMacro{\textmd}{\LWR@HTMLtextmd}
```

\textbf {<text>}

```
8831 \DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{#1}}
8832 \DeclareRobustCommand{\LWR@nulltextbf}[1]{#1}
8833 \LetLtxMacro{\textbf}{\LWR@HTMLtextbf}
```

`\textrm {<text>}`

```
8834 \DeclareRobustCommand{\LWR@HTMLtextrm}[1]{%
8835 \LWR@HTMLtextstyle{font-family:serif}{textrm}{#1}%
8836 }
8837
8838 \DeclareRobustCommand{\LWR@nulltextrm}[1]{#1}
8839
8840 \LetLtxMacro{\textrm}{\LWR@HTMLtextrm}
```

`\textsf {<text>}`

```
8841 \DeclareRobustCommand{\LWR@HTMLtextsf}[1]{%
8842 \LWR@HTMLtextstyle{font-family:sans}{textsf}{#1}%
8843 }
8844 \DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
8845 \LetLtxMacro{\textsf}{\LWR@HTMLtextsf}
```

`\texttt {<text>}`

```
8846 \DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
8847 \DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
8848 \LetLtxMacro{\texttt}{\LWR@HTMLtexttt}
```

`\textup {<text>}`

```
8849 \DeclareRobustCommand{\LWR@HTMLtextup}[1]{%
8850 \LWR@HTMLtextstyle{font-variant:normal}{textup}{#1}%
8851 }
8852
8853 \DeclareRobustCommand{\LWR@nulltextup}[1]{#1}
8854
8855 \LetLtxMacro{\textup}{\LWR@HTMLtextup}
```

`\textit {<text>}`

```
8856 \DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}}
8857 \DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
8858 \LetLtxMacro{\textit}{\LWR@HTMLtextit}
```

`\textsc {<text>}`

```
8859 \DeclareRobustCommand{\LWR@HTMLtextsc}[1]{%
8860 \LWR@HTMLtextstyle{font-variant:small-caps}{textsc}{#1}%
8861 }
8862
```

```

8863 \DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}
8864
8865 \LetLtxMacro{\textsc}{\LWR@HTMLtextsc}

```

`\textsl`     $\{\langle text \rangle\}$

```

8866 \DeclareRobustCommand{\LWR@HTMLtextsl}[1]{%
8867 \LWR@HTMLtextstyle{font-style:oblique}{textsl}{#1}%
8868 }
8869
8870 \DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}
8871
8872 \LetLtxMacro{\textsl}{\LWR@HTMLtextsl}

```

`\textnormal`     $\{\langle text \rangle\}$

```

8873 \DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
8874 \DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
8875 \LetLtxMacro{\textnormal}{\LWR@HTMLtextnormal}

8876 \DeclareRobustCommand{\LWR@nullrmfamily}{ }
8877 \DeclareRobustCommand{\LWR@nullsfamily}{ }
8878 \DeclareRobustCommand{\LWR@nullttfamily}{ }
8879 \DeclareRobustCommand{\LWR@nullbfseries}{ }
8880 \DeclareRobustCommand{\LWR@nullmdseries}{ }
8881 \DeclareRobustCommand{\LWR@nullupshape}{ }
8882 \DeclareRobustCommand{\LWR@nullslshape}{ }
8883 \DeclareRobustCommand{\LWR@nullscshape}{ }
8884 \DeclareRobustCommand{\LWR@nullitshape}{ }
8885 \DeclareRobustCommand{\LWR@nullem}[1]{ }
8886 \DeclareRobustCommand{\LWR@nullnormalfont}{ }

```

`\LWR@nullfonts`    Removes formatting during filename operations.



**Use only inside a group.**

The following are *not* made robust, since they must be expanded to their nullified versions.

```

8887 \newcommand*{\LWR@nullfonts}{%
8888 \LetLtxMacro{\emph}{\LWR@nullemph}%
8889 \LetLtxMacro{\textmd}{\LWR@nulltextmd}%
8890 \LetLtxMacro{\textbf}{\LWR@nulltextbf}%
8891 \LetLtxMacro{\textrm}{\LWR@nulltextrm}%
8892 \LetLtxMacro{\textsf}{\LWR@nulltextsf}%
8893 \LetLtxMacro{\texttt}{\LWR@nulltexttt}%
8894 \LetLtxMacro{\textup}{\LWR@nulltextup}%

```

```

8895 \LetLtxMacro{\textit}{\LWR@nulltextit}%
8896 \LetLtxMacro{\textsc}{\LWR@nulltextsc}%
8897 \LetLtxMacro{\textsl}{\LWR@nulltextsl}%
8898 \LetLtxMacro{\textnormal}{\LWR@nulltextnormal}%
8899 \LetLtxMacro{\rmfamily}{\LWR@nullrmfamily}%
8900 \LetLtxMacro{\sffamily}{\LWR@nullsffamily}%
8901 \LetLtxMacro{\ttfamily}{\LWR@nullttfamily}%
8902 \LetLtxMacro{\bfseries}{\LWR@nullbfseries}%
8903 \LetLtxMacro{\mdseries}{\LWR@nullmdseries}%
8904 \LetLtxMacro{\upshape}{\LWR@nullupshape}%
8905 \LetLtxMacro{\slshape}{\LWR@nullslshape}%
8906 \LetLtxMacro{\scshape}{\LWR@nullscshape}%
8907 \LetLtxMacro{\itshape}{\LWR@nullitshape}%
8908 \LetLtxMacro{\em}{\LWR@nullem}%
8909 \LetLtxMacro{\normalfont}{\LWR@nullnormalfont}%

```

```

8910 \renewcommand*{\,}{-}%
8911 \renewcommand*{~}{-}%
8912 \renewcommand*{\newline}{ }%
8913 \renewcommand*{\textellipsis}{-}%

8914 \renewcommand*{\HTMLunicode}[1]{-}%
8915 \renewcommand*{\HTMLentity}[1]{-}%

```

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

```

8916 \renewcommand*{\&}{and}%

8917 \renewcommand{\textsuperscript}[1]{##1}%
8918 \renewcommand{\textsubscript}[1]{##1}%

8919 \renewcommand{\underline}[1]{##1}%

8920 \RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
8921 \DeclareExpandableDocumentCommand{\InlineClass}{+o +m +m}{##3}%
8922 \DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{##3}%

```

Nullify math macros.

```

8923 \def\({##1}\){}%
8924 \def\[##1\]{}%
8925 \renewcommand{\LWR@subsingledollar}[2]{}
8926 }

```

Remembers the current font family, series, and shape.

```

8927 \newcommand*{\LWR@f@family}{rm}
8928 \newcommand*{\LWR@f@series}{md}
8929 \newcommand*{\LWR@f@shape}{up}

```

`\LWR@textcurrentfont`    `{\langle text \rangle}`

Prints the text with the current font choices.

```

8930 \newcommand*{\LWR@textcurrentfont}[1]{%
8931 \csuse{text\LWR@f@family}{%
8932 \csuse{text\LWR@f@series}{%
8933 \csuse{text\LWR@f@shape}{%
8934 #1%
8935 }%
8936 }%
8937 }%
8938 }

```

`\mdseries`

```

8939 \renewrobustcmd*{\mdseries}{\renewcommand*{\LWR@f@series}{md}}

```

`\bfseries`

```

8940 \renewrobustcmd*{\bfseries}{\renewcommand*{\LWR@f@series}{bf}}

```

`\rmfamily`

```

8941 \renewrobustcmd*{\rmfamily}{\renewcommand*{\LWR@f@family}{rm}}

```

`\sffamily`

```

8942 \renewrobustcmd*{\sffamily}{\renewcommand*{\LWR@f@family}{sf}}

```

`\ttfamily`

```

8943 \renewrobustcmd*{\ttfamily}{\renewcommand*{\LWR@f@family}{tt}}

```

`\upshape`

```

8944 \renewrobustcmd*{\upshape}{\renewcommand*{\LWR@f@shape}{up}}

```

`\itshape`

```
8945 \renewrobustcmd*{\itshape}{\renewcommand*{\LWR@f@shape}{it}}
```

`\scshape`

```
8946 \renewrobustcmd*{\scshape}{\renewcommand*{\LWR@f@shape}{sc}}
```

`\normalfont`

```
8947 \renewrobustcmd*{\normalfont}{\rmfamily\mdseries\upshape}
```

`\sp`  $\langle text \rangle$

For siunitx. Must work in math mode.

```
8948 \renewcommand{\sp}[1]{\text{<sup>#1</sup>}}
```

`\sb`  $\langle text \rangle$

For siunitx. Must work in math mode.

```
8949 \renewcommand{\sb}[1]{\text{<sub>#1</sub>}}
```

`\textsuperscript`  $\langle text \rangle$

```
8950 \renewrobustcmd{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

`\@textsuperscript`  $\langle text \rangle$

```
8951 \renewcommand{\@textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

`\textsubscript`  $\langle text \rangle$

```
8952 \AtBeginDocument{
8953 \renewrobustcmd{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
8954 }
```

`\@textsubscript`  $\langle text \rangle$

```
8955 \AtBeginDocument{
8956 \renewcommand{\@textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
8957 }
```

`\up`  $\langle text \rangle$  Prints superscript.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
8958 \AtBeginDocument{\let\up\textsuperscript}
```

`\fup`  $\langle text \rangle$  Prints superscript.

Supports `fmtcount` package.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
8959 \AtBeginDocument{\let\fup\textsuperscript}
```

`\underline`  $\langle text \rangle$

```
8960 \renewcommand{\underline}[1]{%
8961 \LWR@HTMLtextstyle%
8962   {text-decoration:underline;text-decoration-skip}%
8963   {\underline}{#1}%
8964 }
```

`\LWR@overline`  $\langle text \rangle$

```
8965 \newcommand{\LWR@overline}[1]{%
8966 \LWR@HTMLtextstyle%
8967   {text-decoration:overline;text-decoration-skip}%
8968   {\overline}{#1}%
8969 }
```

```
8970 \end{warpHTML}
```

## 82 Skips, spaces, font sizes

**for HTML output:** 8971 `\begin{warpHTML}`

`\,` must be redefined after `\RequirePackage{printlen}`

Direct-formatting space commands become HTML entities:

```
8972 \renewrobustcmd*{\,}{\HTMLunicode{202f}} % HTML thin non-breakable space
```

```

8973
8974 \renewrobustcmd*{~}{\HTMLentity{nbsp}}
8975
8976 \renewrobustcmd*{\textellipsis}{\HTMLunicode{2026}}

```

Direct-formatting font sizes are ignored:

```

8977 \renewrobustcmd*{\normalsize}{}
8978 \renewrobustcmd*{\small}{}
8979 \renewrobustcmd*{\footnotesize}{}
8980 \renewrobustcmd*{\scriptsize}{}
8981 \renewrobustcmd*{\tiny}{}
8982 \renewrobustcmd*{\large}{}
8983 \renewrobustcmd*{\Large}{}
8984 \renewrobustcmd*{\LARGE}{}
8985 \renewrobustcmd*{\huge}{}
8986 \renewrobustcmd*{\Huge}{}

8987 \DeclareDocumentCommand{\onecolumn}{}{}
8988
8989 \DeclareDocumentCommand{\twocolumn}{0{}}{
8990
8991 #1
8992
8993 }

```

`\hfill`

```

8994 \renewcommand*{\hfill}{\quad}

```

`\hrulefill`

```

8995 \renewcommand*{\hrulefill}{\rule{1in}{1pt}}

```

`\dotfill`

```

8996 \renewcommand*{\dotfill}{\dots}

```

`\newpage`

```

8997 \renewcommand*{\newpage}{
8998
8999 }

```

`\newline` Uses the HTML `<br />` element.



```
9000 \newrobustcmd*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
9001 \LetLtxMacro\newline\LWR@newlinebr
```

`\\` Redefined to `\LWR@endoffline` or `\LWR@tabularendoffline`.

`\LWR@endoffline` \* [*len*]

`\\` is assigned to `\LWR@endoffline` at `\LWR@LwarpStart`.

Inside tabular, `\\` is temporarily changed to `\LWR@tabularendoffline`.

```
9002 \LetLtxMacro\LWR@origendoffline\\
9003 \NewDocumentCommand{\LWR@endoffline}{s o}
9004 {%
9005 \newline%
9006 }
```

`\LWR@minipagestartpars` Minipages are often placed side-by-side inside figures, with a bit of horizontal space to separate them. Since HTML does not allow a `<div>` to be inside a `p`, paragraphs must be turned off during the generation of the minipage, then turned on after the minipage is complete. When this occurs between side-by-side minipages, `lwarp` correctly suppresses the paragraph tags between the minipages, unless some other text is between the minipages. Such text forms its own paragraph, resulting in text after a minipage to be on its own line. Since people often place small horizontal space between minipages, it is desirable to maintain this space if possible. `lwarp` tries to do this by remembering that a minipage has been seen, in which case paragraph tags are suppressed around `\hspace`, `\enskip`, `\quad`, and `\qquad` until the end of the paragraph, when the closing `p` tag is created.

`\hspace`  
`\enskip`  
`\quad`  
`\qquad`

When a minipage is seen, the boolean `LWR@minipagethispar` is set, telling the following horizontal whitespace commands to try to suppress their surrounding paragraph tags. `LWR@minipagethispar` is cleared at the next end of paragraph, when the HTML paragraph closing tag is generated.

Placed just before `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
9007 \newcommand*{\LWR@minipagestartpars}{%
9008 \ifbool{LWR@minipagethispar}{\LWR@startpars}{}%
9009 }
```

`\LWR@minipagestoppars` Placed just after `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
9010 \newcommand*{\LWR@minipagestoppars}{%
9011 \ifbool{LWR@minipagethispar}{\LWR@stoppars}{}%
9012 }
```

`\quad` Handles special minipage & horizontal space interactions.

```
9013 \renewcommand*{\quad}{%
9014 \LWR@minipagestoppars%
9015 \HTMLUnicode{2001}%
9016 \LWR@minipagestartpars%
9017 }
```

`\qquad` Handles special minipage & horizontal space interactions.

```
9018 \renewcommand*{\qquad}{\quad\quad}
```

`\enskip` Handles special minipage & horizontal space interactions.

```
9019 \renewcommand*{\enskip}{%
9020 \LWR@minipagestoppars%
9021 \HTMLUnicode{2000}%
9022 \LWR@minipagestartpars%
9023 }
```

Len `\WR@tempwidth` Used to compute span width, height, raise for `\hspace` and `\rule`:

Len `\WR@tempheight`

Len `\WR@tempraise`

```
9024 \newlength{\LWR@tempwidth}
9025 \newlength{\LWR@tempheight}
9026 \newlength{\LWR@tempraise}
```

`\LWR@hspace` \*  $\{ \langle length \rangle \}$

Handles special minipage & horizontal space interactions.

Prints a span of a given width. Ignores the optional star.

`\hspace{\fill}` is converted to `\hspace{2em}`, equal to `\qquad`.

```
9027 \NewDocumentCommand{\LWR@hspace}{s m}{%
9028 \setlength{\LWR@tempwidth}{#2}%
```

If `\fill`, change to `\qquad`:

```
9029 \ifnum\gluestretchorder\LWR@tempwidth>0%
9030 \setlength{\LWR@tempwidth}{2em}%
9031 \fi%
```

Only if the width is not zero:

```
9032 \ifdimcomp{\LWR@tempwidth}{=}{0pt}{\}{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
9033 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
9034 \ifdimcomp{\LWR@tempwidth}{=}{.16667em}%
9035 {%
9036 \HTMLUnicode{2009}% thin breakable space
9037 }%
```

Print the span with the converted width. Not rounded.

```
9038 {%
9039 \LWR@htmltagc{%
9040 span style="width:\LWR@printlength{\LWR@tempwidth}; % extra space
9041 display:inline-block"%
9042 }%
```

If formatting for a word processor, approximate with a number of \quads, in case a span of a given width is not supported:

```
9043 \ifbool{FormatWP}{%
9044 \setlength{\LWR@templengthone}{\LWR@tempwidth}%
9045 \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
9046 \quad%
9047 \addtolength{\LWR@templengthone}{-1em}%
9048 }%
9049 }{}%
```

Close the span:

```
9050 \LWR@htmltagc{/span}%
9051 }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
9052 \LWR@minipagestartpars%
9053 }% width not 0
9054 }
```

`\LWR@nohspace * {<length>}`

Used to disable \hspace while creating description \items.

```
9055 \NewDocumentCommand{\LWR@nohspace}{s m}{}
```

`\hspace` \*  $\langle length \rangle$

Handles special minipage & horizontal space interactions.

```
9056 \LetLtxMacro{\hspace}{\LWR@hspace}
```

`\LWR@vspace` \*  $\langle length \rangle$  Nullified vspace.

```
9057 \NewDocumentCommand{\LWR@vspace}{s m}{} 
```

`\vspace` \*  $\langle length \rangle$  Nullified.

```
9058 \LetLtxMacro\vspace\LWR@vspace
```

`\linebreak` [ $\langle num \rangle$ ] Inserts an HTML br tag.

```
9059 \renewcommand*{\linebreak}[1] [] {\newline}
```

`\nolinebreak` [ $\langle num \rangle$ ]

```
9060 \renewcommand*{\nolinebreak}[1] [] {}
```

`\pagebreak` [ $\langle num \rangle$ ] Starts a new paragraph.

```
9061 \renewcommand*{\pagebreak}[1] [] {
```

```
9062
```

```
9063 }
```

`\nopagebreak` [ $\langle num \rangle$ ]

```
9064 \renewcommand*{\nopagebreak}[1] [] {}
```

`\enlargethispage` \*  $\langle len \rangle$

```
9065 \RenewDocumentCommand{\enlargethispage}{s m}{} 
```

`\clearpage`

`\cleardoublepage`

```
9066 \renewcommand*{\clearpage}{} 
```

```
9067 \renewcommand*{\cleardoublepage}{} 
```

`\LWR@currenttextcolor` The color to use for text and `\rule`, defaulting to black:

```
9068 \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.

```
9069 \NewDocumentCommand{\LWR@rule}{o m m}{%
```

The width is copied into a temporary  $\text{\TeX}$  length, from which comparisons and conversions may be made:

```
9070 \setlength{\LWR@tempwidth}{#2}%
```

If it's zero-width then skip the entire rule:

```
9071 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}
9072 {}% zero- width
9073 {}% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
9074 \ifthenelse{%
9075 \lengthtest{\LWR@tempwidth>0pt}\AND%
9076 \lengthtest{\LWR@tempwidth<1pt}}%
9077 }%
9078 {\setlength{\LWR@tempwidth}{1pt}}{}%
```

Likewise with height:

```
9079 \setlength{\LWR@tempheight}{#3}%
9080 \ifthenelse{%
9081 \lengthtest{\LWR@tempheight>0pt}\AND%
9082 \lengthtest{\LWR@tempheight<1pt}}%
9083 }%
9084 {\setlength{\LWR@tempheight}{1pt}}{}%
```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
9085 \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.

```
9086 \LWR@htmltagc{%
```

```

9087     span
9088     style="%"

```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```

9089     \ifbool{FormatWP}{\background:\LWR@currenttextcolor ; }%

```

The width and height are printed, converted to PT:

```

9090     width:\LWR@printlength{\LWR@tempwidth} ; %
9091     height:\LWR@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  $\TeX$  length allows a typical  $\TeX$  expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a  $\TeX$  length limits the allowable syntax. To do: A superior method would compute a ratio of  $\TeX$  ex height, then print that to HTML with an ex unit.

```

9092     \IfValueT{#1}%
9093     {%
9094         \setlength{\LWR@tempraise}{Opt-#1}%
9095         \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
9096         \LWR@orignewline%
9097         -ms-transform: translate(0pt,\LWR@printlength{\LWR@tempraise}); %
9098         \LWR@orignewline%
9099         -webkit-transform: translate(0pt,\LWR@printlength{\LWR@tempraise}); %
9100         \LWR@orignewline%
9101         transform: translate(0pt,\LWR@printlength{\LWR@tempraise}); %
9102         \LWR@orignewline%
9103     }%

```

Display inline-block to place the span inline with the text:

```

9104     display:inline-block;"%
9105     }%

```

If formatting for a word processor, approximate with a number of underscores, in case a span of a given width is not supported:

```

9106     \ifbool{FormatWP}{%
9107         \setlength{\LWR@templengthone}{\LWR@tempwidth}%
9108         \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
9109             \_}%
9110             \addtolength{\LWR@templengthone}{-1em}%
9111         }%
9112     }{}%

```

Close the span:

```
9113 \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
9114 \LWR@minipagestartpars%
9115 }% non-zero width
9116 }
```

```
\rule [⟨raise⟩] {⟨width⟩} {⟨height⟩}
```

Handles special minipage & horizontal space interactions.

```
9117 \renewrobustcmd{\rule}{\LWR@rule}
```

```
9118 \end{warpHTML}
```

## 83 \phantomsection

**for HTML output:** 9119 \begin{warpHTML}

`\phantomsection` Emulate the hyperref `\phantomsection` command, often used to insert the bibliography into table of contents:

```
9120 \DeclareDocumentCommand{\phantomsection}{}{%
9121 \section*{}}%
9122 }
```

```
9123 \end{warpHTML}
```

## 84 \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>

## 84.1 HTML logos

for HTML output: 9124 `\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  $\TeX$ ,  $\text{Lua}\TeX$ ,  $\text{Con}\TeX$ t, etc.

```
9125 \let\LWR@origTeX\TeX
9126
9127 \newcommand*{\LWR@TeX}
9128 {%
9129     \InlineClass{latexlogofont}%
9130     {%
9131         \LWR@HTMLtextstyle%
9132         {text-transform:uppercase}%
9133         {latexlogo}%
9134         {T\textsubscript{e}X}%
9135     }%
9136 }
```

`\LaTeX`  $\TeX$ ,  $\text{TeX} 2_{\epsilon}$

`\LaTeXe`

```
9137 \let\LWR@origLaTeX\LaTeX
9138
9139 \newcommand*{\LWR@LaTeX}
9140 {%
9141     \InlineClass{latexlogofont}%
9142     {%
9143         \LWR@HTMLtextstyle%
9144         {text-transform:uppercase}%
9145         {latexlogo}%
9146         {L\textsuperscript{a}T\textsubscript{e}X}%
9147     }%
9148 }
9149
9150 \let\LWR@origLaTeXe\LaTeXe
9151
9152 \renewcommand*{\LaTeXe}
```



```

9153 {\LaTeX\InlineClass{latexlogofont}}%
9154 {\,2\textsubscript{\textit{\HTMLunicode{3B5}}}}

```

`\LuaTeX`     $\text{Lua}\text{T}_{\text{E}}\text{X}$ ,  $\text{Lua}\text{M}\text{T}_{\text{E}}\text{X}$

`\LuaLaTeX`

```

9155 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
9156 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}

```

`\XeTeX`     $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$ ,  $\text{X}_{\text{E}}\text{M}\text{T}_{\text{E}}\text{X}$

`\XeLaTeX`

`xetexlogo` is a css class which aligns the backwards E in  $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$  and spaces  $\text{T}_{\text{E}}\text{X}$  appropriately.

`xelatexlogo` is a css class which aligns the backwards E in  $\text{X}_{\text{E}}\text{M}\text{T}_{\text{E}}\text{X}$  and spaces  $\text{M}\text{T}_{\text{E}}\text{X}$  appropriately.

```

9157 \newcommand*{\Xe}
9158   {\X\textsubscript{\HTMLunicode{18e}}}
9159 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
9160 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}

```

`\ConTeXt`     $\text{Con}\text{T}_{\text{E}}\text{Xt}$

```

9161 \newcommand*{\LWR@ConTeXt}
9162 {\InlineClass{latexlogofont}{Con}\TeX{}}%
9163 \InlineClass{latexlogofont}{t}}

```

`\BibTeX`     $\text{Bib}\text{T}_{\text{E}}\text{X}$ , *MakeIndex*

`\MakeIndex`

```

9164 \providecommand*{\BibTeX}
9165 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}
9166
9167 \newcommand*{\MakeIndex}
9168 {\InlineClass{latexlogofont}{\textit{MakeIndex}}}

```

`\AmS`     $\mathcal{A}\mathcal{M}\mathcal{S}$

`amslogo` is a css class used for the  $\mathcal{A}\mathcal{M}\mathcal{S}$ logo.

```

9169 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}}
9170 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}

```

`\MiKTeX`     $\text{MiK}\text{T}_{\text{E}}\text{X}$

```

9171 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}

```

`\LyX` `LyX`

`lyxlogo` is a CSS class used for the LyX logo.

```
9172 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
9173 \end{warpHTML}
```

## 84.2 Print logos

**for PRINT output:**

```
9174 \begin{warpprint}
9175 \newcommand*{\XeTeXrevE}
9176   {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
9177 \providecommand*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
9178 \providecommand*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
9179 \providecommand*{\AmS}{\%}
9180 \leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex\%}
9181 \hbox{$\mathcal M$}\kern-.2em\mathcal S$}}
9182 \newcommand*{\LyX}{\textsf{LyX}}
9183 \providecommand*{\LuaTeX}{\mbox{Lua\TeX}}
9184 \providecommand*{\LuaLaTeX}{\mbox{Lua\LaTeX}}
9185 \providecommand*{\BibTeX}{\mbox{B\textsc{ib}\TeX}}
9186 \providecommand*{\MakeIndex}{\mbox{\textit{MakeIndex}}}
9187 \providecommand*{\ConTeXt}{\mbox{Con\TeXt}}
9188 \providecommand*{\MiKTeX}{\mbox{MiK\TeX}}
9189 \end{warpprint}
```

## 85 `\AtBeginDocument`, `\AtEndDocument`

**for HTML output:** `9190 \begin{warpHTML}`

`\LWR@LwarpStart` Automatically sets up the HTML-related actions for the start and end of the document.

`\LWR@LwarpEnd`

```
9191 \AfterEndPreamble{\LWR@LwarpStart}
9192 \AtEndDocument{\LWR@LwarpEnd}
9193 \end{warpHTML}
```

## 86 Koma-script

Load patches to koma-script.

**for HTML output:** 9194 \begin{warpHTML}

```
9195 \@ifclassloaded{scrbook}{\RequirePackage{lwarp-patch-komascript}}{}  
9196 \@ifclassloaded{scrartcl}{\RequirePackage{lwarp-patch-komascript}}{}  
9197 \@ifclassloaded{scrreprt}{\RequirePackage{lwarp-patch-komascript}}{}  
  
9198 \end{warpHTML}
```

## 87 Memoir

Load patches to memoir.

**for HTML output:** 9199 \begin{warpHTML}

```
9200 \@ifclassloaded{memoir}{\RequirePackage{lwarp-patch-memoir}}{}  
  
9201 \end{warpHTML}
```

## 88 Trademarks

- $\text{\TeX}$  is a trademark of American Mathematical Society.
- ADOBE® and ADOBE Framemaker® are either registered trademarks or trademarks of ADOBE SYSTEMS INCORPORATED in the United States and/or other countries.
- LINUX® is the registered trademark of Linus Torvalds in the U.S. and other countries.
- MAC OS® is a trademark of APPLE INC.
- MADCAP FLARE™ is the property of MADCAP SOFTWARE, INC.
- MATHJAX is copyright 2009 and later. The MATHJAX CONSORTIUM is a joint venture of the AMERICAN MATHEMATICAL SOCIETY (AMS) and the SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS (SIAM) to advance mathematical and scientific content on the web.
- MICROSOFT®, ENCARTA, MSN, and WINDOWS® are either registered trademarks or trademarks of MICROSOFT CORPORATION in the United States and/or other countries.
- UNIX® is a registered trademark of THE OPEN GROUP.

The following adjustments apply to the lwarp-\* package listings:

---

**File 2** `lwarp-a4.sty`**§ 89**    **Package**    **a4**

**Pkg**    `a4`    `a4` is ignored.

**for HTML output:**    `1 \LWR@ProvidesPackageDrop{a4}`  
  
                          `2 \newcommand*{\WideMargins}{}`

---

**File 3** `lwarp-a4wide.sty`**§ 90**    **Package**    **a4wide**

**Pkg**    `a4wide`    `a4wide` is ignored.

**for HTML output:**    `1 \LWR@ProvidesPackageDrop{a4wide}`

---

**File 4** `lwarp-a5comb.sty`**§ 91**    **Package**    **a5comb**

**Pkg**    `a5comb`    `a5comb` is ignored.


**for HTML output:**    `1 \LWR@ProvidesPackageDrop{a5comb}`

---

**File 5** `lwarp-abstract.sty`**§ 92**    **Package**    **abstract**

*(Emulates or patches code by PETER WILSON.)*

**Pkg**    `abstract`    `abstract` is supported and patched by lwarp.

 **missing TOC**    If using the number option with file splits, be sure to place the table of contents before the abstract. The number option causes a section break which may cause a

file split, which would put a table of contents out of the home page if it is after the abstract.

for HTML output:

memoir provides an abstract environment even though it is not an article or report class. Meanwhile, lwarp loads book to emulate memoir, but book does not have an abstract environment, so when the abstract package is loaded for emulation there is no pre-existing abstract to redefine, which would cause an error. Thus, a null abstract is provide here:

```
1 \ProvideDocumentEnvironment{abstract}{}{}{}
```

Accept all options for lwarp-abstract:

```
2 \LWR@ProvidesPackagePass{abstract}

3 \AtBeginDocument{
4 \BeforeBeginEnvironment{abstract}{
5 \LWR@forcenewpage
6 \BlockClass{abstract}
7 }
8 \AfterEndEnvironment{abstract}{\endBlockClass}
9 }
10
11 \renewcommand{\@bsrunintitle}{%
12 \hspace*{\abstitleskip}%
13 {\abstractnamefont%
14 \InlineClass{abstractrunintitle}{\abstractname}%
15 \@bslabeldelim}%
16 }
17
18 \@ifclassloaded{memoir}
19 {
20 \renewenvironment{abstract}{%
21 % \titlepage
22 \null\vfil
23 \@beginparpenalty\@lowpenalty
24 \if@bsrunin
25 \else
26 \if@bsstyle
27 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
28 \else
29 \ifnumber@bs
30 \num@bs
31 \else
32 \begin{\absnamepos}%
33 \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
34 \endparpenalty\@M
35 \end\absnamepos%
36 %% \vspace{\abstitleskip}%
```

---

```

37      \fi
38      \fi
39      \vspace{\abstitleskip}%
40      \fi
41      \put@bsintoc%
42      \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
43      {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
44      }
45 }{% not memoir
46 \if@titlepage
47   \renewenvironment{abstract}{%
48 %     \titlepage
49     \null\vfil
50     \@beginparpenalty\@lowpenalty
51     \if@bsrunin
52     \else
53       \if@bsstyle
54         \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
55       \else
56         \ifnumber@bs
57           \num@bs
58         \else
59           \begin{\absnamepos}%
60 \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
61           \endparpenalty\@M
62           \end\absnamepos%
63 %%       \vspace{\abstitleskip}%
64       \fi
65     \fi
66     \vspace{\abstitleskip}%
67     \fi
68     \put@bsintoc%
69     \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
70     {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
71     }
72 \else
73   \renewenvironment{abstract}{%
74     \if@bsrunin
75     \else
76       \if@bsstyle
77         \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
78       \else
79         \ifnumber@bs
80           \num@bs
81         \else
82 \begin{\absnamepos}%
83 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
84 \end\absnamepos%
85 %%       \vspace{\abstitleskip}%
86     \fi

```

---

```

87      \fi
88      \vspace{\abstitlekip}%
89      \fi
90      \put@bsintoc%
91      \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
92      {\par\end{@bstr@ctlist}}
93 \fi
94 }% not memoir

```

---

## File 6 lwarp-acro.sty

### § 93 Package **acro**

(Emulates or patches code by CLEMENS NIEDERBERGER.)

Pkg acro acro is patched for use by lwarp.

for HTML output: 1 \LWR@ProvidesPackagePass{acro}

\DeclareAcronym is used in the preamble, where lwarp has not yet made the dollar active, so temporarily enable lwarp math catcode just for this definition:

```

2 \ExplSyntaxOn
3 \NewDocumentCommand \LWR@DeclareAcronym {mm}
4 {
5   \acro_declare_acronym:nn {#1} {#2}
6   \catcode'\$=3% lwarp
7 }
8 \ExplSyntaxOff
9
10 \RenewDocumentCommand{\DeclareAcronym}{}{}{
11   \catcode'\$=\active% lwarp
12   \LWR@DeclareAcronym
13 }

```

Modified to activate the current font:

```

14 \ExplSyntaxOn
15 \cs_gset_protected:Npn \acro_write_short:nn #1#2
16 {
17   \mode_if_horizontal:F { \leavevmode }
18   \group_begin:
19     \bool_if:NTF \l__acro_custom_format_bool
20       { \l__acro_custom_format_tl }
21       { \l__acro_short_format_tl }
22   {\LWR@textcurrentfont{#2}}% lwarp

```



---

```

23   \group_end:
24 }
25
26 \cs_gset_protected:Npn \acro_write_alt:nn #1#2
27 {
28   \mode_if_horizontal:F { \leavevmode }
29   \group_begin:
30     \bool_if:NTF \l__acro_custom_format_bool
31       { \l__acro_custom_format_tl }
32       { \l__acro_alt_format_tl }
33     {\LWR@textcurrentfont{#2}}% lwarp
34   \group_end:
35 }
36
37 \cs_gset_protected:Npn \acro_write_long:nn #1#2
38 {
39   \mode_if_horizontal:F { \leavevmode }
40   \group_begin:
41     \bool_if:NTF \l__acro_custom_long_format_bool
42       { \l__acro_custom_long_format_tl }
43       { \use:n }
44     {
45       \use:x
46       {
47         \exp_not:n {#1}
48         {
49           \bool_if:NTF \l__acro_first_upper_bool
50             { \exp_not:N \l__acro_first_upper_case:n { \exp_not:n {
51               \LWR@textcurrentfont{#2}}% lwarp
52             } } }
53             { \exp_not:n {\LWR@textcurrentfont{#2}} }% lwarp
54           }
55         }
56       }
57     \group_end:
58   }
59 \ExplSyntaxOff

```

---

File 7 **lwarp-acronym.sty**

§ 94 Package **acronym**

*(Emulates or patches code by TOBIAS OETIKER.)*

Pkg acronym acronym is patched for use by lwarp.

 \acresetall does not work with cleveref, causing multiply-defined labels. lwarp

patches acronym for HTML, but not for print mode.

for HTML output: `1 \LWR@ProvidesPackagePass{acronym}`

Uses `\textit` instead of `\itshape`:

```
2 \renewcommand{\acfia}[1]{%
3   {\textit{\AC@acl{#1}}}\ifAC@starred\acs*{#1}\else\acs{#1}\fi}}
```

Removes the mbox to allow math inside:

```
4 \renewcommand*{\AC@acs}[1]{%
5   % \mbox{
6   \expandafter\AC@get\csname fn@#1\endcsname\@firstoftwo{#1}}
7 % }
```

Modified for cleveref and zref:

```
8 \renewcommand*{\AC@und@newl@bel}[3]{%
9   \@ifundefined{#1@#3}%
10  {%
11    \global\expandafter\let\csname#2@#3\endcsname\@nnil
12    \global\expandafter\let\csname#2@#3@cref\endcsname\@nnil% lwarp
13  }%
14  {%
15    \global\expandafter\let\csname#1@#3\endcsname\relax
16    \global\expandafter\let\csname#1@#3@cref\endcsname\relax% lwarp
17    \global\expandafter\let\csname Z@R@#3\endcsname\relax% lwarp
18  }%
19 }%
```

Modified for cleveref and zref:

```
20 \renewcommand*{\AC@testdef}[3]{%
21 \ifstrequal{#1}{Z@R}{}{% lwarp
22   \@ifundefined{s@#2}\@secondoftwo\@firstofone
23   {%
24     \expandafter\ifx\csname s@#2\endcsname\empty
25     \expandafter\@firstofone
26   \else
27     \expandafter\xdef\csname s@#2\endcsname{%
28       \expandafter\expandafter
29       \expandafter\@gobble
30       \csname s@#2\endcsname
31     }%
32     \expandafter\@gobble
33   \fi
34 }%
35 {%
```

---

```

36   \@testdef{#1}{#2}{#3}%
37   }%
38 }% lwarp
39 }%

```

---

## File 8 `lwarp-adjmulticol.sty`

### \$ 95    Package **adjmulticol**

*(Emulates or patches code by BORIS VEYTSMAN.)*

Pkg `adjmulticol`    `adjmulticol` is emulated.

Emulation similar to `multicols` is used, with adjusted margins. If the number of columns is specified as 1, it is set so, but if two or greater are used, `lwarp` allows a variable number of columns up to three.

**for HTML output:**    `1 \LWR@ProvidesPackageDrop{adjmulticol}`

`2 \RequirePackage{multicol}`

`adjmulticols`    `* {\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@mcolstype}{multicols}%
10 \ifnumcomp{#2}{=}{1}{\renewcommand*{\LWR@mcolstype}{singlecolumn}}{}%

```

Help avoid page overflow:

```

11 \LWR@forcenewpage%

```

Create the `<div>` with the given margin and class:

```

12 \BlockClass[%

```

---

```

13 \LWR@origmbox{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
14 \LWR@origmbox{margin-right:\LWR@printlength{\LWR@templengthtwo}}%
15 ]{\LWR@mcolstype}%
16 }
17 {\endBlockClass}

```

---

## File 9 **lwarp-addlines.sty**

### \$ 96    Package **addlines**

*(Emulates or patches code by WILL ROBERTSON.)*

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 10 **lwarp-afterpage.sty**

### \$ 97    Package **afterpage**

*(Emulates or patches code by DAVID CARLISLE.)*

Pkg    **afterpage**    Emulated.

**for HTML output:**    Discard all options for lwarp-afterpage:

```

1 \LWR@ProvidesPackageDrop{afterpage}

2 \newcommand{\afterpage}[1]{#1}

```

---

## File 11 **lwarp-algorithmicx.sty**

### \$ 98    Package **algorithmicx**

*(Emulates or patches code by SZÁSZ JÁNOS.)*

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 319.1.

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 \LWR@htmltagc{%
10 span style="width:\LWR@printlength{\ALG@thistlm}; display:inline-block;"%
11 }%
12 \ifbool{FormatWP}{%
13 \setlength{\LWR@templengthone}{\the\ALG@thistlm}%
14 \whileof{\lengthtest{\LWR@templengthone>1em}}{%
15 \quad%
16 \addtolength{\LWR@templengthone}{-1em}%
17 }%
18 }{}%
19 \LWR@htmltagc{/span}%
20 }%
21
22 \let\LWR@origComment\Comment%
23
24 \renewcommand{\Comment}[1]{%
25   \InlineClass{floatright}{\LWR@origComment{#1}}%
26 }%
27 }
28
29 \renewcommand\algorithmiccomment[1]{%
30 \hfill\HTMLUnicode{25B7} #1% white right triangle
31 }%

32 \end{warpHTML}
```

File 12 **lwarp-alltt.sty**

§ 99 Package **alltt**

(Emulates or patches code by JOHANNES BRAAMS.)

Pkg alltt alltt is patched for use by lwarp.

```
for HTML output: 1 \LWR@ProvidesPackagePass{alltt}

2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching alltt.}
4 \AtBeginEnvironment{alltt}{%
5 \LWR@forcenewpage
6 \LWR@atbeginverbatim{alltt}\unskip\LWR@origvspace*{-\baselineskip}%
7 }
8 \AfterEndEnvironment{alltt}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
9 }
```

File 13 lwarp-amsthm.sty

§ 100 Package amsthm

(Emulates or patches code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)

The original source code is located in amsclass.dtx, and printed in amsclass.pdf.

Pkg amsthm amsthm is patched for use by lwarp.

Table 12: AMSthm package — CSS styling of theorems and proofs

**Theorem:** <div> of class amsthmbody<theoremstyle>

**Theorem Name:** <span> of class amsthmname<theoremstyle>

**Theorem Number:** <span> of class amsthmnumber<theoremstyle>

**Theorem Note:** <span> of class amsthmnote<theoremstyle>

**Proof:** <div> of class amsthmproof

**Proof Name:** <span> of class amsthmproofname

where <theoremstyle> is plain, definition, etc.

```
for HTML output: 1 \LWR@ProvidesPackagePass{amsthm}

Storage for the style being used for new theorems:

2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```

3 \renewcommand{\theoremstyle}[1]{%
4   \ifundefined{th@#1}{%
5     \PackageWarning{amsthm}{Unknown theoremstyle '#1'}%
6     \thm@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% lwarp
8   }{%
9     \thm@style{#1}%
10    \renewcommand{\LWR@newtheoremstyle}{#1}% lwarp
11  }%
12 }

```

Patched to remember the style for this theorem type:

```

13 \def\@xnthm#1#2{%
14   \csedef{LWR@thmstyle#2}{\LWR@newtheoremstyle}% lwarp
15   \let\@tempa\relax
16   \exp\@ifdefinable\csname #2\endcsname{%
17     \global\exp\let\csname end#2\endcsname\@endtheorem
18     \ifx *#1% unnumbered, need to get one more mandatory arg
19       \edef\@tempa##1{%
20         \gdef\@xp\@nx\csname#2\endcsname{%
21           \@nx\@thm{\@xp\@nx\csname th@\the\thm@style\endcsname}%
22           }{##1}}}%
23   \else % numbered theorem, need to check for optional arg
24     \def\@tempa{\@oparg{\@ynthm{#2}}{}}%
25   \fi
26   \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#2}}}% lwarp
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   \item[
47 %   \deferred@thm@head{
48 %     \the\thm@headfont \thm@indent
49   \@ifempty{#1}{\let\thmname\@gobble}{\LWR@haveamsthmname}% lwarp
50   \@ifempty{#2}{\let\thmnumber\@gobble}{\LWR@haveamsthmnumber}% lwarp
51   \@ifempty{#3}{\let\thmnote\@gobble}{\LWR@haveamsthmnote}% lwarp
52   \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
53   \the\thm@headpunct~
54   \thmheadnl % possibly a newline.
55   \hskip\thm@headsep
56 %   }%
57   ]
58   \ignorespaces}
```

Patched for CSS:

```
59 \def\@thm#1#2#3{%
60   \ifhmode\unskip\unskip\par\fi
61   \normalfont
62   \LWR@forcenewpage% lwarp
63   \BlockClass{amsthmbody\LWR@thisthmstyle}% lwarp
64   \trivlist
65   \let\thmheadnl\relax
66   \let\thm@swap\@gobble
67   \thm@notefont{\fontseries\mddefault\upshape}%
68   \thm@headpunct{.}% add period after heading
69   \thm@headsep 5\p@ plus\p@ minus\p@\relax
70   \thm@space@setup
71   #1% style overrides
72   \@topsep \thm@preskip           % used by thm head
73   \@topsepadd \thm@postskip       % used by \@endparenv
74   \def\@tempa{#2}\ifx\@empty\@tempa
75     \def\@tempa{\@oparg{\@begintheorem{#3}{}}{}}}%
76   \else
77     \refstepcounter{#2}%
78     \def\@tempa{\@oparg{\@begintheorem{#3}{\cename the#2\endcsname}}{}}}%
79   \fi
80   \@tempa
81 }
```

cleveref patches \@thm to do \cref@thmoptarg if an optional argument is given.  
lwarp then patches \cref@thmoptarg \AtBeginDocument.

```
82 \AtBeginDocument{
```



```

83 \def\cref@thmoptarg[#1]#2#3#4{%
84   \ifhmode\unskip\unskip\par\fi%
85   \normalfont%
86   \LWR@forcenewpage% lwarp
87   \BlockClass{amsthmbody\LWR@thisthmstyle}% lwarp
88   \trivlist%
89   \let\thmheadnl\relax%
90   \let\thm@swap\@gobble%
91   \thm@notefont{\fontseries\mddefault\upshape}%
92   \thm@headpunct{.}% add period after heading
93   \thm@headsep 5\p@ plus\p@ minus\p@\relax%
94   \thm@space@setup%
95   #2% style overrides
96   \@topsep \thm@preskip           % used by thm head
97   \@topsepadd \thm@postskip       % used by \@endparenv
98   \def\@tempa{#3}\ifx\@empty\@tempa%
99     \def\@tempa{\@oparg{\@begintheorem{#4}{}}{}}[]}%
100  \else%
101    \refstepcounter{#1}{#3}% <<< cleveref modification
102    \def\@tempa{\@oparg{\@begintheorem{#4}{\csname the#3\endcsname}}{}}}%
103  \fi%
104  \@tempa
105 }%
106 %% AtBeginDocument
107
108 \def\@endtheorem{\endtrivlist\endBlockClass\@endpefalse }

```

Proof QED symbol:

```

109 \AtBeginDocument{
110 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
111 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
112 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
113 }

```

Patched for CSS:


```

114 \renewenvironment{proof}[1][\proofname]{\par
115 \LWR@forcenewpage% lwarp
116   \BlockClass{amsthmproof}% lwarp
117   \pushQED{\qed}%
118   \normalfont \topsep6\p@\@plus6\p@\relax
119   \trivlist
120   \item[
121     \InlineClass{amsthmproofname}{#1@addpunct{.}}]\ignorespaces% changes
122 ]{%
123   \InlineClass{theoremendmark}{\popQED}\endtrivlist%
124   \endBlockClass% lwarp
125   \@endpefalse

```

126 }

File 14 **lwarp-anonchap.sty**§ 101 Package **anonchap***(Emulates or patches code by PETER WILSON.)*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  $\LaTeX$  commands to create the titles, allowing other packages to work with it.

The code is shared by tocbibind.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{anonchap}

2 \newcommand{\simplechapter}[1][\@empty]{%
3   \def\@chapcntformat##1{%
4     #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
5   }%
6 }
7
8 \newcommand{\restorechapter}{%
9 \let\@chapcntformat\@secntformat%
10 }
```

File 15 **lwarp-anysize.sty**§ 102 Package **anysize***(Emulates or patches code by MICHAEL SALZENBERG, THOMAS ESSER.)*Pkg **anysize** anysize is emulated.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{anysize}

2 \def\papersize#1#2{}
3 \def\marginsize#1#2#3#4{}
```


---

File 16 **lwarp-appendix.sty**

§ 103      Package **appendix**

*(Emulates or patches code by PETER WILSON.)*

Pkg    appendix    appendix is patched for use by lwarp.

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

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{appendix}

2 \renewcommand*{\@chap@pppage}{%
3 \part*{\appendixpagename}
4 \if@dotoc@pp
5 \addappheadtotoc
6 \fi
7 }
8
9 \renewcommand*{\@sec@pppage}{%
10 \part*{\appendixpagename}
11 \if@dotoc@pp
12 \addappheadtotoc
13 \fi
14 }
```

---

File 17 **lwarp-arabicfront.sty**

§ 104      Package **arabicfront**

Pkg    arabicfront    arabicfront is ignored.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{arabicfront}
```

---

File 18 **lwarp-array.sty**

§ 105      Package **array**

Pkg    array    array is used as-is for print output, and emulated for HTML.

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{array}

2 \let\LWR@origfirstline\firstline
3 \let\LWR@origlastline\lastline
4
5 \appto{\LWR@restoreorigformatting}{%
6 \let\firstline\LWR@origfirstline%
7 \let\lastline\LWR@origlastline%
8 }
9
10 \renewcommand*{\firstline}{\LWR@HTMLhline}%
11 \renewcommand*{\lastline}{\LWR@HTMLhline}%

```

---

File 19 **lwarp-atbegshi.sty**

§ 106      Package **atbegshi**

*(Emulates or patches code by HEIKO OBERDIEK.)*

Pkg    atbegshi    Emulated.

**for HTML output:** Discard all options for lwarp-atbegshi:

```

1 \LWR@ProvidesPackageDrop{atbegshi}[2011/10/05]

2 \newcommand*{\AtBeginShipout}[1]{}
3 \newbox\AtBeginShipoutBox
4 \newcommand*{\AtBeginShipoutNext}[1]{}
5 \newcommand*{\AtBeginShipoutFirst}[1]{}
6 \newcommand*{\AtBeginShipoutDiscard}{}
7 \newcommand*{\AtBeginShipoutInit}{}
8 \newcommand*{\AtBeginShipoutAddToBox}[1]{}
9 \newcommand*{\AtBeginShipoutAddToBoxForeground}[1]{}
10 \newcommand*{\AtBeginShipoutUpperLeft}[1]{}
11 \newcommand*{\AtBeginShipoutUpperLeftForeground}[1]{}
12 \newcommand*{\AtBeginShipoutOriginalShipout}[1]{}

```

---

```

13 \def\AtBeginShipoutBoxWidth{0pt}
14 \def\AtBeginShipoutBoxHeight{0pt}
15 \def\AtBeginShipoutBoxDepth{0pt}
16

```


---

File 20 **lwarp-authblk.sty**

§ 107      Package **authblk**

*(Emulates or patches code by PATRICK W. DALY.)*

Pkg    authblk    authblk is patched for HTML.

package support    lwarp supports the native  $\LaTeX$  titling commands, and also supports the packages  
 **load order**    authblk and titling. If both are used, authblk should be loaded before titling.

$\backslash$ published and  $\backslash$ subtitle    If using the titling package, additional titlepage fields for  $\backslash$ published and  $\backslash$ subtitle may be added by using  $\backslash$ AddSubtitlePublished in the preamble. See section 58.8.

*(Emulates or patches code by PATRICK W. DALY.)*

**for HTML output:**    Require that authblk be loaded before titling:

```

1 \@ifpackageloaded{titling}{
2 \PackageError{lwarp-authblk}
3 {Package authblk must be loaded before titling}
4 {Titling appends authblk's author macro, so authblk must be loaded first.}
5 }
6 {}

```

Load authblk:

```

7 \LWR@ProvidesPackagePass{authblk}

```

Patch to add a class for the affiliation:

```

8 \LetLtxMacro\LWRAB@affil\affil
9
10 \renewcommand{\affil}[2] [] {%
11 \LWRAB@affil[#1]{\protect\InlineClass{affiliation}{#2}}
12 }

```

Create an HTML break for an  $\backslash$ authorcr:

```

13 \renewcommand*{\authorcr}{\protect\LWR@newlinebr}

```

---

File 21 **lwarp-axodraw2.sty**

§ 108     Package **axodraw2**

*(Emulates or patches code by JOHN C. COLLINS, J.A.M. VERMASEREN.)*

Pkg    axodraw2    axodraw2 is patched for use by lwarp.

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{axodraw2}

2 \BeforeBeginEnvironment{axopicture}{\begin{lateximage}[axopicture]}
3
4 \AfterEndEnvironment{axopicture}{\end{lateximage}}
```


---

File 22 **lwarp-backref.sty**

§ 109     Package **backref**

*(Emulates or patches code by DAVID CARLISLE AND SEBASTIAN RAHTZ.)*

Pkg    backref    backref is patched for use by lwarp.

 **loading** Note that backref must be explicitly loaded, and is not automatically loaded by hyperref when generating HTML output.

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{backref}

Force the hyperref option:

2 \def\backref{}\let\backrefxxx\hyper@section@backref
```

---

File 23 **lwarp-balance.sty**

§ 110     Package **balance**

*(Emulates or patches code by PATRICK W. DALY.)*

Pkg    balance    Emulated.

**for HTML output:**

Discard all options for lwarp-balance:

```
1 \LWR@ProvidesPackageDrop{balance}

2 \newcommand*{\balance}{}
3 \newcommand*{\nobalance}{}

```

---

File 24 **lwarp-bigdelim.sty**

## § 111 Package **bigdelim**

(Emulates or patches code by PIET VAN OOSTRUM, ØYSTEIN BACHE, JERRY LEICHTER.)

Pkg **bigdelim** **bigdelim** is used as-is for print or lateximage, and patched for HTML.

The delimiters are displayed in HTML by printing the delimiter, the text, and a thick border across the side of the `\multirow` which indicates the actual height of the delimiter. The delimiter character is given a `<span>` class of `ldelim` or `rdelim`, and the default CSS sets this to `font-size:200%`

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

---

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

```

---


$$\text{left} \left\{ \begin{array}{ll} a & b \\ c & d \\ e & f \\ g & h \end{array} \right.$$


---

for HTML output: First, remove the temporary definitions of `\ldelim` and `\rdelim`, which were previously defined for tabular scanning in case **bigdelim** was not loaded:

```
1 \let\ldelim\relax
2 \let\rdelim\relax

```

Next, load the package's new definitions:

```
3 \LWR@ProvidesPackagePass{bigdelim}

```

Remember the print-mode versions:

```

4 \LetLtxMacro\LWR@origldelim\ldelim
5 \LetLtxMacro\LWR@origrdelim\rdelim

\ldelim {\langle 1:delimiter\rangle} {\langle 2:#rows\rangle} {\langle 3:width\rangle} [\langle 4:text\rangle]
\rdelim

6 \RenewDocumentCommand{\ldelim}{m m m O{}}{%
7 \renewcommand{\LWR@multirowborder}{right}%
8 \multirow{#2}{#3}{#4 \InlineClass{ldelim}{#1}}%
9 }
10
11 \RenewDocumentCommand{\rdelim}{m m m O{}}{%
12 \renewcommand{\LWR@multirowborder}{left}%
13 \multirow{#2}{#3}{\InlineClass{rdelim}{#1} #4}%
14 }
```

When entering a lateximage, restore the print-mode versions:

```

15 \appto{\LWR@restoreorigformatting}{%
16 \LetLtxMacro{\ldelim}{\LWR@origldelim}%
17 \LetLtxMacro{\rdelim}{\LWR@origrdelim}%
18 }
```

---

File 25 **lwarp-bigstrut.sty**

§ 112     Package **bigstrut**

*(Emulates or patches code by PIET VAN OOSTRUM, ØYSTEIN BACHE, JERRY LEICHTER.)*

Pkg    bigstrut    bigstrut is used as-is for print or lateximage, and patched for HTML.

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{bigstrut}

2 \LetLtxMacro\LWR@origbigstrut\bigstrut
3
4 \renewcommand\bigstrut[1][x]{}
5
6 \appto{\LWR@restoreorigformatting}{%
7 \LetLtxMacro{\bigstrut}{\LWR@origbigstrut}%
8 }
```



---

File 26 **lwarp-blowup.sty**

§ 113     Package **blowup**

Pkg   blowup   blowup is ignored.

**for HTML output:**     1 \LWR@ProvidesPackageDrop{blowup}

                          2 \newcommand\*\blowUp[1]{}

---

File 27 **lwarp-bookmark.sty**

§ 114     Package **bookmark**

*(Emulates or patches code by HEIKO OBERDIEK.)*

Pkg   bookmark   bookmark is emulated.

**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 28 **lwarp-booktabs.sty**

§ 115     Package **booktabs**

*(Emulates or patches code by SIMON FEAR.)*

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 29 **lwarp-boxedminipage.sty**

§ 116      Package **boxedminipage**

Pkg   boxedminipage   boxedminipage is superseded by boxedminipage2e.

for HTML output:    1 \LWR@loadnever{boxedminipage}{boxedminipage2e}

---

File 30 **lwarp-boxedminipage2e.sty**

§ 117      Package **boxedminipage2e**

*(Emulates or patches code by SCOTT PAKIN.)*

Pkg   boxedminipage2e   boxedminipage2e is emulated.

for HTML output:    Discard all options for lwarp-boxedminipage2e:

```
1 \LWR@ProvidesPackageDrop{boxedminipage2e}

2 \newenvironment{boxedminipage}{%
3 \begin{BlockClass}{framebox}%
4 \minipage%
5 }
6 {
7 \endminipage%
8 \end{BlockClass}
9 }
```

---

File 31 **lwarp-breakurl.sty**

§ 118      Package **breakurl**

*(Emulates or patches code by VILAR CAMARA NETO.)*

Pkg    breakurl    breakurl is emulated.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{breakurl}

2 \LetLtxMacro\burl\url
3
4 \NewDocumentCommand{\burlalt}{0{} +m m}{%
5 \LWR@ensuredoingapar%
6 \def\LWR@templink{#2}%
7 \@onelevel@sanitize\LWR@templink%
8 \def\LWR@templinktwo{#3}%
9 \@onelevel@sanitize\LWR@templinktwo%
10 \LWR@subhyperref{\LWR@templink}{\LWR@templinktwo}%
11 \LWR@ensuredoingapar%
12 }
13
14 \LetLtxMacro\urlalt\burlalt

```

---

File 32 **lwarp-bytefield.sty**

§ 119      Package **bytefield**

*(Emulates or patches code by SCOTT PAKIN.)*

Pkg    bytefield    bytefield is patched for use by lwarp.

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{bytefield}

2 \BeforeBeginEnvironment{bytefield}{\begin{lateximage}[bytefield]}
3
4 \AfterEndEnvironment{bytefield}{\end{lateximage}}

```

File 33 **lwarp-cancel.sty**§ 120 Package **cancel**

Pkg `cancel` `cancel` is used as-is for SVG math, and emulated for HTML text output.

for HTML output: 

```
1 \LWR@origRequirePackage{lwarp-xcolor}% for \convertcolorspec
2 \LWR@ProvidesPackagePass{cancel}
```

`\cancelto` is math-only, so is used as-is.

```
3 \LetLtxMacro\LWR@origcancel\cancel
4 \LetLtxMacro\LWR@origbcancel\bcancel
5 \LetLtxMacro\LWR@origxcancel\xcancel
6
7 \appto{\LWR@restoreorigformatting}{%
8 \LetLtxMacro\cancel\LWR@origcancel%
9 \LetLtxMacro\bcancel\LWR@origbcancel%
10 \LetLtxMacro\xcancel\LWR@origxcancel%
11 }
```

`\LWR@cancelcolor` `{\text}{\color}{\class}{\colorstyle}{\FormatWPstyle}`

Add colors if not empty:

```
12 \newcommand{\LWR@cancelcolor}[5]{%
13 \ifcsempy{#2}%
14 {\LWR@HTMLtextstyle{#5}{#3}{#1}}%
15 {\LWR@htmlspanclass[#5;#4:\#\LWR@tempcolor]{#3}{#1}}%
16 }
```

`\cancel` `{\text}`

```
17 \DeclareRobustCommand{\cancel}[1]{%
18 \begingroup%
19 \CancelColor%
20 \LWR@findcurrenttextcolor%
21 \color{black}%
22 \LWR@cancelcolor{#1}{\LWR@tempcolor}{sout}{text-decoration-color}%
23 {text-decoration:line-through}%
24 \endgroup%
25 }
26
27 \LetLtxMacro\bcancel\cancel
28 \LetLtxMacro\xcancel\cancel
```

File 34 **lwarp-caption.sty**§ 121 Package **caption***(Emulates or patches code by AXEL SOMMERFELDT.)*

Pkg caption caption is patched for use by lwarp.

for HTML output:

```

1 \LWR@ProvidesPackagePass{caption}

2 \renewcommand\caption@ibox[3]{%
3   \@testopt{\caption@iiibox{#1}{#2}{#3}}{%
4     \wd\@tempboxa%
5     \linewidth% lwarp
6   }%
7 \LWR@traceinfo{caption@ibox: done}%
8 }

9 \long\def\caption@iiibox#1#2#3[#4]{%
10  \@testopt{\caption@iiiibox{#1}{#2}{#3}{#4}}\captionbox@hj@default
11 }

12 \long\def\caption@iiiibox#1#2#3#4[#5]#6{%
13   \setbox\@tempboxa\hbox{#6}%
14   \begingroup
15   #1*% set \caption@position
16   \caption@iftop{%
17     \LWR@traceinfo{caption@iiiibox top}%
18     \endgroup
19     \parbox[t]{#4}{%
20       #1\relax
21       \caption@setposition t%
22 %       \vbox{\caption#2{#3}}%
23       {\caption#2{#3}}% lwarp
24 %       \captionbox@hrule
25 %       \csname caption@hj@#5\endcsname
26 %       \unhbox\@tempboxa
27       #6% lwarp
28     }%
29   }{%
30     \LWR@traceinfo{caption@iiiibox bottom}%
31     \endgroup
32     \parbox[b]{#4}{%
33       #1\relax
34       \caption@setposition b%

```

```

35 %      \csname caption@hj@#5\endcsname
36 %      \unhbox\@tempboxa
37 %      #6% lwarp
38 %      \captionbox@hrule
39 %      \vtop{\caption#2{#3}}}%
40 %      {\caption#2{#3}}% lwarp
41 %      }%
42 %      }%
43 \LWR@traceinfo{caption@iiibox: done}%
44 }
45
46 \def\caption@caption{%
47   \caption@iftype
48   {%
49     \caption@checkgrouplevel\@empty\caption
50     \caption@star
51     {\caption@refstepcounter\@capttype}%
52     {\caption@dblarg{\@caption\@capttype}}}%
53   {\caption@Error{\noexpand\caption outside float}}%
54   \caption@gobble}%
55 }
56
57 \long\def\caption@@caption#1[#2]#3{%

58   \ifcaption@star \else
59     \caption@prepareanchor{#1}{#2}%
60     \memcaptioninfo{#1}{\csname the#1\endcsname}{#2}{#3}%
61     \@nameuse{nag@hascaptiontrue}%
62   \fi

63   \par
64   \caption@beginex{#1}{#2}{#3}%
65   \caption@setfloatcapt{%
66     \caption@boxrestore
67     \if@minipage
68       \@setminipage
69     \fi
70     \caption@normalsize
71     \ifcaption@star
72       \let\caption@makeanchor\@firstofone
73     \fi
74     \@makecaption{\csname fnum@#1\endcsname}%
75     {\ignorespaces\caption@makeanchor{#3}}\par
76     \caption@if@minipage\@minipagetrue\@minipagefalse}%
77   \caption@end%
78 }

```

\caption@@@make    {\caption label} {\caption text}

```

79 \renewcommand\caption@@make[2]{%
80 \LWR@startpars% lwarp
81 %   \sbox\@tempboxa{#1}%
82 %   \ifdim\wd\@tempboxa=\z@
83 %     \let\caption@lsep\relax
84 %   \fi
85 \caption@ifempty{#2}{%
86   \let\caption@lsep\@empty
87   \let\caption@tfmt\@firstofone
88 }%
89 \@setpar{\LWR@closeparagraph\@@par}% lwarp
90 \caption@applyfont
91 \caption@fmt
92   {\ifcaption@star\else
93     \begingroup
94       \captionlabelfont
95       #1%
96     \endgroup
97   \fi}%
98   {\ifcaption@star\else
99     \begingroup
100       \caption@iflf\captionlabelfont
101       \relax\caption@lsep
102     \endgroup
103   \fi}%
104   {\caption@textfont
105     \caption@ifstrut
106       {\vrule\@height\ht\strutbox\@width\z@}%
107       {}%
108     \nobreak\hskip\z@skip % enable hyphenation
109     \caption@tfmt{#2}
110   \LWR@ensuredoingapar% lwarp
111   \caption@ifstrut
112     {\ifhmode\@finalstrut\strutbox\fi}%
113     {}%
114   \par}}
115 \LWR@stoppars% lwarp
116 }

```

\caption@@make@ {<>}{<>}

```

117 \renewcommand{\caption@@make@}[2]{%
118   \caption@stepthecounter
119   \caption@beginhook
120   \caption@@make{#1}{#2}%
121   \caption@endhook
122 }

123 % \DeclareCaptionBox{none}{#2}

```

---

```

124 \DeclareCaptionBox{parbox}{%
125 #2%
126 }
127 \DeclareCaptionBox{colorbox}{%
128 #2%
129 }

```

---

File 35 **lwarp-caption2.sty**

§ 122    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 36 **lwarp-ccaption.sty**

§ 123    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 37 **lwarp-changebar.sty**

§ 124    Package **changebar**

Pkg    changebar    changebar is ignored.

for HTML output:    1 \LWR@ProvidesPackageDrop{changebar}

```

2 \newcommand*{\cbstart}{}
3 \newcommand*{\cbend}{}
4 \newenvironment*{\changebar}{}{}
5 \newcommand*{\cbdelete}{}
6 \newcommand*{\nochnagebars}{}
7 \newcommand*{\cbcolor}[1]{}
8 \newlength{\changebarwidth}
9 \newlength{\deletebarwidth}
10 \newlength{\changebarsep}
11 \newcounter{changebargrey}

```



---

File 38 **lwarp-changepage.sty**

§ 125      Package **changepage**

*(Emulates or patches code by PETER WILSON.)*

Pkg    changepage    changepage is emulated.

for HTML output:    Discard all options for lwarp-changepage:

```

1 \LWR@ProvidesPackageDrop{changepage}

2 \newif\ifoddpage
3 \DeclareRobustCommand{\checkoddpage}{\oddpagetrue}
4 \DeclareRobustCommand{\changetext}[5]{}
5 \DeclareRobustCommand{\changeage}[9]{}
6
7 \@ifundefined{adjustwidth}{
8 \newenvironment{adjustwidth}[2]{}{}
9 \newenvironment{adjustwidth*}[2]{}{}
10 }{
11 \renewenvironment{adjustwidth}[2]{}{}
12 \renewenvironment{adjustwidth*}[2]{}{}
13 }

14 \DeclareDocumentCommand{\strictpagecheck}{}{}
15 \DeclareDocumentCommand{\easypagecheck}{}{}

```

---

File 39 **lwarp-chnpage.sty**

§ 126      Package **chnpage**

Pkg    chnpage    chnpage is superceded by changepage.

for HTML output:    1 \LWR@loadnever{chnpage}{changeage}

---

File 40 **lwarp-chappg.sty**

§ 127      Package **chappg**

*(Emulates or patches code by ROBIN FAIRBAIRNS.)*

Pkg    chappg    chappg is emulated.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{chappg}

2 \renewcommand{\pagenumbering}[2] [] {}

3 \providecommand{\chappgsep}{--}

---

File 41 **lwarp-chapterbib.sty**

§ 128      Package **chapterbib**

*(Emulates or patches code by DONALD ARSENEAU.)*

Pkg    chapterbib    chapterbib is patched for use by lwarp.

**for HTML output:**    1 \LWR@ProvidesPackagePass{chapterbib}

2 \xdef\@savedjobname{\BaseJobname}

3 \let\@currentipfile\@savedjobname

---

File 42 **lwarp-cite.sty**

§ 129      Package **cite**

*(Emulates or patches code by DONALD ARSENEAU.)*

Pkg    cite    cite is patched for use by lwarp.

**for HTML output:**    1 \LWR@ProvidesPackagePass{cite}

For the [super] option, the \kern must be removed:

```
2 \def\LWRCT@biblabel#1{\@citess{#1}\kern-\labelsep\,}
```

```

3
4 \ifdefstrequal{\@biblabel}{\LWRCT@biblabel}
5 {
6   \def\@biblabel#1{\@citess{#1}}
7 }{}}

```

For the [super] option, \textsuperscript is used instead of math superscript:

```

8 \def\@citess#1{\textsuperscript{#1}}
9
10 \DeclareDocumentCommand\citepunct{}{, \, \relax}

```


---

#### File 43 **lwarp-color.sty**

### § 130 Package **color**

Pkg color Allowed but ignored. xcolor is then required as well.

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

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

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{color}
2 \RequirePackage{xcolor}

```

---

#### File 44 **lwarp-colortbl.sty**

### § 131 Package **colortbl**

Pkg colortbl colortbl is emulated.

 **row/cell color** Only use \rowcolor and \cellcolor at the start of a row, in that order.

colortbl ignores the overhang arguments.

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{colortbl}

```

Remember the print-mode definitions:

```

2 \LetLtxMacro\LWR@origcolumncolor\columncolor
3 \LetLtxMacro\LWR@origrowcolor\rowcolor
4 \LetLtxMacro\LWR@origcellcolor\cellcolor
5 \LetLtxMacro\LWR@origarrayrulecolor\arrayrulecolor
6 \LetLtxMacro\LWR@origdoublerulesepcolor\doublerulesepcolor

```

```

7
8 \appto{\LWR@restoreorigformatting}{%
9 \LetLtxMacro\columncolor\LWR@origcolumncolor
10 \LetLtxMacro\rowcolor\LWR@origrowcolor
11 \LetLtxMacro\cellcolor\LWR@origcellcolor
12 \LetLtxMacro\arrayrulecolor\LWR@origarrayrulecolor
13 \LetLtxMacro\doublerulesepcolor\LWR@origdoublerulesepcolor
14 }

```

The following \LWR@HTML versions are used inside an HTML tabular.

`\columncolor` [*<model>*] {<color>} [*<left overhang>*] [*<right overhang>*]

\LWR@getmynexttoken is not used here because \columncolor is not used inside the data area of the tabular.

```

15 \RenewDocumentCommand{\LWR@HTMLcolumncolor}{O{named} m o o}{%
16 \convertcolorspec{#1}{#2}{HTML}\LWR@columnHTMLcolor%
17 \LWR@addtabularcellcolor%
18 }

```

\LWR@getmynexttoken is used for \rowcolor because it is used inside the data area of the tabular.

`\rowcolor` [*<model>*] {<color>} [*<left overhang>*] [*<right overhang>*]

```

19 \RenewDocumentCommand{\LWR@HTMLrowcolor}{O{named} m o o}{%
20 \convertcolorspec{#1}{#2}{HTML}\LWR@rowHTMLcolor%
21 \LWR@getmynexttoken%
22 }

```

`\cellcolor` [*<model>*] {<color>} [*<left overhang>*] [*<right overhang>*]

```

23 \RenewDocumentCommand{\LWR@HTMLcellcolor}{O{named} m o o}{%
24 \convertcolorspec{#1}{#2}{HTML}\LWR@cellHTMLcolor%
25 \LWR@addtabularcellcolor%
26 }

```

`\arrayrulecolor` [*<model>*] {<color>}

The version for use outside a tabular.

```

27 \renewcommand{\arrayrulecolor}[2] [named]{%
28 \convertcolorspec{#1}{#2}{HTML}\LWR@ruleHTMLcolor%
29 }

```

`\LWR@arrayrulecolor` [*<model>*] {<color>}

The version for use inside a tabular.

---

```

30 \renewcommand{\LWR@HTMLarrayrulecolor}[2] [named] {%
31 \convertcolorspec{#1}{#2}{HTML}\LWR@ruleHTMLcolor%
32 \LWR@getmynexttoken%
33 }

```

`\doublerulesepcolor`    [*<model>*] {*<color>*}

The version for use outside a tabular.

```

34 \renewcommand{\doublerulesepcolor}[2] [named] {}

```

`\LWR@doublerulesepcolor`    [*<model>*] {*<color>*}

The version for use inside a tabular.

```

35 \renewcommand{\LWR@HTMLdoublerulesepcolor}[2] [named] {\LWR@getmynexttoken}

```

---

#### File 45 **lwarp-continue.sty**

### § 132    Package **continue**

Pkg    `continue`    `continue` is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{continue}

```

2 \newcommand*{\flagcont}{}
3 \newcommand*{\flagend}{}
4 \newcommand*{\flagword}{}
5 \newcommand*{\preflagword}{}
6 \newcommand*{\postflagword}{}
7 \newlength\contsep
8 \newlength\contdrop

```

---

#### File 46 **lwarp-crop.sty**

### § 133    Package **crop**

*(Emulates or patches code by MELCHIOR FRANZ.)*

Pkg    `crop`    Emulated.

**for HTML output:**    Discard all options for `lwarp-crop`:

```

1 \LWR@ProvidesPackageDrop{crop}

```

---

```
2 \newcommand*{\crop}[1] [] {}
3 \newcommand*{\cropdef}[6] [] {}
```

---

File 47 **lwarp-cuted.sty**

§ 134     Package **cuted**

*(Emulates or patches code by SIGITAS TOLUŠIS.)*

Pkg    cuted    cuted is emulated.

for HTML output:    1 \LWR@ProvidesPackageDrop{cuted}

```
2 \newenvironment{strip}{}{}
3 \newskip\stripsep
4 \def\oldcolsbreak#1{}
```

---

File 48 **lwarp-cutwin.sty**

§ 135     Package **cutwin**

*(Emulates or patches code by PETER WILSON AND ALAN HOENIG.)*

Pkg    cutwin    Emulated.

for HTML output:    Discard all options for lwarp-cutwin:

```
1 \LWR@ProvidesPackageDrop{cutwin}

2 \newcommand*{\opencutleft}{}
3 \newcommand*{\opencutright}{}
4 \newcommand*{\opencutcenter}{}
5 \newcommand*{\cutfuzz}{}
6
7 \newenvironment{cutout}[4]
8 {\marginpar{\windowpagestuff}}
9 {}
10
11 \newcommand*{\windowpagestuff}{}
12
13 \newcommand*{\pageinwindow}{%
14 % \begin{minipage}{.3\linewidth}
15 \windowpagestuff
16 % \end{minipage}}
```

---

```

17 }
18
19 \newenvironment{shapedcutout}[3]
20 {\marginpar{\picinwindow}}
21 {}
22
23 \newcommand*{\putstuffinpic}{}
24
25 \newcommand*{\picinwindow}{%
26 \begin{picture}(0,0)
27 \putstuffinpic
28 \end{picture}}

```

---

File 49 **lwarp-dblfloatfix.sty**

§ 136     Package **dblfloatfix**

Pkg    dblfloatfix    dblfloatfix is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{dblfloatfix}

---

File 50 **lwarp-dblfnote.sty**

§ 137     Package **dblfnote**

*(Emulates or patches code by HIROSHI NAKASHIMA.)*

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 51 **lwarp-dcolumn.sty**

§ 138      Package **dcolumn**

Pkg    dcolumn    dcolumn is emulated by the lwarp core.

```
1 \LWR@ProvidesPackageDrop{dcolumn}
```

---

File 52 **lwarp-diagbox.sty**

§ 139      Package **diagbox**

*(Emulates or patches code by LEO LIU.)*

Pkg    diagbox    diagbox is patched for use by lwarp.

for HTML output:    1 \LWR@ProvidesPackagePass{diagbox}

To restore print-mode inside a lateximage:

```
2 \LetLtxMacro\LWR@origdiagbox@double\diagbox@double
3 \LetLtxMacro\LWR@origdiagbox@triple\diagbox@triple
4
5 \appto\LWR@restoreorigformatting{
6 \LetLtxMacro\diagbox@double\LWR@origdiagbox@double
7 \LetLtxMacro\diagbox@triple\LWR@origdiagbox@triple
8 }
```

```
\LWR@diagbox@AB    {\langle A \rangle} {\langle B \rangle}
9 \newcommand{\LWR@diagbox@AB}[4]{
10 \begingroup%
11 \LetLtxMacro\\\newline%
12 \BlockClassSingle{diagbox#1}{#2}%
13 \BlockClassSingle{diagbox#3}{#4}%
14 \endgroup%
15 \LWR@stoppars%
16 }
```

```
\LWR@diagboxNW    {\langle A \rangle} {\langle B \rangle}
```



```

17 \newcommand{\LWR@diagboxNW}[2]{%
18 \LWR@diagbox@AB{E}{#2}{W}{#1}%
19 }

```

Likewise for NE, SW, SE:

```

20 \newcommand{\LWR@diagboxNE}[2]{%
21 \LWR@diagbox@AB{W}{#1}{E}{#2}%
22 }
23
24 \let\LWR@diagboxSW\LWR@diagboxNE
25 \let\LWR@diagboxSE\LWR@diagboxNW

```

```

\LWR@diagbox@double  {\langle keys \rangle} {\langle A \rangle} {\langle B \rangle}

26 \def\diagbox@double#1#2#3{%
27 \setkeys{diagbox}{dir=NW,#1}%
28 \csuse{\LWR@diagbox\diagbox@dir}{#2}{#3}%
29 }

```

```

\LWR@diagboxTNW  {\langle title \rangle} {\langle A \rangle} {\langle B \rangle}

30 \newcommand{\LWR@diagboxTNW}[3]{%
31 \BlockClassSingle{diagboxtitleN}{#1}
32 \LWR@diagboxNW{#2}{#3}
33 }

```

Likewise for NE, SW, SE:

```

34 \newcommand{\LWR@diagboxTNE}[3]{%
35 \BlockClassSingle{diagboxtitleN}{#1}
36 \LWR@diagboxNE{#2}{#3}
37 }
38
39 \newcommand{\LWR@diagboxTSW}[3]{%
40 \LWR@diagboxSW{#2}{#3}
41 \BlockClassSingle{diagboxtitleS}{#1}
42 }
43
44 \newcommand{\LWR@diagboxTSE}[3]{%
45 \LWR@diagboxSE{#2}{#3}
46 \BlockClassSingle{diagboxtitleS}{#1}
47 }

```

```

\LWR@diagbox@triple  {\langle keys \rangle} {\langle A \rangle} {\langle T \rangle} {\langle B \rangle}

48 \def\diagbox@triple#1#2#3#4{%
49 \setkeys{diagbox}{dir=NW,#1}%

```

---

```
50 \csuse{LWR@diagboxT\diagbox@dir}{#3}{#2}{#4}%
51 }
```

---

File 53 **lwarp-draftwatermark.sty**

§ 140      Package **draftwatermark**

*(Emulates or patches code by SERGIO CALLEGARI.)*

Pkg    draftwatermark    draftwatermark is emulated.

**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 54 **lwarp-easy-todo.sty**

§ 141      Package **easy-todo**

*(Emulates or patches code by JUAN RADA-VILELA.)*

Pkg    easy-todo    easy-todo is patched for use by lwarp.

**for HTML output:**

```
1 \LWR@ProvidesPackagePass{easy-todo}

\listoftodos    Modified to correct buggy use of \flushright.

2 \let\LWR@origlistoftodos\listoftodos
3
4 \renewcommand{\listoftodos}{%
5 \begingroup
6 \renewcommand{\flushright}{}
7 \LWR@origlistoftodos
8 \endgroup
9 }
```

---

```

\todoii Modified to use \textcolor instead of \color.

10 \renewcommand{\todoii}[2]{%
11 \ifthenelse{\equal{\@todoobeyfinal}{true}}{%
12   \ifoptionfinal{\todoenable{false}}{\todoenable{true}}%
13 }{}%
14 \ifthenelse{\equal{\@todoenable}{true}}{%
15 \refstepcounter{todos}%
16 \noindent{%
17   \todocolor%
18   \LWR@textcurrentcolor{%
19     \normalfont\scriptsize{\bfseries{\thetodos.#1}}%
20   }%
21 }%
22 \addcontentsline{lod}{todos}{\protect{\thetodos. }#2}%
23 }{}%
24 }

```

---

File 55 **lwarp-ebook.sty**

§ 142 Package **ebook**

*(Emulates or patches code by JØRGEN STEENSGAARD.)*

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 56 **lwarp-ellipsis.sty**

§ 143 Package **ellipsis**

*(Emulates or patches code by PETER J. HESLIN.)*

Pkg ellipsis ellipsis is emulated.

---

```

1 \LWR@ProvidesPackageDrop{ellipsis}
2
3 \newcommand{\ellipsisgap}{0.1em}

```

---

File 57 **lwarp-emptypage.sty**

\$ 144    Package **emptypage**

Pkg    emptypage    emptypage is ignored.

for HTML output:    Discard all options for lwarp-emptypage:

```

1 \LWR@ProvidesPackageDrop{emptypage}

```

---

File 58 **lwarp-endfloat.sty**

\$ 145    Package **endfloat**

Pkg    endfloat    endfloat is ignored.

for HTML output:    1 \LWR@ProvidesPackageDrop{endfloat}

```

2 \newcommand\figureplace{}
3 \newcommand\tableplace{}
4 \newcommand\floatplace[1]{}
5 \newcounter{posttable}
6 \newcounter{postfigure}
7 \newcommand*{\theposttbl}{}
8 \newcommand*{\thepostfig}{}
9 \newcommand{\AtBeginFigures}[1]{}
10 \newcommand{\AtBeginTables}[1]{}
11 \newcommand{\AtBeginDelayedFloats}[1]{}
12 \newcommand*{\processdelayedfloats}{}
13 \newcommand*{\efloatseparator}{}

```

---

File 59 **lwarp-endheads.sty**

\$ 146    Package **endheads**

Pkg    endheads    endheads is ignored.

for HTML output:    1 \LWR@ProvidesPackageDrop{endheads}

---

```

2 \newcommand{\changesinglepageabbrev}[1]{}
3 \newcommand{\changemultiplepageabbrev}[1]{}
4 \newcommand{\changenotesname}[1]{}
5 \newcommand{\changenotesheader}[1]{}
6 \newcommand{\changenotescontentsname}[1]{}
7 \newcommand{\changechapternotesline}[1]{}
8 \newcommand{\checknoteheaders}{}
9 \newif\ifnotesincontentson \notesincontentsonfalse
10 \newcommand{\notesincontents}{\notesincontentsontrue}
11 \newif\ifendnoteheaderson \endnoteheadersonfalse
12 \newcommand{\setupendnoteheaders}{%
13   \endnoteheadersontrue%
14 }
15 \newif\iftitleinnotes \titleinnotesttrue
16 \newcommand{\styleforchapternotebegin}{}
17 \newcommand{\styleforchapternoteend}{}
18 \newcommand{\setstyleforchapternotebegin}[1]{%
19   \renewcommand{\styleforchapternotebegin}{#1}%
20 }
21 \newcommand{\setstyleforchapternoteend}[1]{%
22   \renewcommand{\styleforchapternoteend}{#1}%
23 }
24 \newcommand{\resetendnotes}{}
25 \newif\ifnotesbychapteron \notesbychapteronfalse
26 \newcommand{\notesbychapter}{\notesbychapterontrue}

```

---

File 60 **lwarp-endnotes.sty**

§ 147     Package **endnotes**

*(Emulates or patches 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 61 **lwarp-enumerate.sty**

§ 148     Package **enumerate**

Pkg    **enumerate**    enumerate is supported with no changes.

This package is only required because it was used in the past to drop and then emulate the package. It cannot be removed because an older version which dropped the package may still remain, for example in a local vs. distribution directory, but it is now supported directly by lwarp and thus must no longer be dropped.

**for HTML output:**    1 \LWR@ProvidesPackagePass{enumerate}

---

File 62 **lwarp-enumitem.sty**

§ 149     Package **enumitem**

*(Emulates or patches code by JAVIER BEZOS.)*

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 63 `lwarp-epigraph.sty`

## § 150 Package `epigraph`

*(Emulates or patches code by PETER WILSON.)*

Pkg `epigraph` `epigraph` is emulated.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{epigraph}

2 \DeclareDocumentCommand{\qitem}{m m}
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 \DeclareDocumentCommand{\epigraph}{m m}
14 {
15 \begin{LWR@BlockClassWP}{\LWR@origmbox{text-align:right}}{\epigraph}
16 \qitem{#1}{#2}
17 \end{LWR@BlockClassWP}
18 }
19
20 \DeclareDocumentEnvironment{epigraphs}{}
21 {\LWR@BlockClassWP{\LWR@origmbox{text-align:right}}{\epigraph}}
22 {\endLWR@BlockClassWP}

```

Use CSS to format epigraphs.

The following are null commands for source compatibility:

---

```

23 \newenvironment*{flushepinormal}{}{}

24 \@ifclassloaded{memoir}{
25 \setlength{\epigraphwidth}{.5\linewidth}
26 \renewcommand{\textflush}{flushepinormal}
27 \renewcommand{\epigraphhead}[2][0]{#2}
28 \renewcommand{\dropchapter}[1]{}
29 \renewcommand*{\undodrop}{}
30 }{% not memoir
31 \newlength{\epigraphwidth}
32 \setlength{\epigraphwidth}{.5\linewidth}
33 \newcommand{\textflush}{flushepinormal}
34 \newcommand{\epigraphflush}{flushright}
35 \newcommand{\sourceflush}{flushright}
36 \newcommand*{\epigraphsize}{\small}
37 \newlength{\epigraphrule}
38 \newlength{\beforeepigraphskip}
39 \newlength{\afterepigraphskip}
40 \newcommand{\epigraphhead}[2][0]{#2}
41 \newcommand{\dropchapter}[1]{}
42 \newcommand*{\undodrop}{}
43 }% not memoir
44
45 \let\cleartoevenpage\relax% also in nextpage
46 \newcommand{\cleartoevenpage}[1][{}]{

```

---

File 64 **lwarp-epstopdf.sty**

§ 151     Package **epstopdf**

Pkg    epstopdf    epstopdf is ignored.

Filenames should be used without a suffix so that SVG, PNG, or JPG versions of the file will be used for HTML output.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{epstopdf}

2 \newcommand*{\epstopdfsetup}[1]{}
3 \newcommand*{\epstopdfcall}[1]{}
4 \newcommand*{\epstopdfDeclareGraphicsRule}[4]{}

```



File 65 **lwarp-eso-pic.sty**§ 152 Package **eso-pic***(Emulates or patches code by ROLF NIEPRASCHK.)*Pkg **eso-pic** **eso-pic** is emulated.

**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 \newcommand{\AddToShipoutPicture}{\AddToShipoutPictureBG}
14 \NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
15 \newcommand*{\ClearShipoutPictureBG}{}
16 \newcommand*{\ClearShipoutPicture}{}
17 \newcommand*{\ClearShipoutPictureFG}{}
18 \newcommand{\gridSetup}[6][1]{}

```

File 66 **lwarp-everypage.sty**§ 153 Package **everypage***(Emulates or patches code by SERGIO CALLEGARI.)*Pkg **everypage** **everypage** is emulated.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{everypage}

2 \newcommand*{\AddEverypageHook}[1]{}
3 \newcommand*{\AddThispageHook}[1]{}

```

---

File 67 **lwarp-everyshi.sty**

§ 154     Package **everyshi**

*(Emulates or patches code by MARTIN SCHRÖDER.)*

Pkg **everyshi**     Emulated.

**for HTML output:**     Discard all options for lwarp-everyshi:

```
1 \LWR@ProvidesPackageDrop{everyshi}

2 \newcommand*{\EveryShipout}[1]{}
3 \newcommand*{\AtNextShipout}[1]{}

```

---

File 68 **lwarp-extramarks.sty**

§ 155     Package **extramarks**

*(Emulates or patches code by PIET VAN OOSTRUM.)*

Pkg **extramarks**     extramarks is emulated.

**for HTML output:**     Discard all options for lwarp-extramarks:

```
1 \LWR@ProvidesPackageDrop{extramarks}

2 \newcommand*{\extramarks}[2]{}
3 \newcommand*{\firstleftxmark}{}
4 \newcommand*{\lastleftxmark}{}
5 \newcommand*{\firstrightxmark}{}
6 \newcommand*{\lastrightxmark}{}
7 \newcommand*{\firstxmark}{}
8 \newcommand*{\lastxmark}{}
9 \newcommand*{\topxmark}{}
10 \newcommand*{\topleftxmark}{}
11 \newcommand*{\firstleftmark}{}
12 \newcommand*{\lastrightmark}{}

```

---

File 69 **lwarp-fancybox.sty**

§ 156     Package **fancybox**

*(Emulates or patches code by TIMOTHY VAN ZANDT.)*

Pkg   fancybox   fancybox is supported with some patches.

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

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

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

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

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

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

```

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

```

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

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

**for HTML output:**

```

1 \begin{warpHTML}

2 \LWR@ProvidesPackagePass{fancybox}

3 \renewcommand*{\@shadowbox}[1]{%
4 \ifbool{FormatWP}%
5 {\InlineClass[border:1px solid black]{shadowbox}{#1}}%
6 {\InlineClass{shadowbox}{#1}}%
7 }
8
9 \renewcommand*{\@doublebox}[1]{%
10 \ifbool{FormatWP}%
11 {\InlineClass[border:1px double black]{doublebox}{#1}}%
12 {\InlineClass{doublebox}{#1}}%
13 }
14
15 \renewcommand*{\@ovalbox}[2]{%
16 \ifbool{FormatWP}%
17 {\InlineClass[border:1px solid black; border-radius:1ex]{ovalbox}{#2}}%
18 {%
19   \ifthenelse{\isequivalentto{#1}{\thinlines}}%
20   {\InlineClass{ovalbox}{#2}}%
21   {\InlineClass{Ovalbox}{#2}}%
22 }%
23 }

```

Convert minipages, parboxes, and lists into linear text using the LWR@nestspan environment:

```

24 \let\LWR@origSbox\Sbox
25
26 \def\Sbox{\LWR@origSbox\LWR@nestspan}
27

```

```

28
29 \let\LWR@origendSbox\endSbox
30
31 \def\endSbox{\endLWR@nestspan\LWR@origendSbox}

```

Beqnarray is adapted for MATHJAX or enclosed inside a lateximage:

```

32 \RenewEnviron{Beqnarray}
33 {\LWR@eqnarrayfactor}
34
35 \csgpreto{Beqnarray*}{\boolfalse{LWR@numbereqnarray}}

```

\GenericCaption is enclosed in an HTML block:

```

36 \renewcommand{\GenericCaption}[1]{%
37 \LWR@figcaption%
38 #1%
39 \endLWR@figcaption%
40 }

```

Btrivlist is enclosed in an HTML block:

```

41 \RenewDocumentEnvironment{Btrivlist}{m o}
42 {\begin{BlockClass}{Btrivlist}\tabular{#1}}
43 {\endtabular\end{BlockClass}}

```

Btrivlist is also neutralized when used inside a span:

```

44 \AtBeginEnvironment{LWR@nestspan}{%
45 \RenewDocumentEnvironment{Btrivlist}{m o}{\}%
46 \RenewDocumentCommand{\LWR@origitem}{d()}{\LWRFB@origitem}%
47 }

```

lwarp's handling of \item is patched to accept fancybox's optional arguments:

```

48 \let\LWRFB@origitemizeitem\LWR@itemizeitem
49 \let\LWRFB@origdescitem\LWR@descitem
50 \LetLtxMacro{\LWRFB@origitem}{\LWR@origitem}
51
52 \RenewDocumentCommand{\LWR@itemizeitem}{d()}{\LWRFB@origitemizeitem}
53 \RenewDocumentCommand{\LWR@descitem}{d()}{\LWRFB@origdescitem}

```

The various boxed lists become regular lists:

```

54 \renewenvironment{Bitemize}[1][\begin{itemize}]{\end{itemize}}
55 \renewenvironment{Benumerate}[1][\begin{enumerate}]{\end{enumerate}}
56 \renewenvironment{Bdescription}[1][\begin{description}]{\end{description}}

```

\boxput simply prints one then the other argument, side-by-side instead of above and behind:

```
57 \RenewDocumentCommand{\boxput}{s d() m m}{%
58 \IfBooleanTF{#1}{#3\quad#4}{#4\quad#3}%
59 }
```

Neutralized commands:

```
60 \RenewDocumentCommand{\fancyput}{s d() m}{%
61 \RenewDocumentCommand{\thisfancyput}{s d() m}{%
62
63 \RenewDocumentCommand{\fancypage}{m m}{%
64 \RenewDocumentCommand{\thisfancypage}{m m}{%
65
66 \def\LandScape#1{%
67 \def\endLandScape{%
68 \def\@LandScape#1#2#3{%
69 \def\endLandScape{%
```

Low-level patches for UseVerbatim and friends:

```
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 70 **lwarp-fancyheadings.sty**

§ 157      Package **fancyheadings**

Pkg    fancyheadings    fancyheadings is superceded by fancyhdr.

for HTML output:    1 \LWR@loadnever{fancyheadings}{fancyhdr}

---

File 71 **lwarp-fancyhdr.sty**

§ 158      Package **fancyhdr**

*(Emulates or patches code by PIET VAN OOSTRUM.)*

Pkg    fancyhdr    fancyhdr is nullified.

for HTML output:    Discard all options for lwarp-fancyhdr:

```
1 \LWR@ProvidesPackageDrop{fancyhdr}

2 \newcommand*{\fancyhead}[2] [] {}
3 \newcommand*{\fancyfoot}[2] [] {}
4 \newcommand*{\fancyhf}[2] [] {}
5 \newcommand*{\fancypagestyle}[2] {}
6 \newcommand*{\lhead}[2] [] {}
7 \newcommand*{\chead}[2] [] {}
8 \newcommand*{\rhead}[2] [] {}
9 \newcommand*{\lfoot}[2] [] {}
10 \newcommand*{\cfoot}[2] [] {}
11 \newcommand*{\rfoot}[2] [] {}
12 \newcommand*{\headrulewidth}{}
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 72 **lwarp-fancyref.sty**

§ 159     Package **fancyref**

Pkg   fancyref   fancyref is emulated.

for HTML output:    1 \LWR@ProvidesPackagePass{fancyref}

To remove the margin option, if \fancyrefhook is anything other than the paren option, then force it to the default instead. (Comparing to the margin option was not possible since lwarp has revised the meaning of \mbox so the comparison failed.)

```

2 \newcommand*{\LWRfref@parenfancyrefhook}[1]{(#1)}
3
4 \ifdefstrequal{\fancyrefhook}{\LWRfref@parenfancyrefhook}
5 {}{
6   \renewcommand*{\fancyrefhook}[1]{#1}%
7 }

```

Modified to ignore the page number and varioref.

```

8 \renewcommand*{\@f@ref}[4]{%
9   \ifundefined{#1r@#2@#3}{%
10    \PackageError{fancyref}{%
11      \backslashchar#1ref\space format ‘#2’
12      undefined\MessageBreak
13      for label type ‘#3’}%
14    }{%
15      The format ‘#2’ was not defined for the label type
16      ‘#3’\MessageBreak
17      and the \backslashchar#1ref\space command. Perhaps
18      you have only misspelled its name.\MessageBreak
19      Otherwise you will have to define it with
20      \protect\new#1refformat\MessageBreak
21      prior to using it.%
22    }%
23  }{%
24    \fancyrefhook{%
25      \@nameuse{#1r@#2@#3}%
26      {\ref{#3\fancyrefargdelim#4}}}%
27    {\pageref{#3\fancyrefargdelim#4}}% original
28    {\@fancyref@page@ref{#3\fancyrefargdelim#4}}% original
29    }% lwarp
30    }% lwarp
31  }%

```



---

```

32 }%
33 }%

```

---

File 73 **lwarp-fancyvrb.sty**

§ 160      Package **fancyvrb**

*(Emulates or patches code by TIMOTHY VAN ZANDT.)*

Pkg fancyvrb fancyvrb is supported with some patches.

**for HTML output:**    1 \RequirePackage{xcolor}% for \convertcolorspec  
                          2  
                          3 \LWR@ProvidesPackagePass{fancyvrb}

**for HTML output:**    4 \begin{warpHTML}

Initial default patch for fancyvrb:

```
5 \fvset{frame=none}%
```

For \VerbatimFootnotes:

```

6 \renewcommand{\VerbatimFootnotes}{
7 \PackageError{lwarp}
8 {Verbatim footnotes are not yet supported by lwarp.}
9 {This may be improved some day.}
10 }

```

After the preamble is loaded, after any patches to Verbatim:

```

11 \AfterEndPreamble{
12 \LWR@traceinfo{Patching Verbatim.}

13 \preto\FVB@Verbatim{\LWR@forcenewpage}
14 \preto\FVB@LVerbatim{\LWR@forcenewpage}
15 % \preto\FVB@BVerbatim{\LWR@forcenewpage}% Fails, so done below.

```

Simplified to remove PDF formatting:

```

16 \def\FV@BeginListFrame@Single{%
17 \FV@SingleFrameLine{z}%
18 }
19
20 \def\FV@EndListFrame@Single{%
21 \FV@SingleFrameLine{@ne}%

```

```

22 }
23
24 \def\FV@BeginListFrame@Lines{%
25   \FV@SingleFrameLine{\z@}%
26 }
27
28 \def\FV@endListFrame@Lines{%
29   \FV@SingleFrameLine{\@ne}%
30 }
31
32 \renewcommand*\FV@SingleFrameSep{}

```

Adds HTML formatting:

```

33 \def\FV@BUseVerbatim#1{%
34   \LWR@atbeginverbatim[\LWR@FVstyle]{verbatim}%
35   \FV@BVerbatimBegin#1\FV@BVerbatimEnd%
36   \LWR@afterendverbatim%
37 }

```

`\LWR@FVstyle` Holds the style of the verbatim.

```

38 \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.

```

39 \newcommand*\LWR@fvstartnone{%
40   \LWR@traceinfo{fvstartnone}%
41   % \hbox to\z@{
42   \LWR@atbeginverbatim[\LWR@FVstyle]{verbatim}%
43   % }%
44 }
45
46 \newcommand*\LWR@fvendnone{%
47   \LWR@traceinfo{fvendnone}%
48   % \hbox to\z@{
49   \LWR@afterendverbatim%
50   % }%
51 }
52
53 \newcommand*\LWR@fvstartsingle{%
54   \LWR@traceinfo{fvstartsingle}%
55   \LWR@fvstartnone%
56   \FV@BeginListFrame@Single%
57 }
58

```

```

59 \newcommand*{\LWR@fvendsingle}{%
60 \LWR@traceinfo{fvendsingle}%
61 \FV@EndListFrame@Single%
62 \LWR@fvendnone%
63 }
64
65 \newcommand*{\LWR@fvstartline}{%
66 \LWR@traceinfo{fvstartline}%
67 \LWR@fvstartnone%
68 % \setlength{\LWR@templengthone}{\baselineskip}%
69 \FV@BeginListFrame@Lines%
70 % \setlength{\baselineskip}{\LWR@templengthone}%
71 % \setlength{\baselineskip}{5pt}%
72 }
73
74 \newcommand*{\LWR@fvendline}{%
75 \LWR@traceinfo{fvendline}%
76 \FV@EndListFrame@Lines%
77 \LWR@fvendnone%
78 }

```

The following patches select the start/left/right/end behaviors depending on frame. Original code is from the fancyvrb package.

```

79 \newcommand*{\LWR@FVfindbordercolor}{%
80 \FancyVerbRuleColor%
81 \LWR@findcurrenttextcolor%
82 \color{black}%
83 }
84
85 % border width of \FV@FrameRule
86 \newcommand*{\LWR@FVborderstyle}[1]{%
87 padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ; %
88 \LWR@FVfindbordercolor %
89 border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt solid \#\LWR@tempcolor ; %
90 }
91
92 \def\FV@Frame@none{%
93 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle}%
94 \let\FV@BeginListFrame\LWR@fvstartnone%
95 \let\FV@LeftListFrame\relax%
96 \let\FV@RightListFrame\relax%
97 \let\FV@EndListFrame\LWR@fvendnone%
98
99 \FV@Frame@none% default values
100
101 \def\FV@Frame@single{%
102 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{}}%
103 \let\FV@BeginListFrame\LWR@fvstartsingle%

```

```

104 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
105 \let\FV@RightListFrame\FV@RightListFrame@Single%
106 \let\FV@EndListFrame\LWR@fvendsingle}
107
108 \def\FV@Frame@lines{%
109 \renewcommand*{\LWR@FVstyle}{%
110   \LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}\LWR@FVborderstyle{-bottom}}%
111 }%
112 \let\FV@BeginListFrame\LWR@fvstartline%
113 \let\FV@LeftListFrame\relax%
114 \let\FV@RightListFrame\relax%
115 \let\FV@EndListFrame\LWR@fvendline}
116
117 \def\FV@Frame@topline{%
118 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}}%
119 \let\FV@BeginListFrame\LWR@fvstartline%
120 \let\FV@LeftListFrame\relax%
121 \let\FV@RightListFrame\relax%
122 \let\FV@EndListFrame\LWR@fvendnone}
123
124 \def\FV@Frame@bottomline{%
125 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-bottom}}%
126 \let\FV@BeginListFrame\LWR@fvstartnone%
127 \let\FV@LeftListFrame\relax%
128 \let\FV@RightListFrame\relax%
129 \let\FV@EndListFrame\LWR@fvendline}
130
131 \def\FV@Frame@leftline{%
132 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-left}}%
133 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
134 \ifx\FancyVerbFillColor\relax%
135 \let\FV@FrameFillLine\relax%
136 \else%
137 \@tempdima\FV@FrameRule\relax%
138 \multiply\@tempdima-\tw@%
139 \edef\FV@FrameFillLine{%
140 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}}%
141 \kern-\number\@tempdima sp}}%
142 \fi%
143 \let\FV@BeginListFrame\LWR@fvstartnone%
144 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
145 \let\FV@RightListFrame\relax%
146 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the fancyvrb package.

```

147 \def\FV@SingleFrameLine#1{%
148 %   \hbox to\z@{%

```

```

149 %    \kern\leftmargin
150    \ifnum#1=\z@\relax
151        \let\FV@Label\FV@LabelBegin
152    \else
153        \let\FV@Label\FV@LabelEnd
154    \fi
155    \ifx\FV@Label\relax
156 %        \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
157    \else
158        \ifnum#1=\z@
159 %            \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
160            \ifx\FV@LabelPositionTopLine\relax
161            \else
162                \LWR@FVfindbordercolor
163                \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
164                \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
165                \LWR@htmltagc{/div}
166            \fi
167        \else
168 %            \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
169            \ifx\FV@LabelPositionBottomLine\relax
170            \else
171                \LWR@FVfindbordercolor
172
173                \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
174                \LWR@origtextrm{\FV@LabelEnd}
175                \LWR@htmltagc{/div}
176            \fi
177        \fi
178    \fi
179 %    \hss
180 %    }
181 }

```

Processes each line, adding optional line numbers. Original code is from the fancyvrb package.

```

182 \def\FV@ListProcessLine#1{%
183     \hbox to \hsize{%
184 %         \kern\leftmargin
185         \hbox to \VerbatimHTMLWidth {%
186             \ifcvoid\FV@LeftListNumber}{\kern 2.5em}%
187             \FV@LeftListNumber%
188 %         \FV@LeftListFrame
189         \FancyVerbFormatLine{#1}%
190         \hss%
191 %         \FV@RightListFrame
192         \FV@RightListNumber%
193     }%

```

```

194      \hss% required to avoid underfull hboxes
195 }
196 }

```

Env BVerbatim

```

197 \AtBeginEnvironment{BVerbatim}
198 {%
199 \LWR@forcenewpage% instead of \preto
200 \LWR@atbeginverbatim{bverbatim}%
201 }
202
203 \AfterEndEnvironment{BVerbatim}
204 {%
205 \leavevmode\par\LWR@origvspace{-\baselineskip}%
206 \LWR@afterendverbatim%
207 }

```

End of the modifications to make at the end of the preamble:

```

208 } % \AfterEndPreamble

209 \end{warpHTML}

```

---

## File 74 lwarp-figcaps.sty

### § 161 Package figcaps

*(Emulates or patches code by PATRICK W. DALY.)*

Pkg figcaps Emulated.

for HTML output: Discard all options for lwarp-figcaps:

```

1 \LWR@ProvidesPackageDrop{figcaps}

2 \newcommand*{\figcapson}{}
3 \newcommand*{\figcapsoff}{}
4 \newcommand*{\printfigures}{}
5 \newcommand*{\figmarkon}{}
6 \newcommand*{\figmarkoff}{}
7 \def\figurecapname{Figure Captions}
8 \def\tablepagename{Tables}
9 \def\figurepagename{Figures}

```

---

File 75 **lwarp-figsize.sty**

§ 162      Package **figsize**

*(Emulates or patches code by ANTHONY A. TANBAKUCHI.)*

Pkg    **figsize**    figsize is emulated.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{figsize}

Emulates a virtual 6×9 inch textsize.

```
2 \newlength{\figwidth}
3 \newlength{\figheight}
4
5 \newcommand{\SetFigLayout}[3][0]{%
6 \setlength{\figheight}{8in}%
7 \setlength{\figheight}{\figheight / #2}%
8 %
9 \setlength{\figwidth}{5.5in}%
10 \setlength{\figwidth}{\figwidth / #3}%
11 }
```

---

File 76 **lwarp-fix2col.sty**

§ 163      Package **fix2col**

Pkg    **fix2col**    fix2col is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{fix2col}


---

File 77 **lwarp-fixme.sty**

§ 164      Package **fixme**

*(Emulates or patches code by DIDIER VERNA.)*

Pkg    **fixme**    fixme is patched for use by lwarp.

 **external layouts**    External layouts (\fxloadlayouts) are not supported.

User control is provided for setting the HTML styling of the “faces”. The defaults are as follows, and may be changed in the preamble after `fixme` is loaded:

```
\def\FXFaceInlineHTMLStyle{font-weight:bold}
\def\FXFaceEnvHTMLStyle{font-weight:bold}
\def\FXFaceSignatureHTMLStyle{font-style:italic}
\def\FXFaceTargetHTMLStyle{font-style:italic}
```

for HTML output: `1 \LWR@ProvidesPackagePass{fixme}`

Restore `lwarp`’s version of `\@wrindex`, ignoring the `fixme` package’s target option:

```
2 \let\@wrindex\LWR@wrindex
```

Float-related macros required by `lwarp`:

```
3 \newcommand{\ext@fixme}{lox}
4
5 \renewcommand{\l@fixme}[2]{\hypertocfloat{1}{fixme}{lox}{#1}{#2}}
```

Other modifications:

```
6 \def\FXFaceInlineHTMLStyle{font-weight:bold}
7
8 \renewcommand*{\FXLayoutInline}[3]{ %
9 \InlineClass[\FXFaceInlineHTMLStyle]{fixmeinline}%
10 {\@fxttextstd{#1}{#2}{#3}}%
11 }
12
13 \def\FXFaceEnvHTMLStyle{font-weight:bold}
14
15 \renewcommand*{\FXEnvLayoutPlainBegin}[2]{%
16 \BlockClass[\FXFaceEnvHTMLStyle]{fixmebold}
17 \ignorespaces#2 \fxnotename{#1}: \ignorespaces}
18
19 \renewcommand*{\FXEnvLayoutPlainEnd}[2]{\endBlockClass}
20
21 \renewcommand*{\FXEnvLayoutSignatureBegin}[2]{%
22 \BlockClass[\FXFaceEnvHTMLStyle]{fixmebold}
23 \fxnotename{#1}: \ignorespaces}
24
25 \renewcommand*{\FXEnvLayoutSignatureEnd}[2]{\@fxsignature{#2}\endBlockClass}
26
27 \def\FXFaceSignatureHTMLStyle{font-style:italic}
28
29 \DeclareRobustCommand*\@fxsignature[1]{%
30 \ifthenelse{\equal{#1}{}}{%
31 {}%
32 { -- {\InlineClass[\FXFaceSignatureHTMLStyle]{fixmesignature}{#1}}}%
33 }
```



---

```

33 }
34
35
36 \def\FXFaceTargetHTMLStyle{font-style:italic}
37
38 \renewcommand\FXTargetLayoutPlain[2]{%
39   \InlineClass[\FXFaceTargetHTMLStyle]{fixmetarget}{#2}%
40 }

```

---

File 78 **lwarp-fixmetodonotes.sty**

§ 165     Package **fixmetodonotes**

*(Emulates or patches code by GIOELE BARABUCCI.)*

Pkg    fixmetodonotes    fixmetodonotes is patched for use by lwarp.

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{fixmetodonotes}

2 \renewcommand{\NOTES@addtolist}[2]{%
3   \refstepcounter{NOTES@note}%
4   \phantomsection% REMOVED
5   \addcontentsline{notes}{NOTES@note}{%
6     \protect\numberline{\theNOTES@note}{\#1: {#2}}%
7   }%
8 }
9
10 \renewcommand{\NOTES@marker}[2]{\fbox{%
11   \textcolor{#2}{% WAS \color
12     \textbf{#1}}%
13   }}
14
15 \renewcommand{\NOTES@colorline}[2]{%
16   \bgroup%
17   \ULon{\LWR@backgroundcolor{#1}{#2}}%
18 }

```

---

File 79 **lwarp-flafter.sty**

§ 166     Package **flafter**

Pkg    flafter    flafter is ignored.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{flafter}
2 \providecommand\fl@trace[1]{%

```

---

File 80 **lwarp-float.sty**

§ 167     Package **float and \newfloat**

(Emulates or patches code by ANSELM LINGNAU.)

Pkg float float is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{float}[2016/03/04]

See section 67.2 for the \listof command.

\newfloat {<1: type>} {<2: placement>} {<3: ext>} [<4: within>]

Emulates the \newfloat command from the float package.

“placement” is ignored.

```
2 \NewDocumentCommand{\newfloat}{m m m o}{%
3 \IfValueTF{#4}{%
4 {\DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}}%
5 {\DeclareFloatingEnvironment[fileext=#3]{#1}}%
6 }
```

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` \*  $\langle style \rangle$

Float styles are ignored.

```
17 \NewDocumentCommand{\restylefloat}{s m}{%
18 }
```

File 81 **lwarp-floatflt.sty**

§ 168 Package **floatflt**

*(Emulates or patches code by MATS DAHLGREN.)*

Pkg floatflt Emulated.

for HTML output: Discard all options for lwarp-floatflt:

```
1 \LWR@ProvidesPackageDrop{floatflt}
```

Env  $[\langle \rangle]$  offset  $\langle type \rangle \langle width \rangle$  Borrowed from the lwarp version of keyfloat:

```
2 \NewDocumentEnvironment{KFLTfloatflt@marginfloat}{0{-1.2ex} m m}
3 {%
4 \setlength{\LWR@templengthone}{#3}%
5 \LWR@BlockClassWP{%
6   float:right; %
7   width:\LWR@printlength{\LWR@templengthone}; % extra space
8   margin:10pt%
9 }{%
10  width:\LWR@printlength{\LWR@templengthone}%
11 }%
12 {marginblock}%
13 \captionsetup{type=#2}%
14 }
15 {%
16 \endLWR@BlockClassWP%
17 }
```

Env floatingfigure  $[\langle placement \rangle] \langle width \rangle$

```
18 \DeclareDocumentEnvironment{floatingfigure}{o m}
19 {\begin{KFLTfloatflt@marginfloat}{figure}{#2}}
20 {\end{KFLTfloatflt@marginfloat}}
```

```

Env floatingtable  [(<placement>)]

21 \DeclareDocumentEnvironment{floatingtable}{o}
22   {\begin{KFLTfloatflt@marginfloat}{table}{1.5in}}
23   {\end{KFLTfloatflt@marginfloat}}

```

---

File 82 **lwarp-floatpag.sty**

§ 169 Package **floatpag**

*(Emulates or patches code by VYTAS STATULEVIČIUS AND SIGITAS TOLUŠIS.)*

Pkg floatpag Emulated.

for HTML output: Discard all options for lwarp-floatpag:

```

1 \LWR@ProvidesPackageDrop{floatpag}

2 \newcommand*{\floatpagestyle}[1]{ }
3 \newcommand*{\rotfloatpagestyle}[1]{ }
4 \newcommand*{\thisfloatpagestyle}[1]{ }

```

---

File 83 **lwarp-floatrow.sty**

§ 170 Package **floatrow**

*(Emulates or patches code by OLGA LAPKO.)*

Pkg floatrow floatrow is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{floatrow}

⚠ misplaced alignment alignment tab character & Use \StartDefiningTabulars and \EndDefiningTabulars before and after defining macros using \ttabbox with a tabular inside. See section 8.9.

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

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

- Use fixed lengths. lwarp will scale the HTML lengths appropriately.

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

After everything has loaded, remember whether subcaption was loaded. If not, it is assumed that subfig is used instead:

```

2 \newbool{LWR@subcaptionloaded}
3
4 \AtBeginDocument{
5 \ifpackageloaded{subcaption}
6 {\booltrue{LWR@subcaptionloaded}}
7 {\boolfalse{LWR@subcaptionloaded}}
8 }

```

`\floatbox` [*<1 preamble>*] [*<2 captype>*] [*<3 width>*] [*<4 height>*] [*<5 vert pos>*] [*<6 caption>*] [*<7 object>*]

Only parameters for captype, width, caption, and object are used.

`LWR@insubfloatrow` is true if inside a `subfloatrow` environment.

There are two actions, depending on the use of subcaption or subfig.

```

9 \NewDocumentCommand{\floatbox}{o m o o o +m +m}{%
10 \ifbool{LWR@subcaptionloaded}%
11 {% subcaption

```

For subcaption:

```

12 \ifbool{LWR@insubfloatrow}%
13 {% subcaption in a subfloatrow

```

`subfigure` and `subtable` environments take width as an argument.

```

14 \IfValueTF{#3}%
15 {\@nameuse{sub#2}{#3}}%
16 {\@nameuse{sub#2}{\linewidth}}%
17 }% subcaption in a subfloatrow
18 {% subcaption not in subfloatrow

```

`figure` and `table` environments do not take a width argument.

```

19 \@nameuse{#2}%
20 }% subcaption not in subfloatrow
21 #6
22
23 #7

```

End the environments:

```

24 \ifbool{LWR@insubfloatrow}%
25 {\@nameuse{endsub#2}}%
26 {\@nameuse{end#2}}%
27}% subcaption
28{% assume subfig

```

For subfig:

```

29\ifbool{LWR@insubfloatrow}%
30{% subfig in a subfloatrow

```

`\subfloat` is a macro, not an environment.

Package subfig's `\subfloat` command takes an optional argument which is the caption, but `\floatbox` argument #6 contains commands to create the caption and label, not the caption itself. Thus, `\caption` is temporarily disabled to return its own argument without braces.

```

31 \begingroup
32 \let\caption\@firstofone
33 \subfloat[#6]{#7}
34 \endgroup
35}% subfig in a subfloatrow
36{% subfig package, but not a subfig

```

`figure` and `table` are environments:

```

37 \@nameuse{#2}
38 #6
39
40 #7
41 \@nameuse{end#2}
42}% subfig package, but not a subfig
43}% assume subfig
44 }

```

Not used:

```

45 \newcommand*{\nocapbeside}{}
46 \newcommand*{\capbeside}{}
47 \newcommand*{\captop}{}
48 \newlength{\FBwidth}
49 \setlength{\FBwidth}{.3\linewidth}
50 \newlength{\FBheight}
51 \setlength{\FBheight}{2in}
52 \newcommand*{\useFCwidth}{}
53 \newcommand{\floatsetup}[2][{}]{
54 \newcommand{\thisfloatsetup}[1]{
55 \newcommand{\clearfloatsetup}[1]{
56 \newcommand*{\killfloatstyle}{}

```

`\newfloatcommand`     $\{\langle 1 \text{ command} \rangle\} \{\langle 2 \text{ captype} \rangle\} [\langle 3 \text{ preamble} \rangle] [\langle 4 \text{ default width} \rangle]$

Preamble and default width are ignored.

```
57 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
58 \@namedef{#1}{
59 \floatbox{#2}
60 }
61 }
```

`\renewfloatcommand`     $\{\langle 1 \text{ command} \rangle\} \{\langle 2 \text{ captype} \rangle\} [\langle 3 \text{ preamble} \rangle] [\langle 4 \text{ default width} \rangle]$

Preamble and default width are ignored.

```
62 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
63 \@namedef{#1}{%
64 \floatbox{#2}
65 }
66 }
```

`\ffigbox`     $[\langle width \rangle] [\langle height \rangle] [\langle vposn \rangle] \{\langle caption commands \rangle\} \{\langle contents \rangle\}$

```
67 \newfloatcommand{ffigbox}{figure}[\nocapbeside] []
```

`\ttabbox`     $[\langle width \rangle] [\langle height \rangle] [\langle vposn \rangle] \{\langle caption commands \rangle\} \{\langle contents \rangle\}$

```
68 \newfloatcommand{ttabbox}{table}[\capttop] [FBwidth]
```

`\fcapside`     $[\langle width \rangle] [\langle height \rangle] [\langle vposn \rangle] \{\langle caption commands \rangle\} \{\langle contents \rangle\}$

```
69 \newfloatcommand{fcapside}{figure}[\capbeside] []
```

Env    `floatrow`     $[\langle numfloats \rangle]$

The row of floats is placed into a `<div>` of class `floatrow`.

```
70 \newenvironment*{floatrow}[1] [2]
71 {
72 \LWR@forcenewpage
73 \BlockClass{floatrow}
```

While inside the `floatrow`, divide the `\linewidth` by the number of floats.

```
74 \booltrue{LWR@infloatrow}
75 \setlength{\linewidth}{6in/#1}
76 }
77 {
78 \boolfalse{LWR@infloatrow}
79 \endBlockClass
80 }
```

Keys for \DeclareNewFloatType:

```

81 \newcommand*{\LWR@frowkeyplacement}{}
82 \newcommand*{\LWR@frowkeyname}{}
83 \newcommand*{\LWR@frowkeyfileext}{}
84 \newcommand*{\LWR@frowkeywithin}{}
85 \newcommand*{\LWR@frowkeycapstyle}{}
86
87 \define@key{frowkeys}{placement}{}%
88 \define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}%
89 \define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}%
90 \define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}%
91 \define@key{frowkeys}{relatedcapstyle}{}%

```

\DeclareNewFloatType {<type>} {<options>}

Use \listof{type}{Title} to print a list of the floats.

```

92 \newcommand*{\DeclareNewFloatType}[2]{%

```

Reset key values:

```

93 \renewcommand*{\LWR@frowkeyplacement}{}%
94 \renewcommand*{\LWR@frowkeyname}{}%
95 \renewcommand*{\LWR@frowkeyfileext}{}%
96 \renewcommand*{\LWR@frowkeywithin}{}%
97 \renewcommand*{\LWR@frowkeycapstyle}{}%

```

Read new key values:

```

98 \LWR@traceinfo{about to setkeys frowkeys}%
99 \setkeys{frowkeys}{#2}%
100 \LWR@traceinfo{finished setkeys frowkeys}%

```

Create a new float with optional [within]:

```

101 \ifthenelse{\equal{\LWR@frowkeywithin}{}}{%
102 {%
103   \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\ %
104     \LWR@frowkeyfileext}%
105   \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}%
106 }%
107 {%
108   \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\ %
109     \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
110   \newfloat{#1}{\LWR@frowkeyplacement}%
111     {\LWR@frowkeyfileext}[\LWR@frowkeywithin}%
112   \LWR@traceinfo{finished newfloat #1}%
113 }%

```

Rename the float if a name was given:

```

114 \ifthenelse{\equal{\LWR@frowkeyname}{}}{%

```



```

115 {}%
116 {\floatname{#1}{\LWR@frowkeyname}}%
117 }

```

Not used:

```

118 \newcommand{\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`     $\{\langle text \rangle\}$

To be used inside a minipage or parbox.

```

127 \newcommand{\RawCaption}[1]{#1}

```

`\floatfoot`     $\{\langle text \rangle\}$

Places additional text inside a float, inside a CSS `<div>` of class `floatfoot`.

```

128 \NewDocumentCommand{\floatfoot}{s +m}{%
129 \begin{BlockClass}{floatfoot}
130 #2
131 \end{BlockClass}
132 }

```

Used to compute `\linewidth`.

```

133 \newbool{LWR@insubfloatrow}
134 \boolfalse{LWR@insubfloatrow}

```

Env    `subfloatrow`     $[\langle num\_floats \rangle]$

```

135 \newenvironment*{subfloatrow}[1][2]
136 {

```

The row of floats is placed into a `<div>` of class `floatrow`:

```

137 \LWR@forcenewpage
138 \BlockClass{floatrow}

```

While inside the `floatrow`, `LWR@insubfloatrow` is set true, which tells `\floatbox` to use `\subfigure` or `\subtable`.

```

139 \begingroup

```

---

```

140 \booltrue{LWR@insubfloatrow}
141 }
142 {
143 \endgroup
144 \endBlockClass
145 \boolfalse{LWR@insubfloatrow}
146 }

```

---

#### File 84 **lwarp-fltrace.sty**

### § 171 Package **fltrace**

Pkg fltrace fltrace is ignored.

**for HTML output:** 1 \LWR@ProvidesPackageDrop{fltrace}

```

2 \def\tracefloats{}
3 \def\tracefloatsoff{}
4 \def\tracefloatvals{}

```

---

#### File 85 **lwarp-flushend.sty**

### § 172 Package **flushend**

*(Emulates or patches code by SIGITAS TOLUŠIS.)*

Pkg flushend Emulated.

**for HTML output:** Discard all options for lwarp-flushend:

```

1 \LWR@ProvidesPackageDrop{flushend}
2 % \end{macrocode}
3 %
4 % \begin{macrocode}
5 \newcommand*{\flushend}{}
6 \newcommand*{\raggedend}{}
7 \newcommand*{\flushcolsend}{}
8 \newcommand*{\raggedcolsend}{}
9 \newcommand*{\atColsBreak}[1]{}
10 \newcommand*{\atColsEnd}[1]{}
11 \newcommand*{\showcolsendrule}{}

```

---

File 86 **lwarp-fncychap.sty**

§ 173     Package **fncychap**

*(Emulates or patches code by Ulf A. Lindgren.)*

Pkg    fncychap    fncychap is emulated.

for HTML output:    Discard all options for lwarp-fncychap:

```

1 \LWR@ProvidesPackageDrop{fncychap}

2 \def\mghrulefill#1{}
3 \def\ChNameLowerCase{}
4 \def\ChNameUpperCase{}
5 \def\ChNameAsIs{}
6 \def\ChTitleLowerCase{}
7 \def\ChTitleUpperCase{}
8 \def\ChTitleAsIs{}
9 \newcommand{\ChRuleWidth}[1]{}
10 \newcommand{\ChNameVar}[1]{}
11 \newcommand{\ChNumVar}[1]{}
12 \newcommand{\ChTitleVar}[1]{}
13 \newcommand{\TheAlphaChapter}{}
14 \newcommand{\DOCH}{}
15 \newcommand{\DOTI}[1]{}
16 \newcommand{\DOTIS}[1]{}
17 \newlength{\mylen}
18 \newlength{\myhi}
19 \newlength{\px}
20 \newlength{\py}
21 \newlength{\pyy}
22 \newlength{\pxx}
23 \newlength{\RW}
24 \newcommand{\FmN}[1]{#1}
25 \newcommand{\FmTi}[1]{#1}

```

---

File 87 **lwarp-fnlineno.sty**

§ 174     Package **fnlineno**

Pkg    fnlineno    fnlineno is ignored.

---

**for HTML output:**    1 \LWR@ProvidesPackageDrop{fnlineno}

---

File 88    **lwarp-fnpos.sty**

§ 175    Package    **fnpos**

*(Emulates or patches code by HIROSHI NAKASHIMA.)*

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 89    **lwarp-fontenc.sty**

§ 176    Package    **fontenc**

Pkg    fontenc    If using pdf<sub>TEX</sub>, lwarp used to require fontspec be loaded before lwarp, but now lwarp itself loads \fontspec with T1 encoding, which lwarp requires. fontspec is now allowed to be loaded with another encoding after lwarp.

lwarp-fontenc is no longer necessary, but is still provided to overwrite older versions.

**for HTML output:**    1 \LWR@ProvidesPackagePass{fontenc}

---

File 90    **lwarp-fontspec.sty**

§ 177    Package    **fontspec**

Pkg    fontspec    Error if fontspec is loaded after lwarp.

Discard all options for lwarp-fontspec:

**for HTML output:**    1 \LWR@ProvidesPackageDrop{fontspec}

2 \LWR@loadbefore{fontspec}

---

File 91 **lwarp-footmisc.sty**

§ 178      Package **footmisc**

*(Emulates or patches code by ROBIN FAIRBAIRNS.)*

Pkg    footmisc    footmisc is emulated.

```
1 \LWR@ProvidesPackageDrop{footmisc}
```

Some nullified commands:

```
2 \newcommand{\footnotelayout}{}
3 \newcommand{\setfnsymbol}[1]{}
4 \NewDocumentCommand{\DefineFNSymbols}{s m o m}{}
5
6 \newdimen\footnotemargin
7 \footnotemargin1.8em\relax
8
9 \newcommand*\hangfootparskip{0.5\baselineskip}
10 \newcommand*\hangfootparindent{0em}%
11
12 \let\pagefootnoterule\footnoterule
13 \let\mpfootnoterule\footnoterule
14 \def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}
15
16 \providecommand*\multiplefootnotemarker}{3sp}
17 \providecommand*\multfootsep}{,}
```

Using cleveref:

```
18 \providecommand*\footref}[1]{\labelcref{#1}}
```

The following work as-is:

```
19 \newcommand\mpfootnotemark{%
20   \@ifnextchar[%
21     \xmpfootnotemark
22     {%
23       \stepcounter\@mpfn
24       \protected@xdef\@thefnmark{\thempfn}%
25       \@footnotemark
26     }%
27 }
```

---

```

28 \def\xmpfootnotemark[#1]{%
29   \begingroup
30     \csname c@\mpfn\endcsname #1\relax
31     \unrestored@protected@xdef\thefnmark{\thempfn}%
32   \endgroup
33   \@footnotemark
34 }

```

---

File 92 **lwarp-footnote.sty**

§ 179     Package **footnote**

*(Emulates or patches code by MARK WOODING.)*

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 \def\fn@endnote{%
10  \LWR@htmltagc{/LWR@tagregularparagraph}%
11  \LWR@orignewline%
12 }

```

Removed print-version formatting:

```

13 \def\fn@startfntext{%
14   \setbox\z@\vbox\bgroup%
15     \fn@startnote%
16     \fn@prefntext%
17     \ignorespaces%
18 }

```

Removed print-version formatting, added closing paragraph tag:

```

19 \def\fn@endfntext{%
20   \LWR@htmltagc{/LWR@tagregularparagraph}%
21   \LWR@orignewline%
22   \fn@postfntext%

```

---

```

23 \egroup%
24 \beginingroup%
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 93 **lwarp-footnotehyper.sty**

§ 180    Package **footnotehyper**

Pkg    footnotehyper    footnotehyper is a hyperref-safe version of footnote. For lwarp, footnotehyper is emulated.

**for HTML output:**    Discard all options for lwarp-footnotehyper:

```

1 \RequirePackage{footnote}
2 \LWR@ProvidesPackageDrop{footnotehyper}

```

---

File 94 **lwarp-footnpag.sty**

§ 181    Package **footnpag**

Pkg    footnpag    footnpag is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{footnpag}

---

File 95 **lwarp-framed.sty**

§ 182    Package **framed**

*(Emulates or patches code by DONALD ARSENEAU.)*

Pkg    framed    framed is supported and patched by lwarp.

for HTML output: Accept all options for lwarp-framed:

```

1 \LWR@ProvidesPackagePass{framed}
2 \RequirePackage{xcolor}% for \convertcolorspec

3
4 \renewenvironment{framed}{%
5 \LWR@forcenewpage
6 \BlockClass{framed}%
7 }
8 {\endBlockClass}
9
10 \renewenvironment{oframed}{%
11 \LWR@forcenewpage
12 \BlockClass{framed}%
13 }
14 {\endBlockClass}
15
16
17 \renewenvironment{shaded}{%
18 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
19 \LWR@forcenewpage
20 \BlockClass[background: \#\LWR@tempcolor]{shaded}%
21 }
22 {\endBlockClass}
23
24 \renewenvironment{shaded*}{%
25 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
26 \LWR@forcenewpage
27 \BlockClass[background: \#\LWR@tempcolor]{shaded}%
28 }
29 {\endBlockClass}
30
31
32 \renewenvironment{leftbar}{%
33 \LWR@forcenewpage
34   \BlockClass{framedleftbar}
35   \def\FrameCommand{}%
36   \MakeFramed {}
37 }%
38 {\endMakeFramed\endBlockClass}
39
40
41 \renewenvironment{snugshade}{%
42 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
43 \LWR@forcenewpage
44 \BlockClass[background: \#\LWR@tempcolor]{snugframed}%
45 }
46 {\endBlockClass}
47

```



```

48 \renewenvironment{snugshade*}{%
49 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
50 \LWR@forcenewpage
51 \BlockClass[background: \#\LWR@tempcolor]{snugframed}%
52 }
53 {\endBlockClass}
54
55 \let\oframed\framed
56 \let\endoframed\endframed
57
58
59 \RenewEnviron{titled-frame}[1]{%
60 \CustomFBox{#1}{}{0pt}{0pt}{0pt}{0pt}{\BODY}
61 }

\CustomFBox {<toptitle>} {<bottitle>} {<thicknesstop>} {<bottom>} {<left>} {<right>}
{<text contents>}

62 \renewcommand{\CustomFBox}[7]{%
63 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
64 \LWR@forcenewpage
65 \begin{BlockClass}[border: 3px solid \#\LWR@tempcolor]{framed}%
66 \ifthenelse{\isempty{#1}}{}{% not empty
67   \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
68   \textcolor{TFTitleColor}{\textbf{#1}}%
69   \end{BlockClass}
70 }% not empty
71
72 #7
73
74 \ifthenelse{\isempty{#2}}{}{% not empty
75   \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
76   \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
77   \textcolor{TFTitleColor}{\textbf{#2}}%
78   \end{BlockClass}
79 }% not empty
80 \end{BlockClass}
81 }

\TitleBarFrame [<marker>] {<title>} {<contents>}

82 \renewcommand\TitleBarFrame[3][ ]{
83 \CustomFBox
84   {#2}{}%
85   \fboxrule\fboxrule\fboxrule\fboxrule
86   {#3}%
87 }

88 \renewcommand{\TF@Title}[1]{#1}

```

---

MakeFramed {\<settings>}

```

89 \let\MakeFramed\relax
90 \let\endMakeFramed\relax
91
92 \NewEnviron{MakeFramed}[1]{%
93 \FrameCommand{\begin{minipage}{\linewidth}\BODY\end{minipage}}%
94 }
```

\fb@put@frame {\<frame cmd no split>} {\<frame cmd split>}

```

95 \renewcommand*\fb@put@frame}[2]{%
96 \relax%
97 \@tempboxa%
98 }
```

---

File 96 **lwarp-ftnright.sty**

§ 183    Package **ftnright**

Pkg    ftnright    ftnright is ignored.

for HTML output:    Discard all options for lwarp-ftnright:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

---

File 97 **lwarp-fullpage.sty**

§ 184    Package **fullpage**

Pkg    fullpage    fullpage is ignored.

for HTML output:    Discard all options for lwarp-fullpage:

```
1 \LWR@ProvidesPackageDrop{fullpage}
```

---

File 98 **lwarp-fullwidth.sty**

§ 185    Package **fullwidth**

(Emulates or patches code by MARCO DANIEL.)

Pkg fullwidth fullwidth is emulated.

A minipage is used, of no HTML width.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{fullwidth}

2 \newenvironment*{fullwidth}[1][1]{%
3 \minipagefullwidth%
4 \minipage{\linewidth}%
5 }
6 {%
7 \endminipage%
8 }
```

---

File 99 **lwarp-fwlw.sty**

§ 186 Package **fwlw**

Pkg fwlw fwlw is ignored.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{fwlw}

2 \newbox\FirstWordBox      \global\setbox\FirstWordBox\hbox{}
3 \newbox\NextWordBox       \global\setbox\NextWordBox\hbox{}
4 \newbox\LastWordBox       \global\setbox\LastWordBox\hbox{}
5 \def\ps@fwlwhead{}
6 \def\ps@NextWordFoot{}
```

---

File 100 **lwarp-geometry.sty**

§ 187 Package **geometry**

*(Emulates or patches code by HIDEO UMEKI.)*

Pkg geometry geometry is preloaded by lwarp, but must be nullified as seen by the user's source code.

**for HTML output:** Discard all options for lwarp-geometry:

```

1 \LWR@ProvidesPackageDrop{geometry}

2 \renewcommand*{\geometry}[1]{}
3 \renewcommand*{\newgeometry}[1]{}

```

---

```

4 \renewcommand*{\restoregeometry}{ }
5 \renewcommand*{\savegeometry}[1]{ }
6 \renewcommand*{\loadgeometry}[1]{ }

```

---

File 101 **lwarp-glossaries.sty**

§ 188      Package **glossaries**

*(Emulates or patches code by NICOLA L.C. TALBOT.)*

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

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

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

[placement and toc options](#)      The glossaries may be placed in a numbered or unnumbered section, given a toc entry, and placed inline or on their own HTML page:

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

```

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

```

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

```

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

```

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

```

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

```

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

Opt lwarpmk      `printglossary`      lwarpmk has the commands `lwarpmk printglossary` and `lwarpmk htmlglossary`  
 Opt lwarpmk      `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   \ifdefempty\@glossarysecstar
9     {%
10      \csname\@glossarysec\endcsname{#2}%
11    }%
12  {%

```

In the original, the TOC entry was made before the section, thus linking to the phantomsection in the printed version, but for HTML this caused the link to point to the page before the glossaries. Here, the TOC entry is made after the section is created:

```

13      \csname\@glossarysec\endcsname*{#2}%
14      \@gls@toc{#1}{\@glossarysec}% Moved after the previous line.
15    }%
16    \@glossaryseclabel
17  }

```

---

File 102 **lwarp-graphics.sty**

§ 189     Package **graphics**

*(Emulates or patches code by D. P. CARLISLE.)*

Pkg    graphics    graphics is emulated.

**for HTML output:**    1 \LWR@ProvidesPackagePass{graphics}

§ 189.1    **Graphics extensions**

\DeclareGraphicsExtensions     $\{\langle list \rangle\}$

\AtBeginDocument allow SVG files instead of PDF:

```

2 \AtBeginDocument{
3 \DeclareGraphicsExtensions{.svg,.SVG,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}
4 \DeclareGraphicsRule{.svg}{svg}{.svg}{}
5 \DeclareGraphicsRule{.SVG}{svg}{.SVG}{}

```

```
6 }
```

Inside a `lateximage`, allow PDF instead of SVG:

```
7 \appto\LWR@restoreorigformatting{%
8 \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
9 }
```

## § 189.2 Length conversions and graphics options



whitespace

A scaled image in  $\TeX$  by default takes only as much space on the page as it requires, but HTML browsers use as much space as the original unscaled image would have taken, with the scaled image over- or under-flowing the area.

Used to store the user's selected dimensions and HTML class.

The class defaults to “`inlineimage`” unless changed by a `class=xyx` option.

```
10 \newlength{\LWR@igwidth}
11 \newlength{\LWR@igheight}
12 \newcommand*{\LWR@igwidthstyle}{%}
13 \newcommand*{\LWR@igheightstyle}{%}
14 \newcommand*{\LWR@igorigin}{%}
15 \newcommand*{\LWR@igangle}{%}
16 \newcommand*{\LWR@igxscale}{1}
17 \newcommand*{\LWR@igyscale}{1}
18 \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:

```
19 \define@key{igraph}{width}{%
20 \setlength{\LWR@igwidth}{#1}%
21 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}{%
22 {%
```

Default to use the converted fixed length given:

```
23 \renewcommand*{\LWR@igwidthstyle}{width:\LWR@printlength{\LWR@igwidth}}%
```

If ex or em dimensions were given, use those instead:

```
24 \IfEndWith{#1}{ex}%
25 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes ex
26 {}% not ex
27 \IfEndWith{#1}{em}%
```

```

28   {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes em
29   {}% not em
30   \IfEndWith{#1}{\}%
31   {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes percent
32   {}% not percent
33   \IfEndWith{#1}{px}%
34   {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
35   {}% not px
36 }{}% end of length > Opt
37 }

```

If an optional height was given, set an HTML style:

```

38 \define@key{igraph}{height}{%
39 \setlength{\LWR@igheight}{#1}%
40 \ifthenelse{\lengthtest{\LWR@igheight > Opt}}%
41 {%

```

Default to use the converted fixed length given:

```

42   \renewcommand*{\LWR@igheightstyle}{%
43   height:\LWR@printlength{\LWR@igheight} % extra space
44   }%

```

If ex or em dimensions were given, use those instead:

```

45   \IfEndWith{#1}{ex}%
46   {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
47   {}% not ex
48   \IfEndWith{#1}{em}%
49   {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
50   {}% not em
51   \IfEndWith{#1}{\}%
52   {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
53   {}% not percent
54   \IfEndWith{#1}{px}%
55   {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
56   {}% not px
57 }{}% end of length > Opt
58 }

```

Handle origin key:

```

59 \define@key{igraph}{origin}{%
60 \renewcommand*{\LWR@igorigin}{#1}%
61 }

```

Handle angle key:

```
62 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}
```

Handle class key:

```
63 \define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}
64
```

It appears that graphicx does not have separate keys for xscale and yscale. scale adjusts both at the same time.

```
65 \define@key{igraph}{scale}{%
66 \renewcommand*{\LWR@igxscale}{#1}%
67 \renewcommand*{\LWR@igyyscale}{#1}}
```

Numerous ignored keys:

```
68 \define@key{igraph}{bb}{}
69 \define@key{igraph}{bblx}{}
70 \define@key{igraph}{bbly}{}
71 \define@key{igraph}{bburx}{}
72 \define@key{igraph}{bbury}{}
73 \define@key{igraph}{natwidth}{}
74 \define@key{igraph}{natheight}{}
75 \define@key{igraph}{hiresbb}{}
76 \define@key{igraph}{viewport}{}
77 \define@key{igraph}{trim}{}
78 \define@key{igraph}{totalheight}{}
79 \define@key{igraph}{keepaspectratio}{}
80 \define@key{igraph}{clip}{}
81 \define@key{igraph}{draft}{}
82 \define@key{igraph}{type}{}
83 \define@key{igraph}{ext}{}
84 \define@key{igraph}{read}{}
85 \define@key{igraph}{command}{}

```

### § 189.3 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.

```
86 \newcommand*{\LWR@rotstyle}[2]{%
87 #1transform:rotate(-#2deg);
88 }
```



`\LWR@scalestyle`  $\{\langle prefix \rangle\} \{\langle xscale \rangle\} \{\langle yscale \rangle\}$

Prints the scale style with the given prefix.

`prefix` is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:scale` style.

```
89 \newcommand*\LWR@scalestyle[3]{%
90   #1transform:scale(#2,#3);
91 }
```

#### § 189.4 `\includegraphics`

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```
92 \newbool{LWR@infloatrow}
93 \boolfalse{LWR@infloatrow}
```

`\LWR@opacity` For HTML, used only for `\includegraphics`.

`\LWR@opacity` may be set by the transparent package.

```
94 \def\LWR@opacity{1}
```

`\LWR@imagesizebox` Used to determine the actual image size if needed.

```
95 \newsavebox{\LWR@imagesizebox}
```

`\Gin@setfile` Sets the parsed filename.

```
96 \let\LWR@origGin@setfile\Gin@setfile
```

Key `Gin` class CSS class for the image.

Define the new class key for the print-mode version of `\includegraphics`, which is enabled inside a `lateximage`.

```
97 \AtBeginDocument{
98   \define@key{Gin}{class}{}
99 }
```

`\LWR@includegraphicsb` \* [ $\langle 2: options \rangle$ ] [ $\langle 3: options \rangle$ ] [ $\langle 4: filename \rangle$ ]

graphics syntax is `\includegraphics * [ $\langle llx, lly \rangle$ ] [ $\langle urx, ury \rangle$ ] [ $\langle file \rangle$ ]`

graphicx syntax is `\includegraphics [ $\langle key values \rangle$ ] [ $\langle file \rangle$ ]`

If #3 is empty, only one optional argument was given, thus graphicx syntax.

```
100 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
101 {%
102 \LWR@traceinfo{\LWR@includegraphicsb #4}%
```

Start the image tag on a new line, allow PDF output word wrap:

```
103 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute \linewidth, \textwidth, \textheight arguments with a 6x9 inch size until the next \endgroup.

```
104 \begingroup%
105 \ifthenelse{\cnttest{\value{\LWR@minipagedepth}}{=}{0}}{%
106 {%
107   \ifbool{\LWR@infloatrow}%
108   {}
109   {% not in a minipage or a floatrow:
110     \setlength{\linewidth}{6in}%
111     \setlength{\textwidth}{6in}%
112     \setlength{\textheight}{9in}%
113   }%
114 }{}%

115 \begingroup%
116 \renewcommand*{\Gin@setfile}[3]{%
117 \LWR@traceinfo{\Gin@setfile ##3}%
118 \xdef\LWR@parsedfilename{##3}%
119 }%
120 \Gin@include@graphics{\detokenize\expandafter{#4}}%
121 \endgroup%
122 \filename@parse{\LWR@parsedfilename}%
123 \LWR@traceinfo{\LWR@parsedfilename is \LWR@parsedfilename}%
124 % \LWR@sanitize{\LWR@parsedfilename}%
```

For correct em sizing during the width and height conversions:

```
125 \large%
```

Reset some defaults, possibly will be changed below if options were given:

```
126 \setlength{\LWR@igwidth}{0pt}%
127 \setlength{\LWR@igheight}{0pt}%
128 \renewcommand*{\LWR@igwidthstyle}{}%
129 \renewcommand*{\LWR@igheightstyle}{}%
130 \renewcommand*{\LWR@igorigin}{}%
131 \renewcommand*{\LWR@igangle}{}%
```

```

132 \renewcommand*{\LWR@igxscale}{1}%
133 \renewcommand*{\LWR@igyscale}{1}%
134 \renewcommand*{\LWR@igclass}{inlineimage}%

```

If #3 is empty, only one optional argument was given, thus graphicx syntax:

```

135 \IfValueF{#3}{%
136 \IfValueTF{#2}%
137 {\setkeys{igraph}{#2}}%
138 {\setkeys{igraph}{}}%
139 }%

```

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:

```

140 \ifbool{FormatWP}{%
141   \begingroup%
142   \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
143   \define@key{Gin}{angle}{}%
144   \IfBooleanTF{#1}%
145   {% starred
146     \IfValueTF{#3}%
147     {%
148       \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics*{#2}[#3]{#4}}%
149     }%
150     {%
151       \IfValueTF{#2}%
152       {%
153         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics*{#2}{#4}}%
154       }{%
155         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics*{#4}}%
156       }%
157     }%
158   }% starred
159   {% not starred
160     \IfValueTF{#3}%
161     {%
162       \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics[#2][#3]{#4}}%
163     }%
164     {%
165       \IfValueTF{#2}%
166       {%
167         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics[#2]{#4}}%
168       }{%
169         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics{#4}}%
170       }%
171     }%
172   }% not starred
173   \endgroup%

```

```

174 \settowidth{\LWR@igwidth}{\usebox{\LWR@imagesizebox}}%
175 \global\renewcommand*{\LWR@igwidthstyle}{width:\LWR@printlength{\LWR@igwidth}}%
176 \settoheight{\LWR@igheight}{\usebox{\LWR@imagesizebox}}%
177 \global\renewcommand*{\LWR@igheightstyle}{height:\LWR@printlength{\LWR@igheight}}%
178 }{}%

```

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.

```

179 \LWR@traceinfo{\LWR@includegraphicsb: about to create href}%
180 \href{\LWR@parsedfilename}%
181 {% start of href
182 \LWR@traceinfo{\LWR@includegraphicsb: about to LWR@htmltag}%
183 \LWR@htmltag{% start of image tags
184 img src="%
185 \begingroup\@sanitize\LWR@parsedfilename\endgroup%
186 " \LWR@orignewline%

```

Only include a style tag if a width, height, angle, or scale was given:

```

187 \ifthenelse{
188 \NOT\equal{\LWR@igwidthstyle}{} \OR
189 \NOT\equal{\LWR@igheightstyle}{} \OR
190 \NOT\equal{\LWR@igorigin}{} \OR
191 \NOT\equal{\LWR@igangle}{} \OR
192 \NOT\equal{\LWR@igxscale}{1} \OR
193 \NOT\equal{\LWR@igyyscale}{1}
194 }%
195 {\LWR@origtilde{} style="%
196 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{}}%
197 {\LWR@igwidthstyle;}{}%
198 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{}}%
199 {\LWR@igheightstyle;}{}%
200 \ifthenelse{\NOT\equal{\LWR@igorigin}{}}%
201 {\LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{}%
202 \ifthenelse{\NOT\equal{\LWR@igangle}{}}%
203 {%
204 \LWR@rotstyle{-ms-}{\LWR@igangle} % extra space
205 \LWR@rotstyle{-webkit-}{\LWR@igangle} % extra space
206 \LWR@rotstyle{}{\LWR@igangle} %
207 }{}%
208 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%
209 \NOT\equal{\LWR@igyyscale}{1}}%
210 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyyscale} % extra space
211 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyyscale} % extra space
212 \LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyyscale}}{} % extra space

```

```

213 %
214 \ifthenelse{\NOT\equal{\LWR@opacity}{1}}%
215 {opacity:\LWR@opacity; }%
216 {}%
217 %
218 " \LWR@orignewline}{}%

```

Set the class:

```

219 \LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%
220 }% end of image tags
221 }% end of href

```

Return to original page size and font size:

```

222 \endgroup
223 \LWR@traceinfo{\LWR@includegraphicsb done}%
224 }

```

`\includegraphics` [*key=val*] {*filename*}

Handles width and height, converted to fixed width and heights.

The user should always use no file suffix in the document source.

```

225 \AtBeginDocument{
226
227 \LWR@traceinfo{Patching includegraphics.}
228
229 \LetLtxMacro\LWR@originincludegraphics\includegraphics
230
231 \renewcommand*\includegraphics{
232 {%

```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```

233 \LWR@traceinfo{includegraphics}%
234 \LWR@ensuredoingapar%
235 \LWR@includegraphicsb%
236 }% includegraphics
237 }% AtBeginDocument

```

## § 189.5 Boxes

`\LWR@rotboxorigin` Holds the origin key letters.

```
238 \newcommand*\LWR@rotboxorigin{}
```

`\LWR@originname`  $\{\langle letter \rangle\}$

Given one  $\LaTeX$  origin key value, translate into an HTML origin word:

```
239 \newcommand*\LWR@originname[1]{%
240 \ifthenelse{\equal{#1}{t}}{top}{}%
241 \ifthenelse{\equal{#1}{b}}{bottom}{}%
242 \ifthenelse{\equal{#1}{c}}{center}{}%
243 \ifthenelse{\equal{#1}{l}}{left}{}%
244 \ifthenelse{\equal{#1}{r}}{right}{}%
245 }
```

`\LWR@originnames`  $\{\langle letters \rangle\}$

Given one- or two-letter  $\LaTeX$  origin key values, translate into HTML origin words:

```
246 \newcommand*\LWR@originnames[1]{%
247 \StrChar{#1}{1}[\LWR@strresult]%
248 \LWR@originname{\LWR@strresult}
249 \StrChar{#1}{2}[\LWR@strresult]%
250 \LWR@originname{\LWR@strresult}
251 }
```

Handle the origin key for `\rotatebox`:

```
252 \define@key{krotbox}{origin}{%
253 \renewcommand*\LWR@rotboxorigin{#1}%
254 }
```

These keys are ignored:

```
255 \define@key{krotbox}{x}{}
256 \define@key{krotbox}{y}{}
257 \define@key{krotbox}{units}{}

```

`\rotatebox`  $[\langle keyval list \rangle] \{\langle angle \rangle\} \{\langle text \rangle\}$

```
258 \LetLtxMacro\LWR@origrotatebox\rotatebox
259
260 \AtBeginDocument{
261 \RenewDocumentCommand{\rotatebox}{O{} m +m}{%
```

Reset the origin to “none-given”:

```
262 \renewcommand*{\LWR@rotboxorigin}{}
```

Process the optional keys, which may set \LWR@rotateboxorigin:

```
263 \setkeys{krotbox}{#1}%
```

Select inline-block so that HTML will transform this span:

```
264 \LWR@htmltagc{span style="display: inline-block; %
```

If an origin was given, translate and print the origin information:

```
265 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}{%
266 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}}%
```

Print the rotation information:

```
267 \LWR@rotstyle{-ms-}{#2} % extra space
268 \LWR@rotstyle{-webkit-}{#2} % extra space
269 \LWR@rotstyle{}{#2} % extra space
270 ")\LWR@orignewline%
```

Print the text to be rotated:

```
271 \begin{LWR@nestspan}%
272 #3%
```

Close the span:

```
273 \LWR@htmltagc{/span}%
274 \end{LWR@nestspan}%
275 }
276 }% AtBeginDocument
```

`\scalebox`     $\{\langle h\text{-scale}\rangle\} [\langle v\text{-scale}\rangle] \{\langle text\rangle\}$

```
277 \LetLtxMacro\LWR@origscalebox\scalebox
278
279 \AtBeginDocument{
280 \RenewDocumentCommand{\scalebox}{m o m}{%
```

Select inline-block so that HTML will transform this span:

```
281 \LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
282 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} % extra space
283 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} % extra space
284 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} % extra space
285 "%
```

Print the text to be scaled:

```
286 \begin{LWR@nestspan}%
287 #3%
```

Close the span:

```
288 \LWR@htmltagc{/span}%
289 \end{LWR@nestspan}%
290 }
291 }% AtBeginDocument
```

`\reflectbox`     $\{\langle text \rangle\}$

```
292 \let\LWR@origreflectbox\reflectbox
293
294 \AtBeginDocument{
295 \renewcommand{\reflectbox}[1]{\scalebox{-1}[1]{#1}}
296 }
```

`\resizebox`     $\{\langle h-length \rangle\} \{\langle v-length \rangle\} \{\langle text \rangle\}$

Simply prints its text argument.

```
297 \LetLtxMacro\LWR@origresizebox\resizebox
298
299 \AtBeginDocument{
300 \renewcommand{\resizebox}[3]{#3}
301 }
```

---

File 103    **lwarp-graphicx.sty**

§ 190    Package    **graphicx**

Pkg    **graphicx**    graphicx is emulated.



graphicx loads graphics, which also loads lwarp-graphics, which remembers the original graphics definitions for use inside a lateximage, and then patches them \AtBeginDocument for HTML output.


lwarp-graphics handles the syntax of either graphics or graphicx.

**for HTML output:** `1 \LWR@ProvidesPackagePass{graphicx}`

---

File 104 **lwarp-grffile.sty**

§ 191 Package **grffile**

 **matching PDF and SVG** 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.

lwarp-grffile now exists as a placeholder since grffile used to be emulated by lwarp, and thus older versions of lwarp-grffile may exist and should be overwritten by this newer version.

**for HTML output:** `1 \LWR@ProvidesPackagePass{grffile}`

---

File 105 **lwarp-hang.sty**

§ 192 Package **hang**

*(Emulates or patches code by ANDREAS NOLDA.)*

Pkg hang hang is emulated.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{hang}

2 \newlength{\hangingindent}
3 \setlength{\hangingindent}{1em}
4 \newlength{\hangingleftmargin}
5 \setlength{\hangingleftmargin}{0em}
6
7 \newcommand*{\LWR@findhangingleftmargin}{%
8 \setlength{\LWR@templengthone}{\hangingleftmargin}%
9 \addtolength{\LWR@templengthone}{\hangingindent}%
10 }
11
12 \newenvironment{hangingpar}
13 {
```

```

14 \LWR@findhangingleftmargin%
15 \BlockClass[%
16 \LWR@origmbox{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
17 \LWR@origmbox{text-indent:-\LWR@printlength{\hangingindent}}%
18 ]%
19 {hangingpar}%
20 }
21 {\endBlockClass}
22
23 \newenvironment{hanginglist}
24 {%
25 \renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}%
26 \renewcommand*{\LWR@printopenlist}{%
27 \LWR@findhangingleftmargin%
28 ul style="%
29 \LWR@origmbox{list-style-type:none;} % extra space
30 \LWR@origmbox{%
31 margin-left:\LWR@printlength{\LWR@templengthone}%
32 } ; % extra space
33 \LWR@origmbox{%
34 text-indent:-\LWR@printlength{\hangingindent}%
35 }%
36 "%
37 }%
38 \let\item\LWR@itemizeitem%
39 \list{}{}%
40 }
41 {\endlist}
42
43 \newenvironment{compacthang}
44 {\hanginglist}
45 {\endhanginglist}
46
47 \newlength{\labeledleftmargin}
48 \setlength{\labeledleftmargin}{0em}
49
50 \newenvironment{labeledpar}[2]
51 {%
52 \BlockClass[%
53 \LWR@findhangingleftmargin%
54 \LWR@origmbox{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
55 \LWR@origmbox{text-indent:-\LWR@printlength{\hangingindent}}%
56 ]{labeledpar}#2%
57 }
58 {\endBlockClass}
59
60 \newenvironment{labeledlist}[1]
61 {\hanginglist}
62 {\endhanginglist}
63

```

---

```

64 \newenvironment{compactlabel}[1]
65 {\hanginglist}
66 {\endhanginglist}

```

---

File 106 **lwarp-hanging.sty**

§ 193     Package **hanging**

Pkg    hanging    hanging is emulated.

for HTML output:    1 \LWR@ProvidesPackageDrop{hanging}

```

2 \@ifclassloaded{memoir}{
3 \let\hangpara\relax
4 \let\hangparas\relax
5 \let\endhangparas\relax
6 \let\hangpunct\relax
7 \let\endhangpunct\relax
8 }{}

```

\hangpara    {\langle*indent*\rangle} {\langle*afternum*\rangle}

Use **hangparas** instead.

```

9 \newcommand*{\hangpara}[2]{

```

Env    hangparas    {\langle*indent*\rangle} {\langle*afternum*\rangle}

```

10 \newenvironment*{hangparas}[2]
11 {
12    \BlockClass[
13      \LWR@origmbox{margin-left:\LWR@printlength{#1}} ; %
14      \LWR@origmbox{text-indent:-\LWR@printlength{#1}}%
15    ]%
16    {hangingpar}%
17 }
18 {\endBlockClass}

```

Env    hangpunct

```

19 \newenvironment*{hangpunct}
20 {\BlockClass{hangpunct}}
21 {\endBlockClass}

22 \newcommand{\nhpt}{.}
23 \newcommand{\nhlq}{'}

```

---

```
24 \newcommand{\nhrq}{'}
```

---

File 107 **lwarp-hypcap.sty**

§ 194      Package **hypcap**

Pkg    hypcap    hypcap is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{hypcap}

```

2 \newcommand*{\capstart}{}
3 \newcommand*{\hypcapspace}{}
4 \newcommand*{\hypcapredef}[1]{}
5 \newcommand*{\capstartfalse}{}
6 \newcommand*{\capstarttrue}{}

```

---

File 108 **lwarp-hypdestopt.sty**

§ 195      Package **hypdestopt**

Pkg    hypdestopt    hypdestopt is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{hypdestopt}

---

File 109 **lwarp-hypernat.sty**

§ 196      Package **hypernat**

Pkg    hypernat    hypernat is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{hypernat}

---

File 110 **lwarp-hyperref.sty**

§ 197      Package **hyperref**

*(Emulates or patches code by SEBASTIAN RAHTZ, HEIKO OBERDIEK.)*

Pkg hyperref hyperref is emulated.

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{\hyperimage}{m +m}{%
11\LWR@ensuredoingapar%
12\def\LWR@templink{#1}%
13\@onelevel@sanitize\LWR@templink%
14\LWR@htmltag{img src="\LWR@templink" alt="#2" class="hyperimage"}%
15\LWR@ensuredoingapar%
16}

```

\hyperdef {<1: category>} {<2: name>} {<3: text>}

Creates an HTML anchor to category.name with the given text.

```

17\NewDocumentCommand{\hyperdef}{m m +m}{%
18\LWR@ensuredoingapar%
19\LWR@sublabel{#1.#2}%
20#3%
21}

```

\LWR@hyperrefb {<1: URL>} {<2: category>} {<3: name>} {<4: text>}

Creates an HTML link to URL#category.name with the given text.

```

22\NewDocumentCommand{\LWR@hyperrefb}{m m m +m}{%
23\def\LWR@templink{#1}%
24\@onelevel@sanitize\LWR@templink%
25\def\LWR@templinktwo{#2}%
26\@onelevel@sanitize\LWR@templinktwo%
27\def\LWR@templinkthree{#3}%
28\@onelevel@sanitize\LWR@templinkthree%
29\LWR@htmltag{a href="\LWR@templink\LWR@hashmark%
30   \LWR@templinktwo.\LWR@templinkthree"%
31}%
32#4%

```

```
33 \LWR@htmltag{/a}%
34 }
```

`\LWR@hyperrefc` [*label*] {*text*}

Creates text as an HTML link to the  $\LaTeX$  label.

```
35 \NewDocumentCommand{\LWR@hyperrefc}{0{label} +m}{%
36 \LWR@startref{#1}%
37 #2%
38 \LWR@htmltag{/a}%
39 }
```

`\hyperref` {*1: URL*} {*2: category*} {*3: name*} {*4: text*} — or —  
[*1: label*] {*2: text*}

```
40 \DeclareRobustCommand*\hyperref{%
41 \LWR@ensuredoingapar%
42 \@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
43 }
```

`\hypertarget` {*name*} {*text*}

Creates an anchor to name with the given text.

```
44 \NewDocumentCommand{\hypertarget}{m +m}{%
45 \label{#1}%
46 #2%
47 }
```

`\hyperlink` {*name*} {*text*}

Creates a link to the anchor created by `hypertarget`, with the given link text.

Declared because also defined by memoir.

```
48 \DeclareDocumentCommand{\hyperlink}{m +m}{%
49 \hyperref{#1}{#2}%
50 }
```

`\autoref` \* {*label*}

For HTML, `\cleveref` is used instead.

```
51 \NewDocumentCommand{\autoref}{s m}{%
52 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
53 }
```

`\autopageref` {*label*}

For HTML, `\cleveref` is used instead.

```
54 \NewDocumentCommand{\autopageref}{s m}{%
55 \IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}}%
56 }
```

```
\pdfstringdef  {\langle macroname \rangle} {\langle TEXstring \rangle}
57 \newcommand{\pdfstringdef}[2] {}
```

```
\pdfbookmark  [[\langle level \rangle] {\langle text \rangle} {\langle name \rangle}
58 \newcommand{\pdfbookmark}[3] [] {}
```

```
\currentpdfbookmark  {\langle text \rangle} {\langle name \rangle}
59 \newcommand{\currentpdfbookmark}[2] {}
```

```
\subpdfbookmark  {\langle text \rangle} {\langle name \rangle}
60 \newcommand{\subpdfbookmark}[2] {}
```

```
\belowpdfbookmark  {\langle text \rangle} {\langle name \rangle}
61 \newcommand{\belowpdfbookmark}[2] {}
```

```
\texorpdfstring  {\langle TEXstring \rangle} {\langle PDFstring \rangle}
62 \newcommand{\texorpdfstring}[2] {#1}
```

```
\hypercalcbp  {\langle dimen \rangle} From hyperref.
63 \def\hypercalcbp#1{%
64 \strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax
65 }%
```

```
\Acrobatmenu  {\langle menuoption \rangle} {\langle text \rangle}
66 \newcommand{\Acrobatmenu}[2] {}
```

```
\TextField  [[\langle parameters \rangle] {\langle label \rangle}
67 \DeclareRobustCommand{\TextField}[2] [] {}
```

```
\CheckBox  [[\langle parameters \rangle] {\langle label \rangle}
68 \DeclareRobustCommand{\CheckBox}[2] [] {}
```

```
\ChoiceMenu  [\langle parameters \rangle] {\langle label \rangle} {\langle choices \rangle}
69 \DeclareRobustCommand{\ChoiceMenu}[3] [] {}
```

```
\PushButton  [\langle parameters \rangle] {\langle label \rangle}
70 \DeclareRobustCommand{\PushButton}[2] [] {}
```

```
\Submit      [\langle parameters \rangle] {\langle label \rangle}
71 \DeclareRobustCommand{\Submit}[2] [] {}
```

```
\Reset       [\langle parameters \rangle] {\langle label \rangle}
72 \DeclareRobustCommand{\Reset}[2] [] {}
```

```
\Gauge       [\langle parameters \rangle] {\langle label \rangle}
73 \DeclareRobustCommand{\Gauge}[2] [] {}
```

```
\LayoutTextField  {\langle label \rangle} {\langle field \rangle}
74 \newcommand*{\LayoutTextField}[2] {}
```

```
\LayoutChoiceField  {\langle label \rangle} {\langle field \rangle}
75 \newcommand*{\LayoutChoiceField}[2] {}
```

```
\LayoutCheckField  {\langle label \rangle} {\langle field \rangle}
76 \newcommand*{\LayoutCheckField}[2] {}
```

```
\MakeRadioField    {\langle width \rangle} {\langle height \rangle}
77 \newcommand*{\MakeRadioField}[2] {}
```

```
\MakeCheckField    {\langle width \rangle} {\langle height \rangle}
78 \newcommand*{\MakeCheckField}[2] {}
```

```
\MakeTextField     {\langle width \rangle} {\langle height \rangle}
79 \newcommand*{\MakeTextField}[2] {}
```



```

\MakeChoiceField  {\langle width\rangle} {\langle height\rangle}
                  80 \newcommand*{\MakeChoiceField}[2] {}

\MakeFieldButton  {\langle text\rangle}
                  81 \newcommand{\MakeFieldButton}[1] {}

```

---

File 111 **lwarp-hyperxmp.sty**

§ 198     Package **hyperxmp**

Pkg    hyperxmp    Emulated.

for HTML output:    Discard all options for lwarp-hyperxmp:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```

---

File 112 **lwarp-hyphenat.sty**

§ 199     Package **hyphenat**

Pkg    hyphenat    hyphenat is emulated during HTML output, while the print-mode version is used inside a lateximage.

for HTML output:    1 \LWR@ProvidesPackagePass{hyphenat}

```

2 \LetLtxMacro\LWRHYNAT@origtextnhtt\textnhtt
3 \LetLtxMacro\LWRHYNAT@originhttfamily\nhttfamily
4 \LetLtxMacro\LWRHYNAT@orignohyphens\nohyphens
5 \LetLtxMacro\LWRHYNAT@origbshyp\bshyp
6 \LetLtxMacro\LWRHYNAT@origfshyp\fshyp
7 \LetLtxMacro\LWRHYNAT@origdothyp\dothyp
8 \LetLtxMacro\LWRHYNAT@origcolonyhyp\colonyhyp
9 \LetLtxMacro\LWRHYNAT@orighyp\hyp
10
11 \LetLtxMacro\textnhtt\texttt
12 \LetLtxMacro\nhttfamily\ttfamily
13
14 \renewcommand{\nohyphens}[1]{#1}
15 \renewrobustcmd{\bshyp}{%
16     \ifmmode\backslash\else\textbackslash\fi%
17 }

```

---

```

18 \renewrobustcmd{\fshyp}{/}
19 \renewrobustcmd{\dothyp}{.}
20 \renewrobustcmd{\colonhyp}{:}
21 \renewrobustcmd{\hyp}{-}
22
23 \appto\LWR@restoreorigformatting{
24 \LetLtxMacro\textnhtt\LWRHYNAT@origtextnhtt
25 \LetLtxMacro\nhttfamily\LWRHYNAT@orignhttfamily
26 \LetLtxMacro\nohyphens\LWRHYNAT@orignohyphens
27 \LetLtxMacro\bshyp\LWRHYNAT@origbshyp
28 \LetLtxMacro\fshyp\LWRHYNAT@origfshyp
29 \LetLtxMacro\dothyp\LWRHYNAT@origdothyp
30 \LetLtxMacro\colonhyp\LWRHYNAT@origcolonhyp
31 \LetLtxMacro\hyp\LWRHYNAT@orighyp
32 }

```

---

File 113 **lwarp-idxlayout.sty**

§ 200      Package **idxlayout**

*(Emulates or patches code by THOMAS TITZ.)*

Pkg    idxlayout    Emulated.

**for HTML output:**    Discard all options for lwarp-idxlayout:

```

1 \LWR@ProvidesPackageDrop{idxlayout}

2 \newcommand{\LWR@indexprenote}{}
3
4 \renewcommand*{\printindex}
5 {
6 \LWR@startpars
7
8 \LWR@indexprenote
9
10 \LWR@origprintindex
11 }
12
13 \newcommand{\setindexprenote}[1]{\renewcommand{\LWR@indexprenote}{#1}}
14 \newcommand*{\noindexprenote}{\renewcommand{\LWR@indexprenote}{} }
15
16 \newcommand{\idxlayout}[1]{ }
17 \newcommand*{\indexfont}{}
18 \newcommand*{\indexjustific}{}
19 \newcommand*{\indexsubsdelim}{}
20 \newcommand*{\indexstheadcase}{}

```

---

File 114 **lwarp-ifoddpagel.sty**

§ 201      Package **ifoddpagel**

*(Emulates or patches code by MARTIN SCHARRER.)*

Pkg ifoddpagel ifoddpagel is emulated.

for HTML output: Discard all options for lwarp-ifoddpagel:

```
1 \LWR@ProvidesPackageDrop{ifoddpagel}

2 \newif\ifoddpagel
3
4 \newif\ifoddpageloroneside
5
6 \DeclareRobustCommand{\checkoddpagel}{\oddpageltrue\oddpageloronesideltrue}
7
8 \def\oddpagel@pagel{1}
9
10 \def\@ifoddpagel{%
11     \expandafter\@firstoftwo
12 }
13
14 \def\@ifoddpageloronesidel{%
15     \expandafter\@firstoftwo
16 }
```

---

File 115 **lwarp-indentfirst.sty**

§ 202      Package **indentfirst**

Pkg indentfirst indentfirst is ignored.

Discard all options for lwarp-indentfirst:

for HTML output: 1 \LWR@ProvidesPackageDrop{indentfirst}

---

File 116 **lwarp-inputenc.sty**

§ 203     Package **inputenc**

Pkg    inputenc    Error if inputenc is loaded after lwarp.

Discard all options for lwarp-inputenc:

**for HTML output:**    1 \LWR@ProvidesPackageDrop{inputenc}

                         2 \LWR@loadbefore{inputenc}


---

File 117 **lwarp-keyfloat.sty**

§ 204     Package **keyfloat**

*(Emulates or patches code by BRIAN DUNN.)*

Pkg    keyfloat    keyfloat is supported with minor adjustments.

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

**for HTML output:**    1 \LWR@ProvidesPackagePass{keyfloat}

After keyfloat has loaded:

```
2 \AtBeginDocument{

3 \RenewDocumentCommand{\KFLT@onefigureimage}{}
4 {%
5 \LWR@traceinfo{KFLT@onefigureimage}%
6 % \begin{lrbox}{\KFLT@envbox}%
7 \ifthenelse{\NOT\equal{\KFLT@lw}{}}{%
8 {\includegraphics%
9 [scale=\KFLT@s,width=\KFLT@imagewidth]{\KFLT@i}}%
10 {% not linewidth
11 \ifthenelse{\dimtest{\KFLT@w}{>}{0pt}}{%
12 {% width is given
13 \ifthenelse{\dimtest{\KFLT@h}{>}{0pt}}%
```

---

```

14 {% w and h
15 \includegraphics%
16 [scale=\KFLT@s,%
17 width=\KFLT@imagewidth,height=\KFLT@h]{\KFLT@i}%
18 }% w and h
19 {% only w
20 \includegraphics%
21 [scale=\KFLT@s,width=\KFLT@imagewidth]{\KFLT@i}%
22 }% only w
23 }% width is given
24 {% width is not given
25 \ifthenelse{\dimtest{\KFLT@h}{>}{0pt}}%
26 {\includegraphics%
27 [scale=\KFLT@s,height=\KFLT@h]{\KFLT@i}}%
28 {\includegraphics%
29 [scale=\KFLT@s]{\KFLT@i}}%
30 }% width is not given
31 }% not linewidth
32 % \end{lrbox}%
33 % \unskip%
34 % \KFLT@findenvboxwidth%
35 % \begin{turn}{\KFLT@r}%
36 % \KFLT@frame{\usebox{\KFLT@envbox}}%
37 % \unskip%
38 % \end{turn}%
39 \LWR@traceinfo{\KFLT@onefigureimage: done}%
40 }

41 \RenewDocumentEnvironment{\KFLT@boxinner}{%
42 {%
43 \LWR@traceinfo{\KFLT@boxinner}%
44 \LWR@stoppars%
45 }
46 {
47 \LWR@startpars%
48 \LWR@traceinfo{\KFLT@boxinner: done}%
49 }

50 \DeclareDocumentEnvironment{\KFLT@marginfloat}{0{-1.2ex} m}
51 {%
52 \LWR@BlockClassWP{float:right; width:2in; margin:10pt}{\marginblock}%
53 \captionsetup{type=#2}%
54 }
55 {%
56 \endLWR@BlockClassWP%
57 }

58 \DeclareDocumentEnvironment{\marginfigure}{o}
59 {\begin{\KFLT@marginfloat}{figure}}

```

---

```

60  {\end{KFLT@marginfloat}}
61
62  \DeclareDocumentEnvironment{margintable}{o}
63  {\begin{KFLT@marginfloat}{table}}
64  {\end{KFLT@marginfloat}}

65  \DeclareDocumentEnvironment{keywrap}{m +m}
66  {%
67  \LWR@ensuredoingapar%
68  \setlength{\LWR@templengthone}{#1}%
69  \begin{LWR@BlockClassWP}{%
70    float:right; width:\LWR@printlength{\LWR@templengthone}; % extra space
71    margin:10pt%
72  }%
73  {%
74    width:\LWR@printlength{\LWR@templengthone}%
75  }%
76  {marginblock}%
77  \setlength{\linewidth}{.95\LWR@templengthone}%
78  #2%
79  \end{LWR@BlockClassWP}%
80  }
81  {%
82  }

83  }% AtBeginDocument

```

---

File 118 **lwarp-layout.sty**

§ 205      Package **layout**

*(Emulates or patches code by KENT MCPHERSON, JOHANNES BRAAMS, HIDEO UMEKI.)*

Pkg    layout    layout is emulated.

**for HTML output:**    Discard all options for lwarp-layout:

```

1  \LWR@ProvidesPackageDrop{layout}

2  \NewDocumentCommand{\layout}{s}{-}

```

---

File 119 **lwarp-letterspace.sty**

§ 206      Package **letterspace**

*(Emulates or patches code by R SCHLICHT.)*

Pkg **letterspace**      letterspace is a subset of microtype, which is pre-loaded by lwarp. All user options and macros are ignored and disabled.

**for HTML output:**      Discard all options for lwarp-letterspace:

```
1 \LWR@ProvidesPackageDrop{letterspace}

2 \newcommand*\lsstyle{}
3 \newcommand\textls[2] [] {}
4 \def\textls#1#{}
5 \newcommand*\lslig[1] {#1}
```

---

File 120 **lwarp-lettrine.sty**

§ 207      Package **lettrine**

*(Emulates or patches code by DANIEL FLIPO.)*

Pkg **lettrine**      Emulated.

**for HTML output:**      Discard all options for lwarp-lettrine:

```
1 \LWR@ProvidesPackageDrop{lettrine}
```

The initial letter is in a <span> of class lettrine, and the following text is in a <span> of class lettrinetext. \lettrine [*<keys>*] {*<letter>*} {*<additional text>*}

```
2 \DeclareDocumentCommand{\lettrine}{o m m}{%
3 \InlineClass{lettrine}{#2}\InlineClass{lettrinetext}{#3} % extra space
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 121 **lwarp-lineno.sty**

§ 208      Package **lineno**

*(Emulates or patches code by STEPHAN I. BÖTTCHER.)*

Pkg    **lineno**    **lineno** is ignored.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{lineno}

2 \newcommand*\resetlinenumber[1][\@ne]{}
3
4 \def\linenumbers{%
5     \@ifnextchar[{\resetlinenumber}%]
6         {\@ifstar{\resetlinenumber}{}{}}%
7     }
8
9 \newcommand*\nolinenumbers{}
10
11 \@namedef{linenumbers*}{\par\linenumbers*}
12 \@namedef{runninglinenumbers*}{\par\runninglinenumbers*}
13
14 \def\endlinenumbers{\par}
15 \let\endrunninglinenumbers\endlinenumbers
16 \let\endpagewiselinenumbers\endlinenumbers
17 \expandafter\let\csname endlinenumbers*\endcsname\endlinenumbers
18 \expandafter\let\csname endrunninglinenumbers*\endcsname\endlinenumbers
19 \let\endnolinenumbers\endlinenumbers
20

```



---

```

21 \def\pagewiselinenumbers{\linenumbers\setpagewiselinenumbers}
22
23 \def\runninglinenumbers{\setrunninglinenumbers\linenumbers}
24
25 \def\setpagewiselinenumbers{}
26
27 \def\setrunninglinenumbers{}
28
29 \def\linenomath{}%
30 \@namedef{linenomath*}{}%
31 \def\endlinenomath{}
32 \expandafter\let\csname endlinenomath*\endcsname\endlinenomath
33
34 \let\linelabel\label
35
36 \def\switchlinenumbers{\@ifstar{}{}}
37 \def\setmakelinenumbers#1{\@ifstar{}{}}
38
39 \def\leftlinenumbers{\@ifstar{}{}}
40 \def\rightlinenumbers{\@ifstar{}{}}
41
42 \newcounter{linenumber}
43 \newcount\c@pagewiselinenumber
44 \let\c@runninglinenumber\c@linenumber
45
46 \def\runningpagewiselinenumbers{}
47 \def\realpagewiselinenumbers{}
48
49
50 \NewDocumentCommand\modulolinenumbers{s o}{}
51
52 \chardef\c@linenumbermodulo=5
53 \modulolinenumbers[1]
54
55 \newcommand*\firstlinenumber[1]{}
56
57 \newcommand\internallinenumbers{}
58 \let\endinternallinenumbers\endlinenumbers
59 \@namedef{internallinenumbers*}{\internallinenumbers*}
60 \expandafter\let\csname endinternallinenumbers*\endcsname\endlinenumbers
61
62 \newcommand*{\linenoplaceholder}[1]{% redefine per language
63   (line number reference for \detokenize\expandafter{#1})
64 }
65
66 \newcommand*{\lineref}[2][\linenoplaceholder{#2}]
67 \newcommand*{\linerefp}[2][\linenoplaceholder{#2}]
68 \newcommand*{\linerefr}[2][\linenoplaceholder{#2}]
69
70 \newcommand\quotelinenumbers

```

---

```

71   {\@ifstar\linenumbers{\@ifnextchar[\linenumbers{\linenumbers*}}}
72
73 \newdimen\linenumbersep
74 \newdimen\linenumberwidth
75 \newdimen\quotelinenumbersep
76
77 \quotelinenumbersep=\linenumbersep
78 \let\quotelinenumberfont\linenumberfont
79
80 \def\linenumberfont{\normalfont\tiny\sffamily}
81
82
83 \linenumberwidth=10pt
84 \linenumbersep=10pt
85
86 \def\thelinenumber{}
87
88 \def\LineNumber{}
89 \def\makeLineNumber{}
90 \def\makeLineNumberLeft{}
91 \def\makeLineNumberRight{}
92 \def\makeLineNumberOdd{}
93 \def\makeLineNumberEven{}
94 \def\makeLineNumberRunning{}
95
96
97 \newenvironment{numquote}      {\quote}{\endquote}
98 \newenvironment{numquotation} {\quotation}{\endquotation}
99 \newenvironment{numquote*}     {\quote}{\endquote}
100 \newenvironment{numquotation*}{\quotation}{\endquotation}
101
102 \newdimen\bframerule
103 \bframerule=\fbboxrule
104
105 \newdimen\bframesep
106 \bframesep=\fbboxsep
107
108 \newenvironment{bframe}
109 {%
110   \LWR@forceminwidth{\bframerule}%
111   \BlockClass[
112     border:\LWR@printlength{\LWR@atleastonept} solid black ; %
113     padding:\LWR@printlength{\bframesep}%
114   ]{bframe}
115 }
116 {\endBlockClass}

```

---

File 122 **lwarp-lips.sty**

§ 209      Package **lips**

*(Emulates or patches code by MATT SWIFT.)*

Pkg    lips    lips is emulated.

```

1 % \LWR@ProvidesPackageDrop{lips}
2 \PackageInfo{lwarp}{Using the lwarp version of package 'lips'.}%
3 \ProvidesPackage{lwarp-lips}
4
5 \NewDocumentCommand{\Lips}{\textellipsis}
6
7 \NewDocumentCommand{\BracketedLips}{\textellipsis}
8
9 \let\lips\Lips
10 \let\olips\lips
11
12 \DeclareOption*{}
13 \DeclareOption{mla}{
14 \let\lips\BracketedLips
15 }
16 \ProcessOptions\relax
17
18 \newcommand \LPNobreakList {}

```

---

File 123 **lwarp-listings.sty**

§ 210      Package **listings**

*(Emulates or patches 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}

```

Force flexible columns:

```

3 \lst@column@flexible

```

Patches to embed listings inside pre tags:

```

4 \let\LWR@origlst@Init\lst@Init
5 \let\LWR@origlst@DeInit\lst@DeInit
6
7 \let\LWR@origlsthkEveryPar\lsthk@EveryPar
8
9 \renewcommand{\l@lstlisting}[2]{\hypertocfloat{1}{\lstlisting}{1ol}{#1}{#2}}

```

`\lst@Init`    *{\backslash-processing}*    Done at the start of a listing.

```
10 \renewcommand{\lst@Init}[1]{%
```

First, perform the listings initialization:

```

11 \LWR@traceinfo{\lst@Init}%
12 \renewcommand*{\@capttype}{\lstlisting}%
13 \LWR@origlst@Init{#1}%
14 \LWR@traceinfo{finished origlst@Init}%
15 \lst@ifdisplaystyle%

```

Creating a display.

Disable line numbers, produce the <pre>, then reenale line numbers.

```

16 \LWR@traceinfo{About to create verbatim.}%
17 \let\lsthk@EveryPar\relax%
18 \LWR@forcenewpage
19 \LWR@atbeginverbatim{programlisting}%
20
21 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
22 \else%

```

Inline, so open a <span>:

```

23 \ifbool{\LWR@verbtags}{\LWR@htmltag{span class="inlineprogramlisting"}}{}%
24 \fi%
25 }

```

`\lst@DeInit`    Done at the end of a listing.

```

26 \renewcommand*{\lst@DeInit}{%
27 \lst@ifdisplaystyle%

```

Creating a display.

Disable line numbers, produce the </pre>, then reenale line numbers:

```

28 \let\lsthk@EveryPar\relax%
29
30 \LWR@afterendverbatim%
31 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
32 \else%

```

Inline, so create the closing `</span>`:

```
33 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{</span>}}{}%
34 \fi%
```

Final listings deinit:

```
35 \LWR@origlst@DeInit%
36 }
```

```
\lst@MakeCaption {</b>}
```

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```
37 \def\lst@MakeCaption#1{%
38 \LWR@traceinfo{MAKING CAPTION at #1}%
39 \lst@ifdisplaystyle
40 \LWR@traceinfo{making a listings display caption}%
41 \ifx #1%
42 \ifx\lst@@caption\@empty\expandafter\lst@HRefStepCounter \else
43 \expandafter\refstepcounter
44 \fi {lstlisting}%
45 \LWR@traceinfo{About to assign label: !\lst@label!}%
46 % \ifx\lst@label\@empty\else
47 % \label{\lst@label}\fi
48 \LWR@traceinfo{Finished assigning the label.}%
49 \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
50 \global\let\lst@name\lst@arg \global\let\lstname\lst@name
51 \lst@ifnolol\else
52 \ifx\lst@@caption\@empty
53 \ifx\lst@caption\@empty
54 \ifx\lst@intname\@empty \else \def\lst@temp{ }%
55 \ifx\lst@intname\lst@temp \else
```

This code places a contents entry for a non-float. This would have to be modified for lwarp:

```
56 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
57 % \addcontentsline{lol}{lstlisting}{\lst@name}
58 \fi\fi
59 \fi
60 \else
```

This would have to be modified for lwarp:

```
61 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
62 \addcontentsline{lol}{lstlisting}%
63 {\protect\numberline{\thelstlisting}%
64 {\protect\ignorespaces \lst@@caption \protect\relax}}%
65 \fi
66 \fi
67 \fi
```

```

68 \ifx\lst@caption\@empty\else
69 \LWR@traceinfo{lst@caption not empty-}%
70 \lst@ifsubstring #1\lst@captionpos
71 {\begingroup
72 \LWR@traceinfo{at the selected position}%

```

These space and box commands are not needed for HTML output:

```

73 % \let\@vskip\vskip
74 % \def\vskip{\afterassignment\lst@vskip \@tempkipa}%
75 % \def\lst@vskip{\nobreak\@vskip\@tempkipa\nobreak}%
76 % \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
77 % \ifx #1\allowbreak \fi
78 \ifx\lst@title\@empty

```

New lwarp code to create a caption:

```

79 \lst@makecaption\fnnum\lstlisting{\ignorespaces \lst@caption}
80 \else

```

New lwarp code to create a title:

```

81 % \lst@maketitle\lst@title % (AS)
82 \LWR@traceinfo{Making title: \lst@title}%
83 \begin{BlockClass}{\lstlistingtitle}% lwarp
84 \lst@maketitle\lst@title% lwarp
85 \end{BlockClass}% lwarp
86 \fi
87 \LWR@traceinfo{About to assign label: !\lst@label!}%
88 \ifx\lst@label\@empty\else
89 \leavevmode% gets rid of bad space factor error
90 \GetTitleStringExpand{\lst@caption}%
91 \edef\LWR@lntemp{\GetTitleStringResult}%
92 \edef\@currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
93 \label{\lst@label}\fi
94 \LWR@traceinfo{Finished assigning the label.}%

```

Not needed for lwarp:

```

95 % \ifx #1b\allowbreak \fi
96 \endgroup}{}%
97 \fi
98 \LWR@traceinfo{end of making a listings display caption}%
99 \else
100 \LWR@traceinfo{INLINE}%
101 \fi
102 \LWR@traceinfo{DONE WITH CAPTION at #1}%
103 }

```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```

104 \lst@Key{numbers}{none}{%

```

---

```

105 \let\lst@PlaceNumber\@empty
106 \lstKV@SwitchCases{#1}%
107 {none&\\%
108 left&\def\lst@PlaceNumber{%
109 % \llap{
110 \LWR@orignormalfont%
111 \lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
112 % }
113 }
114 \\%
115 right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont
116 \kern\VerbatimHTMLWidth \kern\lst@numbersep
117 \lst@numberstyle{\thelstnumber}}}%
118 }\PackageError{Listings}{Numbers #1 unknown}\@ehc}}

119 \end{warpHTML}

```

---

File 124 **lwarp-longtable.sty**

## § 211 Package **longtable**

*(Emulates or patches code by DAVID CARLISLE.)*

Pkg **longtable** longtable is emulated.

**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.



**lateximage** longtable is not supported inside a lateximage.

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{\LTcapttype}%
5 \caption@setoptions{longtable}%
6 \caption@setoptions{@longtable}%
7 \caption@LT@setup%
8 \booltrue{LWR@starredlongtable}%
9 \let\captionlistentry\LWR@LTcaptionlistentry%
10 \LWR@tabular{#2}
11 }
12 {\endLWR@tabular\LWR@floatend}
13
14 \newenvironment{longtable}[2][{}]{%
15 \LWR@floatbegin{table}%
16 \setcaptiontype{\LTcapttype}%
17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{\LTcapttype}%
21 \let\captionlistentry\LWR@LTcaptionlistentry%
22 \LWR@tabular{#2}
23 }
24 {\endLWR@tabular\LWR@floatend}
25

```

Provided for compatibility, but ignored:

```

26 \newcounter{LTchunksize}
27 \def\endhead{\LWR@tabularendoffline}% throws away options //[dim] and /**
28 \def\endfirsthead{\LWR@tabularendoffline}
29 \def\endfoot{\LWR@tabularendoffline}
30 \def\endlastfoot{\LWR@tabularendoffline}
31 \newcommand\tabularnewline{\LWR@tabularendoffline}
32 \newcommand\setlongtables{}% Obsolete command, does nothing.
33 \newlength{\LTleft}
34 \newlength{\LTRight}
35 \newlength{\LTpre}

```



---

```

36 \newlength{\LTpost}
37 \newlength{\LTcapwidth}

38 \renewcommand*{\kill}{\LWR@tabularendoffline}

```

---

File 125 **lwarp-lscape.sty**

§ 212     Package **lscape**

*(Emulates or patches code by D. P. CARLISLE.)*

Pkg    **lscape**    **lscape** is emulated.

**for HTML output:**    Discard all options for lwarp-lscape.

```

1 \LWR@ProvidesPackageDrop{lscape}

2 \newenvironment*{landscape}{}{}

```

---

File 126 **lwarp-ltcaption.sty**

§ 213     Package **ltcaption**

*(Emulates or patches code by AXEL SOMMERFELDT.)*

Pkg    **ltcaption**    **ltcaption** is emulated.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{ltcaption}

\LTcaptype is already defined by lwarp.

longtable\* is already defined by lwarp-longtable.

```

2 \newlength{\LTcapskip}
3 \newlength{\LTcapleft}
4 \newlength{\LTcapright}
5 \newcommand*{\LTcapmarginfalse}{}

```

---

File 127 **lwarp-ltxtable.sty**

§ 214    Package **ltxtable**

Pkg    ltxtable    ltxtable is emulated.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{ltxtable}

\LTxtable    {\width} {\file}  
               2 \newcommand\*\LTxtable}[2]{%  
               3 \input{#2}%  
               4 }

---

File 128 **lwarp-luacolor.sty**

§ 215    Package **luacolor**

Pkg    luacolor    luacolor is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{luacolor}  
                       2 \newcommand{\luacolorProcessBox}[1]{}

---

File 129 **lwarp-luatodonotes.sty**

§ 216    Package **luatodonotes**

*(Emulates or patches code by FABIAN LIPP.)*

Pkg    luatodonotes    luatodonotes is emulated.

The documentation for todonotes and luatodonotes have an example with a todo inside a caption. If this example does not work it will be necessary to move the todo outside of the caption.

**for HTML output:**    1 \LWR@ProvidesPackagePass{luatodonotes}

Nullify options:

```

2 \@todonotes@additionalMarginEnabledfalse

3 \if@todonotes@disabled
4 \else
5
6 \newcommand{\ext@todo}{tdo}
7
8 \renewcommand{\l@todo}[2]{\hypertocfloat{1}{\todo}{ldo}{#1}{#2}}

9 \let\LWRTODONOTES@orig@todototoc\todototoc
10
11 \renewcommand*{\todototoc}{%
12 \phantomsection%
13 \LWRTODONOTES@orig@todototoc%
14 }
15
16
17 \renewcommand{\@todonotes@drawMarginNoteWithLine}{%
18 \fcolorbox
19   {\@todonotes@currentbordercolor}
20   {\@todonotes@currentbackgroundcolor}
21   {\arabic{\@todonotes@numberoftodonotes}}
22 \marginpar{\@todonotes@drawMarginNote}
23 }
24
25 \renewcommand{\@todonotes@drawInlineNote}{%
26 \fcolorboxBlock%
27   {\@todonotes@currentbordercolor}%
28   {\@todonotes@currentbackgroundcolor}%
29   {%
30     \if@todonotes@authorgiven%
31     {\@todonotes@author:\,}%
32     \fi%
33     \@todonotes@text%
34   }%
35 }
36
37 \newcommand{\@todonotes@drawMarginNote}{%
38   \if@todonotes@authorgiven%
39     \@todonotes@author\par%
40   \fi%
41   \arabic{\@todonotes@numberoftodonotes}: %
42   \fcolorbox%
43     {\@todonotes@currentbordercolor}%
44     {\@todonotes@currentbackgroundcolor}%
45     {%
46       \@todonotes@sizecommand%

```

---

```

47      \@todonotes@text %
48    }%
49 }%
50
51 \renewcommand{\missingfigure}[2] [] {%
52 \setkeys{todonotes}{#1}%
53 \addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
54 \fcolorboxBlock%
55   {\@todonotes@currentbordercolor}%
56   {\@todonotes@currentfigcolor}%
57   {%
58     \setlength{\fboxrule}{4pt}%
59     \fcolorbox{red}{white}{Missing figure} \quad #2%
60   }
61 }
62
63 \LetLtxMacro\LWRTODONOTES@orig@todocommon\@todocommon
64
65 \RenewDocumentCommand{\@todocommon}{m m} {%
66 \begingroup%
67 \renewcommand*\phantomsection{}%
68 \LWRTODONOTES@orig@todocommon{#1}{#2}%
69 \endgroup%
70 }
71
72 \renewcommand{\@todoarea}[3] [] {%
73   \@todonotes@areaselectedtrue%
74   \@todocommon{#1}{#2}%
75   \@todonotes@textmark@highlight{#3}%
76   \zref@label{\@todonotes@arabic{\@todonotes@numberoftodonotes}@end}%
77 }%
78
79
80 \DeclareDocumentCommand{\@todonotes@textmark@highlight}{m} {%
81 \InlineClass[background:\#B3FFB3]{highlight}{#1}%
82 }
83
84 \fi% \if@todonotes@disabled

```

---

File 130 **lwarp-marginfit.sty**

§ 217 Package **marginfit**

Pkg marginfit marginfit is ignored.

for HTML output: Discard all options for lwarp-marginfit:

---

```
1 \LWR@ProvidesPackageDrop{marginfix}
```

---

File 131 **lwarp-marginfix.sty**

§ 218      Package **marginfix**

*(Emulates or patches code by STEPHEN HICKS.)*

Pkg marginfix Emulated.

for HTML output: Discard all options for lwarp-marginfix:

```
1 \LWR@ProvidesPackageDrop{marginfix}

2 \newcommand*{\marginsskip}[1]{}
3 \newcommand*{\clearmargin}{}
4 \newcommand*{\softclearmargin}{}
5 \newcommand*{\extendmargin}[1]{}
6 \newcommand*{\mparshift}[1]{}
7 \newdimen\marginheightadjustment
8 \newdimen\marginposadjustment
9 \newcommand*{\blockmargin}[1][1]{}
10 \newcommand*{\unblockmargin}[1][1]{}
11 \newcommand*{\marginphantom}[2][1]{}

```

---

File 132 **lwarp-marginnote.sty**

§ 219      Package **marginnote**

*(Emulates or patches code by MARKUS KOHM.)*

Pkg marginnote Emulated.

for HTML output: Discard all options for lwarp-marginnote:

```
1 \LWR@ProvidesPackageDrop{marginnote}

2 \NewDocumentCommand{\marginnote}{o +m o}{\marginpar{#2}}
3 \newcommand*{\marginnoteleftadjust}{}
4 \newcommand*{\marginnoterightadjust}{}
5 \newcommand*{\marginnotetextwidth}{}
6 \let\marginnotetextwidth\textwidth
7 \newcommand*{\marginnotevadjust}{}
8 \newcommand*{\marginfont}{}

```

---

```

9 \newcommand*{\raggedleftmarginnote}{}
10 \newcommand*{\raggedrightmarginnote}{}

```

---

File 133 **lwarp-mcaption.sty**

§ 220    Package    **mcaption**

*(Emulates or patches code by STEPHAN HENNIG.)*

Pkg    mcaption    mcaption is nullified.

for HTML output:    Discard all options for lwarp-mcaption:

```

1 \LWR@ProvidesPackageDrop{mcaption}

2 \newenvironment{margincap}{}{}
3 \newcommand*{\margincapalign}{}
4 \newlength{\margincapsep}

```

---

File 134 **lwarp-mdframed.sty**

§ 221    Package    **mdframed**

*(Emulates or patches code by MARCO DANIEL, ELKE SCHUBERT.)*

Pkg    mdframed    mdframed is loaded with options forced to framemethod=none.

#### § 221.1    **Limitations**

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

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

font    For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

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

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

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

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

**CSS classes** Environments created or encapsulated by `mdframed` are enclosed in a `<div>` of class `md<environmentname>`, or `mdframed` otherwise.

Frame titles are placed into a `<span>` of class `mdframedtitle`. Subtitles are in a `<span>` of class `mdframedsubtitle`, and likewise for subsubtitles.

Pre-existing hooks are used to patch extra functions before and after the frames.

## § 221.2 Package loading

**for HTML output:**

```
1 \RequirePackage{xcolor}% for \convertcolorspec
2
3 \LWR@ProvidesPackageDrop{mdframed}
```

`amsthm` must be loaded before `mdframed`

```
4 \LWR@origRequirePackage{amsthm}
```

Do not require `Tikz` or `pstricks`:

```
5 \LWR@origRequirePackage[framemethod=none]{mdframed}
```

## § 221.3 Patches

Patch to remove PDF formatting and add HTML tags:

```
6 \AtBeginDocument{
7 \def\mdf@trivlist#1{%
8   \edef\mdf@temp{%
9     \topsep=\the\topsep\relax%
10    \partopsep=\the\partopsep\relax%
11    \parsep=\the\parsep\relax%
12  }%
13    \setlength{\topsep}{#1}%
14    \topskip\z@%
15    \partopsep\z@%
16    \parsep\z@%
```

```

17 % \@nmbrlistfalse%
18 % \@trivlist%
19 % \labelwidth\z@%
20 % \leftmargin\z@%
21 % \itemindent\z@%
22 \let\@itemlabel\@empty%
23 \def\makelabel##1{##1}%
24 % \item\relax\mdf@temp\relax%
25 }
26
27 \renewcommand*{\endmdf@trivlist}{%
28 \LWR@traceinfo{endmdf@trivlist}%
29 % \endtrivlist%
30 \LWR@listend%
31 }
32 }% AtBeginDocument

```

#### § 221.4 Initial setup

To handle CSS and paragraphs, patch code at start and end of environment and contents. \LWR@origraggedright helps avoid hyphenation.

```

33 \mdfsetup{
34 startcode={\LWR@mdframedstart\LWR@origraggedright},
35 endcode={\LWR@mdframedend},
36 startinnercode={\LWR@startpars\LWR@origraggedright},
37 endinnercode={\LWR@stoppars},
38 }

```

#### § 221.5 Color and length HTML conversion

\LWR@mdfprintcolor    {\<mdfcolorkey>}

Given the mdframed key, print the color.

```

39 \newcommand*{\LWR@mdfprintcolor}[1]{%
40 \convertcolorspec{named}{\csuse{mdf@#1}}{HTML}\LWR@tempcolor%
41 \#\LWR@tempcolor
42 }

```

\LWR@mdfprintlength    {\<mdflengthkey>}

Given the mdframed key, print the length.

```

43 \newcommand*{\LWR@mdfprintlength}[1]{%
44 \LWR@printlength{\csuse{mdf@#1@length}}
45 }

```



## § 221.6 Environment encapsulation

`\LWR@mdframedstart` Actions before an mdframe starts.

Encapsulate a frame inside a `<div>` of the desired class.

```
46 \newcommand*{\LWR@mdframedstart}{%
47 \LWR@traceinfo{\LWR@mdframedstart start}%
```

Turn off paragraph handling during the generation of the encapsulating tags:

```
48 \LWR@stoppars%
```

Open a `<div>` and with custom class and custom style:

```
49 \LWR@htmltagc{div class="\LWR@mdthisenv" \LWR@orignewline
50 style=" \LWR@orignewline
```

Convert and print the background color:

```
51 background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```
52 border: \LWR@mdfprintlength{linewidth} solid
53 \LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
54 border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
55 \ifbool{mdf@shadow}{%
56   box-shadow:
57     \LWR@mdfprintlength{shadowsize}
58     \LWR@mdfprintlength{shadowsize}
59     \LWR@mdfprintlength{shadowsize}
60     \LWR@mdfprintcolor{shadowcolor} ;
61 }
62 {box-shadow: none ;}
63 \LWR@orignewline
```

```
64 "}
```

```
65 % \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`:

```
66 \LetLtxMacro{\hspace}{\LWR@orighspace}%
67 \LetLtxMacro{\rule}{\LWR@origrule}%
68 \LetLtxMacro{\makebox}{\LWR@origmakebox}%
69 \LWR@startpars%
70 \LWR@traceinfo{\LWR@mdframedstart done}%
71 }
```

`\LWR@mdframedend` Actions after an mdframe ends.

After closing the `<div>`, globally restore to the default environment type:

```
72 \newcommand*{\LWR@mdframedend}{%
73 \LWR@traceinfo{\LWR@mdframedend start}%
```

Close the custom `<div>`:

```
74 \LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
75 \gdef\LWR@mdthisenv{mdframed}
```

Resume paragraph handling:

```
76 \LWR@startpars%
77 \LWR@traceinfo{\LWR@mdframedend done}%
78 }
```

## § 221.7 Mdframed environment

```
79 \renewenvironment{mdframed}[1][\fi]{%
80 \color@begingroup%
81 \mdfsetup{userdefinedwidth=\linewidth,#1}%
82 \mdf@startcode%
83 \mdf@preenvsetting%
84 \ifdefempty{\mdf@firstframetitle}{}%
85     {\let\mdf@frametitlesave\mdf@frametitle%
86      \let\mdf@frametitle\mdf@firstframetitle%
87      }%
88 \ifvmode\nointerlineskip\fi%
89     \ifdefempty{\mdf@frametitle}{}%
90     {\mdfframedtitleenv{\mdf@frametitle}%
91      \mdf@@frametitle@use%
92      }%
93 \mdf@trivlist{\mdf@skipabove@length}%%
94 \mdf@settings%
95 \mdf@lrbox{\mdf@splitbox@one}%
96 \mdf@startinnercode%
97 }%
98 {%
99 \mdf@@ignorelastdescenders%
100 \par%
101 \unskip\ifvmode\nointerlineskip\hrule \@height\z@ \@width\hsize\fi%
102 \ifmdf@footnoteinside%
103     \def\mdf@reserveda{%
104         \mdf@footnoteoutput%
105         \mdf@endinnercode%
106         \endmdf@lrbox%
107         \ifdefempty{\mdf@frametitle}{}%
108         {\mdfframedtitleenv{\mdf@frametitle}\mdf@@frametitle@use}%
109         \detected@mdf@put@frame
```

```

110     }%
111   \else%
112     \def\mdf@reserveda{%
113 %       \mdf@endinnercode%
114 %       \endmdf@lrbbox%
115 %       \ifdefempty{\mdf@frametitle}{}%
116 %       {\mdfframedtitleenv{\mdf@frametitle}\mdf@@frametitle@use}%
117 %       \detected@mdf@put@frame%
118 %       \mdf@footnoteoutput%
119     }%
120   \fi%
121   \mdf@reserveda%
122   \aftergroup\endmdf@trivlist%
123   \color@endgroup%
124   \mdf@endcode%
125 }

```

\mdf@footnoteoutput

```

126 \renewrobustcmd*\mdf@footnoteoutput{%
127   \LWR@printpendingmpfootnotes%
128 }

```

## § 221.8 Titles and subtitles

\mdfframedtitleenv {<title>}

Encapsulation of the original which places the title inside a <span> of class mdfframedtitle:

```

129 \LetLtxMacro\LWR@origmdfframedtitleenv\mdfframedtitleenv
130
131 \newlength{\LWR@titleroundcorner}
132
133 \renewrobustcmd\mdfframedtitleenv[1]{%
134 \LWR@traceinfo{\LWR@mdfframedtitleenv start}%
135 % \LWR@origmdfframedtitleenv{%

```

Open a <span> with a custom class and custom style:

```

136 \LWR@htmltagc{span class="mdfframedtitle" \LWR@orignewline
137 style=" \LWR@orignewline

```

Convert and print the title background color:

```

138 background:
139 \LWR@mdfprintcolor{frametitlebackgroundcolor}
140 ; \LWR@orignewline

```

Convert and print the title rule:

```

141 \ifbool{mdf@frametitlerule}{%
142   border-bottom:
143   \LWR@mdfprintlength{frametitlerulewidth}

```

```

144     solid
145     \LWR@mdfprintcolor{frametitlecolor}
146     ; \LWR@orignewline
147 }{}%

```

The title's top border radius is adjusted for the line width:

```

148 border-radius:
149 \setlength{\LWR@titleroundcorner}
150     {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{0pt}}
151     \LWR@printlength{\LWR@titleroundcorner}
152     \LWR@printlength{\LWR@titleroundcorner}
153     Opt Opt
154     \LWR@orignewline

```

Finish the custom style and the opening span tag:

```

155 " \LWR@orignewline
156 }% span

```

Restrict paragraph tags inside a span:

```

157 \begin{LWR@nestspan}%

```

Print the title inside the span:

```

158 #1%

```

Close the span and unnest the paragraph tag restriction:

```

159 \LWR@htmltagc{/span}%
160 \end{LWR@nestspan}%
161 % }
162 \LWR@traceinfo{LWR@mdframedtitleenv end}%
163 }

```

```

\LWR@mdfsubtitlecommon {\sub -or- subsub} [\langle options \rangle] {\langle title \rangle}

```

Common code for \LWR@mdfsubtitle and \LWR@mdfsubsubtitle.

Encapsulate the subtitle inside a <span> of class mdframedsubtitle:

```

164 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
165 {% the following empty line is required
166
167 \LWR@traceinfo{LWR@mdframedsubtitlecommon start}%

```

Special handling for mdframed: Subtitles have \pars around them, so temporarily disable them here.

```

168 \let\par\LWR@origpar%

```

Open a <span> with a custom class and custom style:

```

169 \LWR@htmltagc{span class="mdframed#1title"
170 style=" \LWR@orignewline

```

Convert and print the background color:

```
171 background:
172 \LWR@mdfprintcolor{#1titlebackgroundcolor}
173 ; \LWR@orignewline
```

Convert and print the above line:

```
174 \ifbool{mdf@#1titleaboveline}{%
175     border-top:
176     \LWR@mdfprintlength{#1titleabovelinewidth}
177     solid
178     \LWR@mdfprintcolor{#1titleabovelinecolor}
179     ; \LWR@orignewline
180 }{}%
```

Convert and print the below line:

```
181 \ifbool{mdf@#1titlebelowline}{%
182     border-bottom:
183     \LWR@mdfprintlength{#1titlebelowlinewidth}
184     solid
185     \LWR@mdfprintcolor{#1titlebelowlinecolor}
186     ; \LWR@orignewline
187 }{}%
```

Finish the custom style and the opening span tag:

```
188 "% span
```

Restrict paragraph tags inside a span:

```
189 \begin{LWR@nestspan}%
```

Perform the original subtitle action:

```
190 \IfNoValueTF{#2}
191 {\csuse{LWR@origmdf#1title}{#3}}%
192 {\csuse{LWR@origmdf#1title}[#2]{#3}}%
```

Close the span and unnest the paragraph tag restriction:

```
193 \LWR@htmltagc{/span}% the following empty line is required
194 \end{LWR@nestspan}% must follow the /span or an extra <p> appears
195
196 \LWR@traceinfo{LWR@mdframedsubtitlecommon end}%
197 }
```

```
\LWR@mdfsubtitle    [⟨options⟩] {⟨title⟩}

198 \newcommand*{\LWR@mdfsubtitle}{%
199 \LWR@mdfsubtitlecommon{sub}%
200 }
201 \let\mdfsubtitle\LWR@mdfsubtitle
```

```

\LWR@mdfsubsubtitle  [\langle options \rangle] {\langle title \rangle}
202 \newcommand*{\LWR@mdfsubsubtitle}{%
203 \LWR@mdfsubsubtilecommon{subsub}%
204 }
205 \let\mdfsubsubtitle\LWR@mdfsubsubtitle

```

## § 221.9 New environments

`\LWR@mdthisenv` Stores the environment of the frame about to be created:

```
206 \newcommand*{\LWR@mdthisenv}{mdframed}
```

`\newmdenv` [*\langle options \rangle*] {*\langle env-name \rangle*}

Modified from the original to remember the environment.

```

207 \renewrobustcmd*\newmdenv[2] [] {%
208 \newenvironment{#2}%
209 {%
210 \mdfsetup{#1}%
211 \renewcommand*{\LWR@mdthisenv}{md#2}%
212 \begin{mdframed}%
213 }
214 {\end{mdframed}}%
215 }

```

`\surroundwithmdframed` [*\langle options \rangle*] {*\langle environment \rangle*}

Modified from the original to remember the environment.

```

216 \renewrobustcmd*\surroundwithmdframed[2] [] {%
217 \BeforeBeginEnvironment{#2}{%
218 \renewcommand*{\LWR@mdthisenv}{md#2}%
219 \begin{mdframed}[#1]}%
220 \AfterEndEnvironment{#2}{\end{mdframed}}%
221 }

```

`\mdtheorem` [*\langle mdframed-options \rangle*] {*\langle envname \rangle*} [*\langle numberedlike \rangle*] {*\langle caption \rangle*} [*\langle within \rangle*]

Modified from the original to remember the environment.

```

222 \DeclareDocumentCommand{\mdtheorem}{ 0 {} m o m o }%
223 {\ifcsdef{#2}%
224   {\mdf@PackageWarning{Environment #2 already exists}\MessageBreak}}%
225   {%
226     \IfNoValueTF {#3}%
227     {%#3 not given -- number relationship
228       \IfNoValueTF {#5}%
229       {%#3+#5 not given
230         \@definecounter{#2}%

```

```

231 \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
232 \newenvironment{#2}[1][]{%
233 \refstepcounter{#2}%
234 \ifstrepty{##1}%
235 {\let\@temptitle\relax}%
236 {%
237 \def\@temptitle{\mdf@theoremseparator%
238 \mdf@theoremspace%
239 \mdf@theoremtitlefont%
240 ##1}%
241 \mdf@thm@caption{#2}{\csname the#2\endcsname}{##1}}%
242 }%
243 \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
244 \@temptitle}]]%
245 {\end{mdframed}}%
246 \newenvironment{#2*}[1][]{%
247 \ifstrepty{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
248 \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]%
249 {\end{mdframed}}%
250 }%
251 {%#5 given -- reset counter
252 \@definecounter{#2}\@newctr{#2}[#5]%
253 \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
254 \expandafter\xdef\csname the#2\endcsname{%
255 \expandafter\noexpand\csname the#5\endcsname \@thmcountersep%
256 \@thmcounter{#2}}%
257 \newenvironment{#2}[1][]{%
258 \refstepcounter{#2}%
259 \ifstrepty{##1}%
260 {\let\@temptitle\relax}%
261 {%
262 \def\@temptitle{\mdf@theoremseparator%
263 \mdf@theoremspace%
264 \mdf@theoremtitlefont%
265 ##1}%
266 \mdf@thm@caption{#2}{\csname the#2\endcsname}{##1}}%
267 }
268 \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
269 \@temptitle}]]%
270 {\end{mdframed}}%
271 \newenvironment{#2*}[1][]{%
272 \ifstrepty{##1}%
273 {\let\@temptitle\relax}%
274 {%
275 \def\@temptitle{\mdf@theoremseparator%
276 \mdf@theoremspace%
277 \mdf@theoremtitlefont%
278 ##1}%
279 \mdf@thm@caption{#2}{\csname the#2\endcsname}{##1}}%
280 }%

```

```

281      \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]}%
282      {\end{mdframed}}}%
283    }%
284  }%
285  {%#3 given -- number relationship
286    \global\@namedef{the#2}{\@nameuse{the#3}}%
287    \newenvironment{#2}[1][]{%
288      \refstepcounter{#3}%
289      \ifstrempy{##1}%
290        {\let\@temptitle\relax}%
291        {%
292          \def\@temptitle{\mdf@theoremseparator%
293            \mdf@theoremspace%
294            \mdf@theoremtitlefont%
295            ##1}%
296          \mdf@thm@caption{#2}{#{4}}{\csname the#2\endcsname}{##1}}%
297        }
298    \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
299      \@temptitle}]]}%
300    {\end{mdframed}}}%
301    \newenvironment{#2*}[1][]{%
302      \ifstrempy{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
303      \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]}%
304      {\end{mdframed}}}%
305    }%
306    \BeforeBeginEnvironment{#2}{\renewcommand*{\LWR@mdthisenv}{md#2}}% lwarp
307    \BeforeBeginEnvironment{#2*}{\renewcommand*{\LWR@mdthisenv}{md#2}}% lwarp
308  }%
309 }

```

`\newmdtheoremenv` [*<mdframed-options>*] [*<envname>*] [*<numberedlike>*] [*<caption>*] [*<within>*]  
 Modified from the original to remember the environment.

```

310 \DeclareDocumentCommand\newmdtheoremenv{0}{m o m o }{%
311   \ifboolexpr{ test {\IfNoValueTF {#3}} and test {\IfNoValueTF {#5}} }%
312     {\newtheorem{#2}{#4}}%
313     {%
314       \IfValueT{#3}{\newtheorem{#2}[#3]{#4}}%
315       \IfValueT{#5}{\newtheorem{#2}{#4}[#5]}%
316     }%
317   \BeforeBeginEnvironment{#2}{%
318     \renewcommand*{\LWR@mdthisenv}{md#2}%
319     \begin{mdframed}[#1]}%
320   \AfterEndEnvironment{#2}{%
321     \end{mdframed}}%
322 }

```



---

File 135 **lwarp-memhfixc.sty**

§ 222    Package **memhfixc**

Pkg    **memhfixc**    memhfixc is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{memhfixc}

---

File 136 **lwarp-metalogo.sty**

§ 223    Package **metalogo**

*(Emulates or patches code by ANDREW GILBERT MOSCHOU.)*

Pkg    **metalogo**    metalogo is emulated.

**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]{}  
 8

---

File 137 **lwarp-microtype.sty**

§ 224    Package **microtype**

*(Emulates or patches code by R SCHLICHT.)*

Pkg    **microtype**    microtype is pre-loaded by lwarp. All user options and macros are ignored and disabled.

**for HTML output:**    Discard all options for lwarp-microtype:

1 \LWR@ProvidesPackageDrop{microtype}

2 \DeclareDocumentCommand{\DeclareMicrotypeSet}{o m m}{}  
 3

---

```

3 \DeclareDocumentCommand{\UseMicrotypeSet}{o m}{}
4 \DeclareDocumentCommand{\DeclareMicrotypeSetDefault}{o m}{}
5 \DeclareDocumentCommand{\SetProtrusion}{o m}{}
6 \DeclareDocumentCommand{\SetExpansion}{o m m}{}
7 \DeclareDocumentCommand{\SetTracking}{o m m}{}
8 \DeclareDocumentCommand{\SetExtraKerning}{o m m}{}
9 \DeclareDocumentCommand{\SetExtraSpacing}{o m m}{}
10 \DeclareDocumentCommand{\DisableLigatures}{o m}{}
11 \DeclareDocumentCommand{\DeclareCharacterInheritance}{o m m}{}
12 \DeclareDocumentCommand{\DeclareMicrotypeVariants}{m}{}
13 \DeclareDocumentCommand{\DeclareMicrotypeAlias}{m m}{}
14 \DeclareDocumentCommand{\LoadMicrotypeFile}{m}{}
15 \DeclareDocumentCommand{\DeclareMicrotypeBabelHook}{m m}{}
16 \DeclareDocumentCommand{\microtypesetup}{m}{}
17 \DeclareDocumentCommand{\microtypecontext}{m}{}
18 \DeclareDocumentCommand{\textmicrotypecontext}{m m}{#2}
19 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
20 \DeclareDocumentCommand{\lsstyle}{}{}
21 \DeclareDocumentCommand{\textls}{o +m}{}
22 \DeclareDocumentCommand{\slig}{m}{#1}
23 }
24 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
25 \def\DeclareMicrotypeVariants#1#{\@gobble}
26 \@onlypreamble\DeclareMicrotypeSet
27 \@onlypreamble\UseMicrotypeSet
28 \@onlypreamble\DeclareMicrotypeSetDefault
29 \@onlypreamble\DisableLigatures
30 \@onlypreamble\DeclareMicrotypeVariants
31 \@onlypreamble\DeclareMicrotypeBabelHook

```

---

File 138 **lwarp-midfloat.sty**

§ 225      Package **midfloat**

*(Emulates or patches code by SIGITAS TOLUŠIS.)*

Pkg midfloat midfloat is emulated.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{midfloat}

2 \newenvironment{strip}[1] [] {}{}
3 \newskip\stripsep

```

---

File 139 **lwarp-midpage.sty**

§ 226      Package **midpage**

Pkg    midpage    midpage is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{midpage}

2 \newenvironment{midpage}

3 {\begin{BlockClass}[\LWR@origmbox{margin-top:6ex} ; \LWR@origmbox{margin-bottom:6ex}]{midpage}}

4 {\end{BlockClass}}

---

File 140 **lwarp-morefloats.sty**

§ 227      Package **morefloats**

Pkg    morefloats    morefloats is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{morefloats}

---

File 141 **lwarp-moreverb.sty**

§ 228      Package **moreverb**

*(Emulates or patches code by ROBIN FAIRBAIRNS.)*

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@verbatiminput\@verbatiminput
13
14 \renewcommand{\@verbatiminput}[2][]{%
15 \LWR@forcenewpage
16 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
17 \LWRMV@orig@verbatiminput[#1]{#2}%
18 \LWR@afterendverbatim%
19 }
20
21 \BeforeBeginEnvironment{listing}{%
22 \LWR@forcenewpage
23 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
24 }
25
26 \AfterEndEnvironment{listing}{%
27 % \unskip\LWR@origvspace*{-\baselineskip}%
28 \LWR@afterendverbatim%
29 }
30
31 \BeforeBeginEnvironment{listingcont}{%
32 \LWR@forcenewpage
33 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
34 }
35
36 \AfterEndEnvironment{listingcont}{%
37 % \unskip\LWR@origvspace*{-\baselineskip}%
38 \LWR@afterendverbatim%
39 }

40 \LetLtxMacro\LWRMV@@listinginput\@listinginput
41
42 \renewcommand{\@listinginput}[3][]{%
43 \LWR@forcenewpage
44 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
45 \LWRMV@@listinginput[#1]{#2}{#3}
46 \LWR@afterendverbatim%
47 }
48
49
50 \renewenvironment*{boxedverbatim}
51 {
52 \LWR@forcenewpage
53 \LWR@atbeginverbatim{boxedverbatim}\unskip\LWR@origvspace*{-2\baselineskip}%
54 \verbatim%
55 }
56 {
57 \endverbatim%

```

---

```

58 \unskip%
59 \LWR@afterendverbatim%
60 }
61
62

63 \end{warpHTML}

```

---

File 142 **lwarp-morewrites.sty**

§ 229     Package **morewrites**

Pkg    **morewrites**    Error if morewrites is loaded after lwarp.

Discard all options for lwarp-morewrites:

**for HTML output:**    1 \LWR@ProvidesPackageDrop{morewrites}

                          2 \LWR@loadbefore{morewrites}

---

File 143 **lwarp-mparhack.sty**

§ 230     Package **mparhack**

Pkg    **mparhack**    Ignored.

**for HTML output:**    Discard all options for lwarp-mparhack:

                          1 \LWR@ProvidesPackageDrop{mparhack}

---

File 144 **lwarp-multicol.sty**

§ 231     Package **multicol**

*(Emulates or patches code by FRANK MITTELBACH.)*

Pkg    **multicol**    multicol is emulated.

**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 * {\langle numcols \rangle} [\langle heading \rangle]
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 145 **lwarp-multirow.sty**

§ 232     Package **multirow**

*(Emulates or patches code by PIET VAN OOSTRUM, ØYSTEIN BACHE, JERRY LEICHTER.)*

Pkg    multirow    multirow is emulated during HTML output, and used as-is while inside a lateximage.

In a lateximage, the original print-mode versions are temporarily restored by `\LWR@restoreorigformatting`.

See section 64.19 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}{}%

§ 232.1    **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@htmltag{td rowspan="#2" %
```

The vertical alignment, if given:

```
10 \IfValueT{#1}{%
11 \ifstrequal{#1}{b}{style="\LWR@origmbox{vertical-align:bottom}" }{}%
12 \ifstrequal{#1}{t}{style="\LWR@origmbox{vertical-align:top}" }{}%
13 }%
```

The left/right border, if given:

```
14 \ifdefvoid{\LWR@multirowborder}{\}%
15 style="\LWR@origmbox{border-\LWR@multirowborder:} 2px dotted black ; %
16 \LWR@origmbox{padding-\LWR@multirowborder:} 2px" %
17 }%
```

A class adds the column spec and the rule:

```
18 class="td%
```

Append this column's spec:

```
19 \StrChar{\LWR@tablecolspec}{\arabic{\LWR@tablecolindex}}%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag. Also add the vertical bar class.

```
20 \LWR@addcmidruletrim%
21 \LWR@addleftmostbartag%
22 \LWR@printbartag{\arabic{\LWR@tablecolindex}}%
23 "%
```

```
24 \LWR@tdstartstyles%
25 \LWR@addcmidrulewidth%
26 \LWR@addtabularrulecolors%
27 \LWR@tdendstyles%
28 }%
```

The column's < spec:

```
29 \LWR@gettexpararray{\LWR@colbefore spec}{\arabic{\LWR@tablecolindex}}%
```

While printing the text, redefine \ to generate a new line

```
30 \begingroup\LetLtxMacro{\}\{\LWR@endoffline}\#6\endgroup%
31 \LWR@stoppars%
32 \global\boolfalse{\LWR@intabularmetadata}%
33 \renewcommand{\LWR@multirowborder}{\}%
34 \LWR@traceinfo{*** multirow done}%
35 }%
```



## § 232.2 Combined multicolumn and multirow

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

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

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

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

⚠ **empty cells**

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

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

```
\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{\arabic{LWR@tablecolindex}}{#1}
```

Create the multicolumn/multirow tag:

```
38 \begingroup%
39 \LetLtxMacro{\}\{LWR@endofline}%
40 \LWR@domulticolumn[#3][#4]{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#8}%
41 \endgroup%
```

Move to the next  $\LaTeX$  column:

```
42 \addtocounter{LWR@tablecolindex}{#1}%
43 \addtocounter{LWR@tablecolindex}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
44 \booltrue{LWR@skipatbang}%
45 }
```

---

```

46 \appto{\LWR@restoreorigformatting}{%
47 \LetLtxMacro\multirow\LWR@origmultirow%
48 \renewcommand{\multicolumnrow}{\LWR@origmulticolumnrow}%
49 }

```

---

File 146 **lwarp-multitoc.sty**

§ 233     Package **multitoc**

Pkg    multitoc    multitoc is ignored.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{multitoc}

2 \newcommand{\multicolumntoc}{2}
3 \newcommand{\multicolumnlot}{2}
4 \newcommand{\multicolumnlof}{2}
5 \newcommand*{\immediateaddtocontents}[2]{ }

```

---

File 147 **lwarp-nameref.sty**

§ 234     Package **nameref**

Pkg    nameref    nameref is emulated by lwarp.

**for HTML output:**    Discard all options for lwarp-nameref:

```

1 \typeout{Using the lwarp html version of package 'nameref' -- discarding options.}
2 \typeout{    Are not using ProvidesPackage, so that other packages}
3 \typeout{    do not attempt to patch lwarp's version of 'nameref'.}
4 \DeclareOption*{}
5 \ProcessOptions\relax

```

---

File 148 **lwarp-natbib.sty**

§ 235     Package **natbib**

*(Emulates or patches code by PATRICK W. DALY.)*

Pkg    natbib    natbib is patched for use by lwarp.

**for HTML output:**    1 \LWR@ProvidesPackagePass{natbib}

Replace math  $<$  and  $>$  with `\textless` and `\textgreater`:

A macro to compare:

```
2 \newcommand{\LWRNB@NAT@open}{<}
```

To patch `\NAT@open` and `\NAT@close`

```
3 \newcommand{\LWRNB@patchnatbibopenclose}{
4 \ifdefstrequal{\NAT@open}{\LWRNB@NAT@open}
5 {
6   \renewcommand{\NAT@open}{\textless}
7   \renewcommand{\NAT@close}{\textgreater}
8 }{}}
9 }
```

Do it now in case `angle` was selected as an option:

```
10 \LWRNB@patchnatbibopenclose
```

Also patch `\setcitestyle` to patch after settings are made:

```
11 \let\LWRNB@origsetcitestyle\setcitestyle
12
13 \renewcommand{\setcitestyle}[1]{%
14 \LWRNB@origsetcitestyle{#1}%
15 \LWRNB@patchnatbibopenclose%
16 }
```

---

File 149 **lwarp-needspace.sty**

§ 236     Package **needspace**

*(Emulates or patches code by PETER WILSON.)*

Pkg    **needspace**    needspace is not used during HTML conversion.

**for HTML output:**    Discard all options for `lwarp-needspace`:

```
1 \LWR@ProvidesPackageDrop{needspace}
2
3 \DeclareDocumentCommand{\needspace}{m}{}
4 \DeclareDocumentCommand{\Needspace}{s m}{}
5 }
```

---

File 150 **lwarp-newclude.sty**

§ 237     Package **newclude**

Pkg   newclude   Error if newclude is loaded after lwarp.

Discard all options for lwarp-newclude:

**for HTML output:**     1 \LWR@ProvidesPackageDrop{newclude}

                         2 \LWR@loadbefore{newclude}

---

File 151 **lwarp-newunicodechar.sty**

§ 238     Package **newunicodechar**

Pkg   newunicodechar   Error if newunicodechar is loaded after lwarp.

Discard all options for lwarp-newunicodechar:

**for HTML output:**     1 \LWR@ProvidesPackageDrop{newunicodechar}

                         2 \LWR@loadbefore{newunicodechar}

---

File 152 **lwarp-nextpage.sty**

§ 239     Package **nextpage**
*(Emulates or patches code by PETER WILSON.)*

Pkg   nextpage   nextpage is nullified.

**for HTML output:**     Discard all options for lwarp-nextpage.

1 \LWR@ProvidesPackageDrop{nextpage}

2 \DeclareDocumentCommand{\cleartoevenpage}{o}{}  
3 \DeclareDocumentCommand{\movetoevenpage}{o}{}  
4 \DeclareDocumentCommand{\cleartooddpage}{o}{}  
5 \DeclareDocumentCommand{\movetooddpage}{o}{}

---

File 153 **lwarp-nonfloat.sty**

§ 240     Package **nonfloat**

*(Emulates or patches code by KAI RASCHER.)*

Pkg   nonfloat   nonfloat is emulated.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{nonfloat}

                         2 \LetLtxMacro\topcaption\caption

                         3 \newcommand{\figcaption}{\def\@capttype{figure}\caption}

                         4 \newcommand{\tabcaption}{\def\@capttype{table}\topcaption}

                         5 \newenvironment{narrow}[2]{\{\}}

---

File 154 **lwarp-nonumonpart.sty**

§ 241     Package **nonumonpart**

Pkg   nonumonpart   nonumonpart is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{nonumonpart}

---

File 155 **lwarp-nopageno.sty**

§ 242     Package **nopageno**

Pkg   nopageno   nopageno is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{nopageno}

---

File 156 **lwarp-nowidow.sty**

§ 243     Package **nowidow**

*(Emulates or patches code by RAPHAËL PINSON.)*

Pkg nowidow nowidow is not used during HTML conversion.

Discard all options for lwarp-nowidow:

for HTML output: 1 \LWR@ProvidesPackageDrop{nowidow}

\nowidow [\langle lines \rangle]  
\setnowidow [\langle lines \rangle]  
  
2 \newcommand\*{\nowidow}[1] [] {}  
3 \newcommand\*{\setnowidow}[1] [] {}

\noclub [\langle lines \rangle]  
\setnoclub [\langle lines \rangle]  
  
4 \newcommand\*{\noclub}[1] [] {}  
5 \newcommand\*{\setnoclub}[1] [] {}

File 157 lwarp-nttheorem.sty

§ 244

Package

nttheorem

(Emulates or patches code by WOLFGANG MAY, ANDREAS SCHEDLER.)

Pkg nttheorem nttheorem is patched for use by lwarp.

Table 13: Ntheorem package — CSS styling of theorems and proofs

**Theorem:** <div> of class theorembody<theoremstyle>

**Theorem Header:** <span> of class theoremheader<style>


where <theoremstyle> is plain, break, etc.

§ 244.1

Limitations

 Font control

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

 Equation numbering

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

## § 244.2 Options

Options `amsthm` or `standard` choose which set of theorems and proofs to initialize.

### Disabled options

The options `thmmarks` and `amsmath` are disabled, since they heavily modify the underlying math code. Theorem marks are emulated. The AMS-math modifications are not done.

Option `thref` is disabled because `cleveref` functions are used instead. `\thref` is emulated.

Option `hyperref` is disabled because `lwarp` emulated `hyperref`.

for HTML output:

Some disabled options:

```

1 \DeclareOption{thref}{}
2
3
4 \newbool{LWR@theoremmarks}
5 \boolfalse{LWR@theoremmarks}
6
7 \DeclareOption{thmmarks}{
8 \booltrue{LWR@theoremmarks}
9 \newif\ifsetendmark\setendmarktrue
10 }
11
12
13 \newbool{LWR@theoremamsthm}
14 \boolfalse{LWR@theoremamsthm}
15
16 \DeclareOption{amsthm}{\booltrue{LWR@theoremamsthm}}
17
18
19 \DeclareOption{amsmath}{}
20 \DeclareOption{hyperref}{}
21
22 \LWR@ProvidesPackagePass{ntheorem}

```

## § 244.3 Remembering the theorem style

Storage for the style being used for new theorems.

```

23 \newcommand{\LWR@newtheoremstyle}{plain}

24 \AtBeginDocument{
25 \ifpackageloaded{cleveref}{
26 \gdef\@thm#1#2#3{%
27   \if@thmmarks
28     \stepcounter{end\InTheoType ctr}%
29   \fi

```

---

```

30 \renewcommand{\InTheoType}{#1}%
31 \if@thmmarks
32   \stepcounter{curr#1ctr}%
33   \setcounter{end#1ctr}{0}%
34 \fi
35 \refstepcounter[#1]{#2}% <<< cleveref modification
36 \theorem@prework
37   \LWR@forcenewpage% lwarp
38   \BlockClass{theorembody#1}%\LWR@thisthmstyle% lwarp
39   \trivlist % latex's \trivlist, calling latex's \@trivlist unchanged
40   \ifuse@newframeskips % cf. latex.ltx for topsepadd: \@trivlist
41     \ifthm@inframe
42       \thm@topsep\theoreminframepreskipamount
43       \thm@topsepadd\theoreminframepostskipamount
44     \else
45       \thm@topsep\theorempreskipamount
46       \thm@topsepadd\theorempostskipamount
47     \fi
48   \else% oldframeskips
49     \thm@topsep\theorempreskipamount
50     \thm@topsepadd \theorempostskipamount
51     \ifvmode\advance\thm@topsepadd\partopsep\fi
52   \fi
53   \@topsep\thm@topsep
54   \@topsepadd\thm@topsepadd
55   \advance\linewidth -\theorem@indent
56   \advance\linewidth -\theorem@rightindent
57   \advance\@totalleftmargin \theorem@indent
58   \parshape \@ne \@totalleftmargin \linewidth
59   \@ifnextchar [{\@ythm{#1}{#2}{#3}}{\@xthm{#1}{#2}{#3}}
60 }
61 }{% not @ifpackageloaded{cleveref}
62 \gdef\@thm#1#2#3{%
63   \if@thmmarks
64     \stepcounter{end\InTheoType ctr}%
65   \fi
66   \renewcommand{\InTheoType}{#1}%
67   \if@thmmarks
68     \stepcounter{curr#1ctr}%
69     \setcounter{end#1ctr}{0}%
70   \fi
71   \refstepcounter{#2}%
72   \theorem@prework
73     \LWR@forcenewpage% lwarp
74     \BlockClass{theorembody#1}%\LWR@thisthmstyle% lwarp
75     \trivlist % latex's \trivlist, calling latex's \@trivlist unchanged
76     \ifuse@newframeskips % cf. latex.ltx for topsepadd: \@trivlist
77       \ifthm@inframe
78         \thm@topsep\theoreminframepreskipamount
79         \thm@topsepadd\theoreminframepostskipamount

```



```

80     \else
81       \thm@topsep\theorempreskipamount
82       \thm@topsepadd\theorempostskipamount
83     \fi
84   \else% oldframeskips
85     \thm@topsep\theorempreskipamount
86     \thm@topsepadd \theorempostskipamount
87     \ifvmode\advance\thm@topsepadd\partopsep\fi
88   \fi
89   \@topsep\thm@topsep
90   \@topsepadd\thm@topsepadd
91   \advance\linewidth -\theorem@indent
92   \advance\linewidth -\theorem@rightindent
93   \advance\@totalleftmargin \theorem@indent
94   \parshape \@one \@totalleftmargin \linewidth
95   \@ifnextchar[{\@ythm{#1}{#2}{#3}}{\@xthm{#1}{#2}{#3}}
96 }
97 }
98 }% AtBeginDocument

```

Patched to remember the style being used for new theorems:

```

99 \gdef\theoremstyle#1{%
100   \@ifundefined{th@#1}{\@warning
101     {Unknown theoremstyle ‘#1’. Using ‘plain’}%
102     \theorem@style{plain}
103     \renewcommand{\LWR@newtheoremstyle}{plain}% lwarp
104   }%
105   {
106     \theorem@style{#1}
107     \renewcommand{\LWR@newtheoremstyle}{#1}% lwarp
108   }
109 }

```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

110
111 \gdef\xnthm#1#2[#3]{%
112   \ifthm@tempif
113     \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
114     \expandafter\@ifundefined{c@#1}%
115       {\@definecounter{#1}}{}%
116     \@newctr{#1}[#3]%
117     \expandafter\xdef\csname the#1\endcsname{%
118       \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
119       {\noexpand\csname the\theoremnumbering\endcsname{#1}}}%
120     \expandafter\gdef\csname mkheader@#1\endcsname
121       {\csname setparams@#1\endcsname

```

```

122     \@thm{#1}{#1}{#2}
123     }%
124     \global\@namedef{end#1}{\@endtheorem}
125     \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\csuse{LWR@thmstyle#1}}}% lwarp
126 \fi
127 }
128
129 \gdef\@ynthm#1#2{%
130   \ifthm@tempif
131     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
132     \expandafter\@ifundefined{c@#1}%
133       {\@definecounter{#1}}{}%
134     \expandafter\xdef\csname the#1\endcsname
135       {\noexpand\csname\the\theoremnumbering\endcsname{#1}}%
136     \expandafter\gdef\csname mkheader@#1\endcsname
137       {\csname setparms@#1\endcsname
138         \@thm{#1}{#1}{#2}
139         }%
140     \global\@namedef{end#1}{\@endtheorem}
141     \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\csuse{LWR@thmstyle#1}}}% lwarp
142 \fi
143 }
144
145 \gdef\@othm#1[#2]#3{%
146   \@ifundefined{c@#2}{\@nocounterr{#2}}%
147   {\ifthm@tempif
148     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
149     \global\@namedef{the#1}{\@nameuse{the#2}}%
150     \expandafter\protected@xdef\csname num@addtheorem#1\endcsname{%
151       \noexpand\@num@addtheorem#1{#1}{#3}}%
152     \expandafter\protected@xdef\csname nonum@addtheorem#1\endcsname{%
153       \noexpand\@nonum@addtheorem#1{#1}{#3}}%
154     \theoremkeyword{#3}%
155     \expandafter\protected@xdef\csname #1Keyword\endcsname
156       {\the\theoremkeyword}%
157     \expandafter\gdef\csname mkheader@#1\endcsname
158       {\csname setparms@#1\endcsname
159         \@thm{#1}{#2}{#3}
160         }%
161     \global\@namedef{end#1}{\@endtheorem}
162     \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\csuse{LWR@thmstyle#1}}}% lwarp
163 \fi}
164 }

```

#### § 244.4 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.

```

165 \newcommand{\LWR@inctheorem}{%
166 \addtocounter{LWR@thisautoid}{1}%
167 \LWR@stoppars%
168 \LWR@htmltag{a id="\LWR@origmbox{autoid-\arabic{LWR@thisautoid}}"}\LWR@htmltag{/a}%
169 \LWR@startpars%
170 }
```

#### § 244.5 \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>.

```

171 \gdef\newtheoremstyle#1#2#3{%
172   \expandafter\@ifundefined{th@#1}%
173   {\expandafter\gdef\csname th@#1\endcsname{%
174     \def\@begintheorem###1###2{%
175       \LWR@inctheorem% lwarp
176       #2}%
177     \def\@opargbegintheorem###1###2###3{%
178       \LWR@inctheorem% lwarp
179       #3}%
180   }%
181   }%
182 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
183 }
```

#### § 244.6 Standard styles

```

184 \renewtheoremstyle{plain}%
185 {\item[
186   \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]]}%
187 {\item[
188   \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]]}
189
190 \renewtheoremstyle{break}%
191 {\item[
192   \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
193   ]}%

```

```

194 {\item[
195   \InlineClass{theoremheaderbreak}%
196   {##1\ ##2\ (##3)\theorem@separator}\newline
197   ]}
198
199 \renewtheoremstyle{change}%
200 {\item[
201   \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]]%
202 {\item[
203   \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]]}
204
205 \renewtheoremstyle{changebreak}%
206 {\item[
207   \InlineClass{theoremheaderchangebreak}%
208   {##2\ ##1\theorem@separator}\newline
209   ]}%
210 {\item[
211   \InlineClass{theoremheaderchangebreak}%
212   {##2\ ##1\ (##3)\theorem@separator}\newline
213   ]}
214
215 \renewtheoremstyle{margin}%
216 {\item[
217   \InlineClass{theoremheadermargin}{##2 \quad ##1\theorem@separator}
218   ]}%
219 {\item[
220   \InlineClass{theoremheadermargin}{##2 \quad ##1\ (##3)\theorem@separator}
221   ]}
222
223 \renewtheoremstyle{marginbreak}%
224 {\item[
225   \InlineClass{theoremheadermarginbreak}%
226   {##2 \quad ##1\theorem@separator}\newline
227   ]}%
228 {\item[
229   \InlineClass{theoremheadermarginbreak}%
230   {##2 \quad ##1\ (##3)\theorem@separator}\newline
231   ]}
232
233 \renewtheoremstyle{nonumberplain}%
234 {\item[
235   \InlineClass{theoremheaderplain}{##1\theorem@separator}]]%
236 {\item[
237   \InlineClass{theoremheaderplain}{##1\ (##3)\theorem@separator}]]}
238
239 \renewtheoremstyle{nonumberbreak}%
240 {\item[
241   \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline
242   ]}%
243 {\item[

```

```

244 \InlineClass{theoremheaderbreak}{##1\ (##3)\theorem@separator}\newline
245   ]}
246
247 \renewtheoremstyle{empty}%
248   {\item[]}%
249   {\item[
250     \InlineClass{theoremheaderplain}{##3}]}
251
252 \renewtheoremstyle{emptybreak}%
253   {\item[]}%
254   {\item[
255     \InlineClass{theoremheaderplain}{##3}] \ \newline}

```

### § 244.7 Additional objects

The following manually adjust the css for the standard configuration objects which are not a purely plain style:

```

256 \ifbool{LWR@ntheoremamsthm}{-}{%

```

Upright text via CSS:

```

257 \newtheoremstyle{plainupright}%
258   {\item[
259     \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
260   {\item[
261     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}

```

Upright text and small caps header via CSS:

```

262 \newtheoremstyle{nonumberplainuprightsc}%
263   {\item[
264     \InlineClass{theoremheadersc}{##1\theorem@separator}]}%
265   {\item[
266     \InlineClass{theoremheadersc}{##1\ (##3)\theorem@separator}]}
267 }% not amsthm

```

### § 244.8 Renewed standard configuration

The following standard configuration is renewed using the new css:

```

268 \ifbool{LWR@ntheoremamsthm}{-}{%

269 \ifx\thm@usestd\@undefined
270 \else
271   \theoremnumbering{arabic}
272   \theoremstyle{plain}
273   \RequirePackage{latexsym}

```

---

```

274 \theoremsymbol{\Box}
275 \theorembodyfont{\itshape}
276 \theoremheaderfont{\normalfont\bfseries}
277 \theoremseparator{}
278 \renewtheorem{Theorem}{Theorem}
279 \renewtheorem{theorem}{Theorem}
280 \renewtheorem{Satz}{Satz}
281 \renewtheorem{satz}{Satz}
282 \renewtheorem{Proposition}{Proposition}
283 \renewtheorem{proposition}{Proposition}
284 \renewtheorem{Lemma}{Lemma}
285 \renewtheorem{lemma}{Lemma}
286 \renewtheorem{Korollar}{Korollar}
287 \renewtheorem{korollar}{Korollar}
288 \renewtheorem{Corollary}{Corollary}
289 \renewtheorem{corollary}{Corollary}
290
291 \theoremstyle{plainupright}
292 \theorembodyfont{\upshape}
293 \theoremsymbol{\HTMLUnicode{25A1}}% UTF-8 white box
294 \renewtheorem{Example}{Example}
295 \renewtheorem{example}{Example}
296 \renewtheorem{Beispiel}{Beispiel}
297 \renewtheorem{beispiel}{Beispiel}
298 \renewtheorem{Bemerkung}{Bemerkung}
299 \renewtheorem{bemerkung}{Bemerkung}
300 \renewtheorem{Anmerkung}{Anmerkung}
301 \renewtheorem{anmerkung}{Anmerkung}
302 \renewtheorem{Remark}{Remark}
303 \renewtheorem{remark}{Remark}
304 \renewtheorem{Definition}{Definition}
305 \renewtheorem{definition}{Definition}
306
307 \theoremstyle{nonumberplainuprightsc}
308 \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
309 \renewtheorem{Proof}{Proof}
310 \renewtheorem{proof}{Proof}
311 \renewtheorem{Beweis}{Beweis}
312 \renewtheorem{beweis}{Beweis}
313 \qedsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
314
315 \theoremsymbol{}
316 \fi
317}% not amsthm

```

§ 244.9 **amsthm option**

Only if the amsthm option was given:

```

318 \ifbool{LWR@theoremamsthm}{
319
320 \gdef\th@plain{%
321   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
322   \def\@begintheorem##1##2{%
323     \LWR@intheorem% lwarp
324     \item[
325   \InlineClass{theoremheaderplain}{##1\ ##2.}
326   ]}%
327   \def\@opargbegintheorem##1##2##3{%
328     \LWR@intheorem% lwarp
329     \item[
330   \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
331   ]}}
332
333 \gdef\th@nonumberplain{%
334   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
335   \def\@begintheorem##1##2{%
336     \LWR@intheorem% lwarp
337     \item[
338   \InlineClass{theoremheaderplain}{##1.}
339   ]}%
340   \def\@opargbegintheorem##1##2##3{%
341     \LWR@intheorem% lwarp
342     \item[
343   \InlineClass{theoremheaderplain}{##1\ (##3).}
344   ]}}
345
346 \gdef\th@definition{%
347   \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
348   \def\@begintheorem##1##2{%
349     \LWR@intheorem% lwarp
350     \item[
351   \InlineClass{theoremheaderdefinition}{##1\ ##2.}
352   ]}%
353   \def\@opargbegintheorem##1##2##3{%
354     \LWR@intheorem% lwarp
355     \item[
356   \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
357   ]}}
358
359 \gdef\th@nonumberdefinition{%
360   \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
361   \def\@begintheorem##1##2{%
362     \LWR@intheorem% lwarp
363     \item[

```

---

```

364 \InlineClass{theoremheaderdefinition}{##1.}
365 }}%
366 \def\@opargbegintheorem##1##2##3{%
367     \LWR@intheorem% lwarp
368     \item[
369 \InlineClass{theoremheaderdefinition}{##1\ (##3).}
370     ]}}
371
372 \gdef\th@remark{%
373 \def\theorem@headerfont{\itshape}\normalfont%
374 \def\@begintheorem##1##2{%
375     \LWR@intheorem% lwarp
376     \item[
377 \InlineClass{theoremheaderremark}{##1\ ##2.}
378     ]}%
379 \def\@opargbegintheorem##1##2##3{%
380     \LWR@intheorem% lwarp
381     \item[
382 \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
383     ]}}
384
385 \gdef\th@nonumberremark{%
386 \def\theorem@headerfont{\itshape}\normalfont%
387 \def\@begintheorem##1##2{%
388     \LWR@intheorem% lwarp
389     \item[
390 \InlineClass{theoremheaderremark}{##1.}
391     ]}%
392 \def\@opargbegintheorem##1##2##3{%
393     \LWR@intheorem% lwarp
394     \item[
395 \InlineClass{theoremheaderremark}{##1\ (##3).}
396     ]}}
397
398 \gdef\th@proof{%
399 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
400 \def\@begintheorem##1##2{%
401     \LWR@intheorem% lwarp
402     \item[
403 \InlineClass{theoremheaderproof}{##1.}
404     ]}%
405 \def\@opargbegintheorem##1##2##3{%
406     \LWR@intheorem% lwarp
407     \item[
408 \InlineClass{theoremheaderproof}{##1\ (##3).}
409     ]}}
410
411
412
413 \newcounter{proof}%

```



```

414 \if@thmmarks
415   \newcounter{currproofctr}%
416   \newcounter{endproofctr}%
417 \fi
418
419 \gdef\proofSymbol{\openbox}
420
421 \newcommand{\proofname}{Proof}
422
423 \newenvironment{proof}[1][\proofname]{
424   \th@proof
425   \def\theorem@headerfont{\itshape}%
426   \normalfont
427   \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
428   \@thm{proof}{proof}{#1}
429 }%
430 {\@endtheorem}
431
432 }{}% amsthm option

```

## § 244.10 Ending a theorem

Patched for CSS:

```

433 \let\LWR@origendtheorem\@endtheorem
434 \renewcommand{\@endtheorem}{%
435 \ifbool{LWR@theoremmarks}{%
436   \ifsetendmark%
437   \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
438   \setendmarkfalse%
439   \fi%
440 }{}%
441 \LWR@origendtheorem% also does \@endtrivlist
442 \ifbool{LWR@theoremmarks}{\global\setendmarktrue}{}%
443 \endBlockClass%
444 }

```

## § 244.11 \NoEndMark

```

445 \gdef\NoEndMark{\global\setendmarkfalse}

```

## § 244.12 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}

446 \renewcommand{\thm@thmline@noname}[4]{%
447 \hypertocfloat{1}{theorem}{thm}{#2 #3}{}%

```

```

448 }
449
450 \renewcommand{\thm@@thmline@name}[4]{%
451 \hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
452 }

```

This was redefined by ntheorem when loaded, so it is now redefined for lwarp:

```

453 \def\thm@@thmline{\thm@@thmline@name}

```

Patch for CSS:

```

454 \def\listtheorems#1{
455 \LWR@html@elementclass{nav}{lothm}%
456 \begingroup
457 \c@tocdepth=-2%
458 \def\thm@list{#1}\thm@processlist
459 \endgroup
460 \LWR@html@elementclassend{nav}{lothm}%
461 }

```

### § 244.13 Symbols

Proof QED symbol:

```

462 \newcommand{\qed}{\quad\the\qedsymbol}
463
464 \AtBeginDocument{
465 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
466 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
467 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
468 }

```

### § 244.14 Cross-referencing

`\thref {<label>}`

```

469 \newcommand*{\thref}[1]{\cref{#1}}

```


---

File 158 **lwarp-overpic.sty**

### § 245 Package **overpic**

*(Emulates or patches code by ROLF NIEPRASCHK.)*

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 76.2 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   \begin{lateximage}%
7   \fontsize{\overpicfontsize}{\overpicfontskip}%
8   \selectfont%
9 }
10
11 \AfterEndEnvironment{overpic}{\end{lateximage}}
12
13 \BeforeBeginEnvironment{Overpic}{%
14   \begin{lateximage}%
15   \fontsize{\overpicfontsize}{\overpicfontskip}%
16   \selectfont%
17 }
18
19 \AfterEndEnvironment{Overpic}{\end{lateximage}}
```

---

File 159 **lwarp-pagegrid.sty**

§ 246    Package **pagegrid**

Pkg    pagegrid    pagegrid is ignored.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{pagegrid}

2 \newcommand*{\pagegridsetup}[1]{}
```

---

File 160 **lwarp-pagenote.sty**

§ 247    Package **pagenote**

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

---

**for HTML output:**    1 \DeclareOption{page}{}\br/>
                          2 \LWR@ProvidesPackagePass{pagenote}

---

File 161 **lwarp-pagesel.sty**

§ 248    Package **pagesel**

Pkg    pagesel    pagesel is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{pagesel}

---

File 162 **lwarp-paralist.sty**

§ 249    Package **paralist**

*(Emulates or patches code by BERND SCHANDL.)*

Pkg    paralist    paralist is supported with minor changes.

**for HTML output:**    1 \LWR@ProvidesPackagePass{paralist}

---

The compact environments are identical to the regular ones:

```
2 \AtBeginEnvironment{compactitem}{\LWR@itemizestart}
3 \AtBeginEnvironment{compactenum}{\LWR@enumeratestart}
4 \AtBeginEnvironment{compactdesc}{\LWR@descriptionstart}
5 \AtEndEnvironment{compactitem}{\LWR@listend}
6 \AtEndEnvironment{compactenum}{\LWR@listend}
7 \AtEndEnvironment{compactdesc}{\LWR@listend}
```

For the inline environments, revert \item to its original print-mode version:

```
8 \AtBeginEnvironment{inparaitem}{\LetLtxMacro\item\LWR@origitem}
9 \AtBeginEnvironment{inparaenum}{\LetLtxMacro\item\LWR@origitem}
10 \AtBeginEnvironment{inparadesc}{\LetLtxMacro\item\LWR@origitem}
```

Manual formatting of the description labels:

```
11 \def\paradescriptionlabel#1{{\normalfont\textbf{#1}}}
```

---

File 163 **lwarp-parskip.sty**

§ 250     Package **parskip**

Pkg    parskip    parskip is ignored.

for HTML output:    Discard all options for lwarp-parskip.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

---

File 164 **lwarp-pbox.sty**

§ 251     Package **pbox**

*(Emulates or patches code by SIMON LAW.)*

Pkg    pbox    pbox is emulated.

for HTML output:    1 \LWR@ProvidesPackageDrop{pbox}

```

2 \NewDocumentCommand{\pbox}{0{t} o 0{t} m +m}{%
3 \booltrue{LWR@minipagefullwidth}%
4 \parbox[#1] [#2] [#3] {#4} {#5}%
5 }
6
7 \newcommand{\settominwidth}[3] [\columnwidth] {%
8 \settowidth{#2} {#3}%
9 }
10
11 \newcommand{\widthofpbox}[1] {%
12 \widthof{#1}%
13 }
```

---

File 165 **lwarp-pdfscape.sty**

§ 252     Package **pdfscape**

Pkg    pdfscape    pdfscape is ignored.

for HTML output:

Discard all options for lwarp-pdfjscape:

```
1 \LWR@ProvidesPackageDrop{pdfjscape}
```

---

File 166 **lwarp-pdfrender.sty**

§ 253     Package **pdfrender**

Pkg   pdfrender   pdfrender is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{pdfrender}

```
2 \newcommand*{\pdfrender}[1]{}
```

```
3 \newcommand{\textpdfrender}[2]{#2}
```

---

File 167 **lwarp-pdfsync.sty**

§ 254     Package **pdfsync**

*(Emulates or patches code by J. LAURENS.)*

Pkg   pdfsync   Emulated.

**for HTML output:**    Discard all options for lwarp-pdfsync:

```
1 \LWR@ProvidesPackageDrop{pdfsync}
```

```
2 \newcommand*{\pdfsync}{}
3 \newcommand*{\pdfsyncstart}{}
4 \newcommand*{\pdfsyncstop}{}


```

---

File 168 **lwarp-pfnote.sty**

§ 255     Package **pfnote**

Pkg   pfnote   pfnote is emulated.

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

---

**for HTML output:** 1 \LWR@ProvidesPackageDrop{pfnote}

---

File 169 **lwarp-phfqit.sty**

\$ 256 Package **phfqit**

*(Emulates or patches code by PHILIPPE FAIST.)*

Pkg phfqit phfqit is patched for use by lwarp.

**for HTML output:** 1 \LWR@ProvidesPackagePass{phfqit}

2 \LetLtxMacro\LWR@origbitstring\bitstring  
 3  
 4 \renewcommand\bitstring[1]{%  
 5 \InlineClass[%  
 6 text-decoration: overline underline ;  
 7 ]{bitstring}{#1}%  
 8 % \phfqit@bitstring{#1}%  
 9 }  
 10  
 11 \appto{\LWR@restoreorigformatting}{  
 12 \LetLtxMacro\bitstring\LWR@origbitstring  
 13 }

---

File 170 **lwarp-placeins.sty**

\$ 257 Package **placeins**

*(Emulates or patches code by DONALD ARSENEAU.)*

Pkg placeins placeins is not used during HTML conversion.

Discard all options for lwarp-placeins:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{placeins}

2 \newcommand\*{\FloatBarrier}{}  
 3

---

File 171 **lwarp-prelim2e.sty**

§ 258     Package **prelim2e**

*(Emulates or patches code by MARTIN SCHRÖDER.)*

Pkg    **prelim2e**    Emulated.

**for HTML output:**    Discard all options for lwarp-prelim2e:

```
1 \LWR@ProvidesPackageDrop{prelim2e}

2 \newcommand{\PrelimText}{}
3 \newcommand{\PrelimTextStyle}{}
4 \newcommand{\PrelimWords}{}

```

---

File 172 **lwarp-prettyref.sty**

§ 259     Package **prettyref**

*(Emulates or patches code by KEVIN S. RULAND.)*

Pkg    **prettyref**    prettyref is patched for use by lwarp.

**for HTML output:**    1 \LWR@ProvidesPackagePass{prettyref}

```
2 \newreformat{fig}{Figure \ref{#1}}
3 \newreformat{tab}{Table \ref{#1}}

```

---

File 173 **lwarp-preview.sty**

§ 260     Package **preview**

Pkg    **preview**    preview is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{preview}

```
2 \newenvironment{preview}{}{}
3 \newenvironment{nopreview}{}{}

```



---

```

4 \NewDocumentCommand{\PreviewMacro}{s o o +m}{}
5 \NewDocumentCommand{\PreviewEnvironment}{s o o +m}{}
6 \newcommand{\PreviewSnarfEnvironment}[2] [] {}
7 \NewDocumentCommand{\PreviewOpen}{s o}{}
8 \NewDocumentCommand{\PreviewClose}{s o}{}
9 \let\ifPreview\iffalse% \fi for syntax highlighting

```

---

File 174 **lwarp-quotchap.sty**

§ 261 Package **quotchap**

*(Emulates or patches code by KARSTEN TINNEFELD, JAN KLEVER.)*

Pkg quotchap quotchap is emulated.

**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 \begin{BlockClass}[max-width: \LWR@printlength{\LWR@quotchapwidth}]{quotchap}
9 %\begin{minipage}{\LWR@quotchapwidth}
10 \@quotchap
11 %\end{minipage}
12 \end{BlockClass}

```

Deactivate the quote printing:

```

13 \global\let\@printcites\relax
14 }
15
16 \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>

```

17 \setlength{\LWR@quotchapwidth}{#1*2}%
18 \global\LWR@quotchapwidth=\LWR@quotchapwidth%

```

Remember the body, and activate the quote printing:

```
19 \global\let\@quotchap\BODY
20 \global\let\@printcites\@iprintcites%
21 }
```

The quotation author is placed inside a <div> of class qauthor:

```
22 \newcommand{\qauthor}[1]{\begin{BlockClass}{qauthor}{#1}\end{BlockClass}}
```

\qsetcnfont is ignored:

```
23 \newcommand{\qsetcnfont}[1]{}
```

---

File 175 **lwarp-ragged2e.sty**

§ 262     Package **ragged2e**

*(Emulates or patches code by MARTIN SCHRÖDER.)*

Pkg ragged2e ragged2e is not used during HTML conversion.

Discard all options for lwarp-ragged2e:

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{ragged2e}

2 \newcommand*{\Centering}{\centering}
3 \newcommand*{\RaggedLeft}{\raggedleft}
4 \newcommand*{\RaggedRight}{\raggedright}
5 \newcommand*{\justifying}{\}
6 \newlength{\CenteringLeftskip}
7 \newlength{\RaggedLeftLeftskip}
8 \newlength{\RaggedRightLeftskip}
9 \newlength{\CenteringRightskip}
10 \newlength{\RaggedLeftRightskip}
11 \newlength{\RaggedRightRightskip}
12 \newlength{\CenteringParfillskip}
13 \newlength{\RaggedLeftParfillskip}
14 \newlength{\RaggedRightParfillskip}
15 \newlength{\JustifyingParfillskip}
16 \newlength{\CenteringParindent}
17 \newlength{\RaggedLeftParindent}
18 \newlength{\RaggedRightParindent}
19 \newlength{\JustifyingParindent}
20 \newenvironment*{Center}{\center}{\endcenter}
21 \newenvironment*{FlushLeft}{\flushleft}{\endflushleft}
22 \newenvironment*{FlushRight}{\flushright}{\endflushright}
```

---

```
23 \newenvironment*{justify}{\justifying}{\endjustifying}
```

---

File 176 **lwarp-realscripts.sty**

§ 263      Package **realscripts**

*(Emulates or patches code by WILL ROBERTSON.)*

Pkg    realscripts    realscripts is emulated. See lwarp.css for the <span> of class supsubscript.

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{realscripts}

2 \let\realsuperscript\textsuperscript
3 \let\realsubscript\textsubscript
4
5 \let\fakesuperscript\textsuperscript
6 \let\fakesubscript\textsubscript
7
8 \newlength{\subsupersep}
9
10 \newcommand*{\LWR@realscriptsalign}{}
11
12 \newcommand*{\LWR@setrealscriptsalign}[1]{%
13 \renewcommand*{\LWR@realscriptsalign}{}%
14 \ifthenelse{\equal{#1}{c}}{\renewcommand{\LWR@realscriptsalign}{\LWR@origmbox{text-align:center}}
15 \ifthenelse{\equal{#1}{r}}{\renewcommand{\LWR@realscriptsalign}{\LWR@origmbox{text-align:right}}
16 }
17
18 \DeclareDocumentCommand \textsubsuperscript {s O{1} mm} {%
19 \LWR@setrealscriptsalign{#2}%
20 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
21 \textsuperscript{#4}\textsubscript{#3}%
22 }%
23 }
24
25 \DeclareDocumentCommand \textsupersubscript {s O{1} mm} {%
26 \LWR@setrealscriptsalign{#2}%
27 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
28 \textsubscript{#4}\textsuperscript{#3}%
29 }%
30 }
```

---

File 177 **lwarp-reysize.sty**

§ 264     Package **relsize**


*(Emulates or patches code by DONALD ARSENEAU, BERNIE COSELL, MATT SWIFT.)*

Pkg    relsize    relsize is patched for use by lwarp.

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

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

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

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

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{relsize}

2 \let\LWR@origrelsize\relsize
3 \LetLtxMacro\LWR@origlarger\larger
4 \LetLtxMacro\LWR@origsmaller\smaller
5 \let\LWR@relscale\relscale
6 \LetLtxMacro\LWR@origtextlarger\textlarger
7 \LetLtxMacro\LWR@origtextsmaller\textsmaller
8 \let\LWR@textscale\textscale
9
10 \appto{\LWR@restoreorigformatting}{%
11 \let\relsize\LWR@origrelsize%
12 \LetLtxMacro\larger\LWR@origlarger%
13 \LetLtxMacro\smaller\LWR@origsmaller%
14 \let\relscale\LWR@relscale%
15 \LetLtxMacro\textlarger\LWR@origtextlarger%
16 \LetLtxMacro\textsmaller\LWR@origtextsmaller%
17 \let\textscale\LWR@textscale%
18 }
19
20 \newcounter{LWR@relsizetemp}
21
22 \renewcommand*{\relsize}[1]{%
23 \renewcommand*{\larger}[1][1]{%
24 \renewcommand*{\smaller}[1][1]{%
25 \renewcommand*{\relscale}[1]{%
```

---

```

26
27 \renewcommand*{\textlarger}[2][1]{%
28 \setcounter{LWR@relsize}{100+(#1*20)}%
29 \InlineClass[font-size:\arabic{LWR@relsize}\%]{textlarger}{#2}%
30 }
31
32 \renewcommand*{\textsmaller}[2][1]{%
33 \setcounter{LWR@relsize}{100-(#1*20)}%
34 \InlineClass[font-size:\arabic{LWR@relsize}\%]{textsmaller}{#2}%
35 }
36
37 \renewcommand*{\textscale}[2]{%
38 \setcounter{LWR@relsize}{100*\real{#1}}%
39 \InlineClass[font-size:\arabic{LWR@relsize}\%]{textscale}{#2}%
40 }

```

---

File 178 **lwarp-resizegather.sty**

§ 265     Package **resizegather**

Pkg    `resizegather`    `resizegather` is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{resizegather}

                          2 \newcommand\*{\resizegather}{1}{}

---

File 179 **lwarp-romanbar.sty**

§ 266     Package **romanbar**

*(Emulates or patches code by H.-MARTIN MÜNCH.)*

Pkg    `romanbar`    `romanbar` is patched for use by `lwarp`.

An inline class with an overline and underline is used.

**for HTML output:**    1 \LWR@ProvidesPackagePass{romanbar}

                          2 \DeclareRobustCommand{\Roman@bar}[1]{% #1 is in Roman, i.e. MMXII

                          3 \InlineClass[%

                          4     text-decoration: overline underline ;

                          5 ]{romanbar}{#1}%

                          6 }

---

File 180 **lwarp-romanbarpagenumber.sty**

§ 267     Package **romanbarpagenumber**

Pkg   romanbarpagenumber   romanbarpagenumber is ignored.

**for HTML output:**     1 \LWR@ProvidesPackageDrop{romanbarpagenumber}

---

File 181 **lwarp-rotating.sty**

§ 268     Package **rotating**

*(Emulates or patches code by ROBIN FAIRBAIRNS, SEBASTIAN RAHTZ, LEONOR BARROCA.)*

Pkg   rotating   rotating is emulated.

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}{-}{-}  
 12 \let\rotcaption\caption  
 13 \let\@makerotcaption\@makecaption

---

File 182 **lwarp-rotfloat.sty**

§ 269     Package **rotfloat**

*(Emulates or patches code by AXEL SOMMERFELDT.)*

Pkg   rotfloat   rotfloat is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{rotfloat}
2
3 \RequirePackage{float}

```

`\newfloat`  $\{\langle 1: type \rangle\} \{\langle 2: placement \rangle\} \{\langle 3: ext \rangle\} [\langle 4: within \rangle]$

Emulates the `\newfloat` command from the `float` package. Sideways floats are `\let` to the same as regular floats.

“placement” is ignored.

```

4 \RenewDocumentCommand{\newfloat}{m m m o}{%
5 \IfValueTF{#4}%
6 {%
7   \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}%
8 }%
9 {%
10  \DeclareFloatingEnvironment[fileext=#3]{#1}%

11  \DeclareFloatingEnvironment[fileext=#3]{sideways#1}%
12 }%
13 \csletcs{sideways#1}{#1}%
14 \csletcs{endsideways#1}{end#1}%

```

`newfloat` package automatically creates the `\listof` command for new floats, but `float` does not, so remove `\listof` here in case it is manually created later.

```

15 \cslet{listof#1s}\relax%
16 \cslet{listof#1es}\relax%
17 }

```

---

File 183 **lwarp-savetrees.sty**

§ 270     Package **savetrees**

Pkg    **savetrees**    Emulated.

for HTML output:    Discard all options for `lwarp-savetrees`:

```

1 \LWR@ProvidesPackageDrop{savetrees}

```

---

File 184 **lwarp-scalefnt.sty**

§ 271 Package **scalefnt**

*(Emulates or patches code by D. CARLISLE.)*

Pkg scalefnt scalefnt is ignored.

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{scalefnt}

2 \DeclareRobustCommand\scalefont[1]{}

```

---

File 185 **lwarp-schemata.sty**

§ 272 Package **schemata**

*(Emulates or patches code by CHARLES P. SCHAUUM.)*

Pkg schemata schemata is patched for use by lwarp.

**for HTML output:**

```
1 \LWR@ProvidesPackagePass{schemata}

2 \LetLtxMacro\LWR@schemata@origschema\schemata
3 \LetLtxMacro\LWR@schemata@origSchema\Schema
4
5 \renewcommand{\schemata}[3][open]{%
6 \begin{lateximage}%
7 \LWR@orignormalsize
8 \LWR@schemata@origschema[#1]{#2}{#3}%
9 \end{lateximage}%
10 }
11
12 \renewcommand{\Schema}[5][open]{%
13 \begin{lateximage}%
14 \LWR@orignormalsize
15 \LWR@schemata@origSchema[#1]{#2}{#3}{#4}{#5}%
16 \end{lateximage}%
17 }

```



---

File 186 **lwarp-scrextend.sty**

§ 273      Package **scrextend**

Pkg    **scrextend**    scrextend is emulated.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. `\DeclareDocumentCommand` is used to overwrite the koma-script definitions.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{scrextend}

2 \DeclareDocumentCommand{\setkomafont}{m m}{}
3 \DeclareDocumentCommand{\addkomafont}{m m}{}
4 \DeclareDocumentCommand{\usekomafont}{m}{}
5
6 \DeclareDocumentCommand{\usefontofkomafont}{m}{}
7 \DeclareDocumentCommand{\useencodingofkomafont}{m}{}
8 \DeclareDocumentCommand{\usesizeofkomafont}{m}{}
9 \DeclareDocumentCommand{\usefamilyofkomafont}{m}{}
10 \DeclareDocumentCommand{\useseriesofkomafont}{m}{}
11 \DeclareDocumentCommand{\useshapeofkomafont}{m}{}
12
13 \AtBeginDocument{
14 \let\LWR@maketitle\maketitle
15 \DeclareDocumentCommand{\maketitle}{o}{\LWR@maketitle}
16 }
17
18 \DeclareDocumentCommand{\extratitle}{m}{}
19 \DeclareDocumentCommand{\titlehead}{m}{}
20 \DeclareDocumentCommand{\subject}{m}{}
21 \DeclareDocumentCommand{\publishers}{m}{\published{#1}}
22 \DeclareDocumentCommand{\uppertitleback}{m}{}
23 \DeclareDocumentCommand{\lowertitleback}{m}{}
24 \DeclareDocumentCommand{\dedication}{m}{}
25
26 \DeclareDocumentCommand{\ifthispageodd}{m m}{#1}
27
28 \DeclareDocumentCommand{\titlepagestyle}{}{}
29
30 \DeclareDocumentCommand{\cleardoublepageusingstyle}{m}{}
31 \DeclareDocumentCommand{\cleardoubleemptypage}{}{}
32 \DeclareDocumentCommand{\cleardoubleplainpage}{}{}
33 \DeclareDocumentCommand{\cleardoublestandardpage}{}{}
34 \DeclareDocumentCommand{\cleardoubleoddpaper}{}{}

```

```

35 \DeclareDocumentCommand{\cleardoubleoddpagesusingstyle}{m}{}
36 \DeclareDocumentCommand{\cleardoubleoddeftypage}{}{}
37 \DeclareDocumentCommand{\cleardoubleoddplainpage}{}{}
38 \DeclareDocumentCommand{\cleardoubleoddstandardpage}{}{}
39 \DeclareDocumentCommand{\cleardoubleevenpage}{}{}
40 \DeclareDocumentCommand{\cleardoubleevenpagesusingstyle}{m}{}
41 \DeclareDocumentCommand{\cleardoubleevenemptypage}{}{}
42 \DeclareDocumentCommand{\cleardoubleevenplainpage}{}{}
43 \DeclareDocumentCommand{\cleardoubleevenstandardpage}{}{}
44
45 \DeclareDocumentCommand{\multiplefootnoteseparator}{}{%
46   \begingroup\let\thefootnotemark\multfootsep\makefnmark\endgroup
47 }
48
49 \DeclareDocumentCommand{\multfootsep}{}{,}
50
51 \DeclareDocumentCommand{\footref}{m}{%
52   \begingroup
53     \unrestored@protected@xdef\@thefnmark{\ref{#1}}%
54   \endgroup
55   \@footnotemark
56 }
57
58 \DeclareDocumentCommand{\deffootnote}{o m m m}{}
59 \DeclareDocumentCommand{\deffootnotemark}{m}{}
60 \DeclareDocumentCommand{\setfootnoterule}{o m}{}
61 \DeclareDocumentCommand{\raggedfootnote}{}{}
62
63 \DeclareDocumentCommand{\dictum}{o m}{
64   \begin{LWR@BlockClassWP}{\LWR@origmbox{text-align:right}}{}{\dictum}
65     #2
66     \IfValueT{#1}
67     {
68       \ifbool{FormatWP}
69       {\begin{BlockClass}[\LWR@origmbox{border-top:} 1px solid gray]{dictumauthor}}
70       {\begin{BlockClass}{dictumauthor}}
71       \dictumauthorformat{#1}
72       \end{BlockClass}
73     }
74   \end{LWR@BlockClassWP}
75 }
76
77 \DeclareDocumentCommand{\dictumwidth}{}{}
78 \DeclareDocumentCommand{\dictumauthorformat}{m}{(#1)}
79 \DeclareDocumentCommand{\dictumrule}{}{}
80 \DeclareDocumentCommand{\raggeddictum}{}{}
81 \DeclareDocumentCommand{\raggeddictumtext}{}{}
82 \DeclareDocumentCommand{\raggeddictumauthor}{}{}
83
84 \DeclareDocumentEnvironment{labeling}{o m}

```

```

85 {%
86 \def\sc@septest{#1}%
87 \list{}{}%
88 \let\makelabel\labelinglabel%
89 }
90 {
91 \endlist
92 }
93
94 \DeclareDocumentCommand{\labelinglabel}{m}{%
95 #1 \qqquad \sc@septest%
96 }
97
98 \let\addmargin\relax
99 \let\endaddmargin\relax
100 \cslet{addmargin*}{\relax}
101 \cslet{endaddmargin*}{\relax}
102
103 \NewDocumentEnvironment{addmargin}{s O{} m}
104 {
105 \setlength{\LWR@templengthtwo}{#3}
106 \ifblank{#2}
107 {
108     \begin{BlockClass}[
109         \LWR@origmbox{margin-left:\LWR@printlength{\LWR@templengthtwo}} ;
110         \LWR@origmbox{margin-right:\LWR@printlength{\LWR@templengthtwo}}
111     ]{addmargin}
112 }
113 {
114     \setlength{\LWR@templengthone}{#2}
115     \begin{BlockClass}[
116         \LWR@origmbox{margin-left:\LWR@printlength{\LWR@templengthone}} ;
117         \LWR@origmbox{margin-right:\LWR@printlength{\LWR@templengthtwo}}
118     ]{addmargin}
119 }
120 }
121 {\end{BlockClass}}

```

Ref to create a starred environment:

<https://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment>

```

122
123 \ExplSyntaxOn
124 \cs_new:cpn {addmargin*} {\addmargin*}
125 \cs_new_eq:cN {endaddmargin*} \endaddmargin
126 \ExplSyntaxOff
127
128 \DeclareDocumentCommand{\marginline}{m}{\marginpar{#1}}

```

---

File 187 **lwarp-scrhack.sty**

§ 274    Package **scrhack**

Pkg    scrhack    scrhack is ignored.

for HTML output:    1 \LWR@ProvidesPackageDrop{scrhack}

---

File 188 **lwarp-scrlayer.sty**

§ 275    Package **scrlayer**

*(Emulates or patches code by MARKUS KOHM.)*

Pkg    scrlayer    scrlayer is emulated.

 **Not fully tested!** [Please send bug reports!](#)

for HTML output:    1 \LWR@ProvidesPackageDrop{scrlayer}

```

2 \newcommand*{\DeclareSectionNumberDepth}[2]{}
3 \newcommand*{\DeclareLayer}[2] [] {}
4 \newcommand*{\DeclareNewLayer}[2] [] {}
5 \newcommand*{\ProvideLayer}[2] [] {}
6 \newcommand*{\RedeclareLayer}[2] [] {}
7 \newcommand*{\ModifyLayer}[2] [] {}
8 \newcommand*{\layerhalign}{}
9 \newcommand*{\layervalign}{}
10 \newcommand*{\layerxoffset}{}
11 \newcommand*{\layeryoffset}{}
12 \newcommand*{\layerwidth}{}
13 \newcommand*{\layerheight}{}
14 \providecommand*{\LenToUnit}[1]{\strip@pt\dimexpr#1*\p@/\unitlength}
15 \newcommand*{\putUL}[1]{}
16 \newcommand*{\putUR}[1]{}
17 \newcommand*{\putLL}[1]{}
18 \newcommand*{\putLR}[1]{}
19 \newcommand*{\putC}[1]{}
20 \newcommand*{\GetLayerContents}[1]{}
21 \newcommand*{\IfLayerExists}[3]{#3}
22 \newcommand*{\DestroyLayer}[1]{}
23 \newcommand*{\layercontentsmeasure}{}

```

```

24 \newcommand*{\currentpagestyle}{}
25 \newcommand*{\BeforeSelectAnyPageStyle}[1]{}
26 \newcommand*{\AfterSelectAnyPageStyle}[1]{}
27 \newcommand*{\DeclarePageStyleAlias}[2]{}
28 \newcommand*{\DeclareNewPageStyleAlias}[2]{}
29 \newcommand*{\ProvidePageStyleAlias}[2]{}
30 \newcommand*{\RedeclarePageStyleAlias}[2]{}
31 \newcommand*{\DestroyPageStyleAlias}[1]{}
32 \newcommand*{\GetRealPageStyle}[1]{}
33 \newcommand*{\DeclarePageStyleByLayers}[3]{}
34 \newcommand*{\DeclareNewPageStyleByLayers}[3]{}
35 \newcommand*{\ProvidePageStyleByLayers}[3]{}
36 \newcommand*{\RedeclarePageStyleByLayers}[3]{}
37 \NewDocumentCommand{\ForEachLayerOfPageStyle}{s m m}{}
38 \newcommand*{\AddLayersToPageStyle}[2]{}
39 \newcommand*{\AddLayersAtBeginOfPageStyle}[2]{}
40 \newcommand*{\AddLayersAtEndOfPageStyle}[2]{}
41 \newcommand*{\RemoveLayersFromPageStyle}[2]{}
42 \newcommand*{\AddLayersToPageStyleBeforeLayer}[3]{}
43 \newcommand*{\AddLayersToPageStyleAfterLayer}[3]{}
44 \newcommand*{\UnifyLayersAtPageStyle}[1]{}
45 \newcommand*{\ModifyLayerPageStyleOptions}[2]{}
46 \newcommand*{\AddToLayerPageStyleOptions}[2]{}
47 \newcommand*{\IfLayerPageStyleExists}[3]{#3}
48 \newcommand*{\IfRealLayerPageStyleExists}[3]{#3}
49 \newcommand*{\IfLayerAtPageStyle}[4]{#4}
50 \newcommand*{\IfSomeLayerAtPageStyle}[4]{#4}
51 \newcommand*{\IfLayersAtPageStyle}[4]{#4}
52 \newcommand*{\DestroyRealLayerPageStyle}[1]{}
53 \@ifundefined{footheight}{\newlength{footheight}}{}
54 \DeclareDocumentCommand{\automark}{s o m}{}
55 \DeclareDocumentCommand{\manualmark}{}{}
56 \DeclareDocumentCommand{\MakeMarkcase}{m}{#1}
57 \DeclareDocumentCommand{\GenericMarkFormat}{}{}
58 \newcommand*{\@mkleft}[1]{}
59 \newcommand*{\@mkright}[1]{}
60 \newcommand*{\@mkdouble}[1]{}
61 \newcommand*{\@mkboth}[2]{}
62 \newcommand*{\scrlayerInitInterface}[1]{}
63 \newcommand*{\scrlayerAddToInterface}[3]{}
64 \newcommand*{\scrlayerAddCsToInterface}[3]{}
65 \newcommand*{\scrlayerOnAutoRemoveInterface}[2]{}

```

---

File 189 **lwarp-scrlayer-notecolumn.sty**

§ 276      Package **scrlayer-notecolumn**

*(Emulates or patches code by MARKUS KOHM.)*

Pkg    scrlayer-notecolumn    scrlayer-notecolumn is emulated.

 **Not fully tested!** [Please send bug reports!](#)

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{scrlayer-notecolumn}

2 \newcommand*{\DeclareNoteColumn}[2] [] {}
3 \newcommand*{\DeclareNewNoteColumn}[2] [] {}
4 \newcommand*{\ProvideNoteColumn}[2] [] {}
5 \newcommand*{\RedeclareNoteColumn}[2] [] {}
6 \NewDocumentCommand{\makenote}{s o m}{\marginpar{#3}}
7 \newcommand*{\syncwithnotecolumn}[1] [] {}
8 \newcommand*{\syncwithnotecolumns}[1] [] {}
9 \newcommand*{\clearnotecolumn}[1] [] {}
10 \newcommand*{\clearnotecolumns}[1] [] {}

```

---

File 190 **lwarp-scrlayer-scrpage.sty**

§ 277      Package **scrlayer-scrpage**

*(Emulates or patches code by MARKUS KOHM.)*

Pkg    scrlayer-scrpage    scrlayer-scrpage is emulated.

 **Not fully tested!** [Please send bug reports!](#)

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{scrlayer-scrpage}

2 \@ifundefined{footheight}{\newlength{footheight}}{}
3 \NewDocumentCommand{\lehead}{s o m}{}
4 \NewDocumentCommand{\cehead}{s o m}{}
5 \NewDocumentCommand{\rehead}{s o m}{}
6 \NewDocumentCommand{\lohead}{s o m}{}
7 \NewDocumentCommand{\cohead}{s o m}{}
8 \NewDocumentCommand{\rohead}{s o m}{}
9 \NewDocumentCommand{\lefoot}{s o m}{}
10 \NewDocumentCommand{\cefoot}{s o m}{}

```

---

```

11 \NewDocumentCommand{\refoot}{s o m}{}
12 \NewDocumentCommand{\lofoot}{s o m}{}
13 \NewDocumentCommand{\cofoot}{s o m}{}
14 \NewDocumentCommand{\rofoot}{s o m}{}
15 \NewDocumentCommand{\ohead}{s o m}{}
16 \NewDocumentCommand{\chead}{s o m}{}
17 \NewDocumentCommand{\ihead}{s o m}{}
18 \NewDocumentCommand{\ofoot}{s o m}{}
19 \NewDocumentCommand{\cfoot}{s o m}{}
20 \NewDocumentCommand{\ifoot}{s o m}{}
21 \DeclareDocumentCommand{\automark}{s o m}{}
22 \DeclareDocumentCommand{\manualmark}{}{}
23 \DeclareDocumentCommand{\MakeMarkcase}{m}{#1}
24 \DeclareDocumentCommand{\GenericMarkFormat}{}{}
25 \newcommand*{\defpairofpagestyles}[3] [] {}
26 \newcommand*{\newpairofpagestyles}[3] [] {}
27 \newcommand*{\renewpairofpagestyles}[3] [] {}
28 \newcommand*{\providepairofpagestyles}[3] [] {}
29 \newcommand*{\clearmainofpairofpagestyles}
30 \newcommand*{\clearplainofpairofpagestyles}
31 \newcommand*{\clearpairofpagestyles}
32 \NewDocumentCommand{\deftriplepagestyle}{m o o m m m m m m}{}
33 \NewDocumentCommand{\newtriplepagestyle}{m o o m m m m m m}{}
34 \NewDocumentCommand{\renewtriplepagestyle}{m o o m m m m m m}{}
35 \NewDocumentCommand{\providetriplespagestyle}{m o o m m m m m m}{}
36 \newcommand*{\defpagestyle}[3] {}
37 \newcommand*{\newpagestyle}[3] {}
38 \newcommand*{\providepagestyle}[3] {}
39 \newcommand*{\renewpagestyle}[3] {}

```

---

File 191 **lwarp-section.sty**

§ 278 Package **section**

Pkg section section is ignored.

*(Emulates or patches code by OLIVER PRETZEL.)*

**for HTML output:** 1 \LWR@ProvidesPackageDrop{section}

```

2 \ifx\chapter\undefined
3 \def\chsize{\Large}\def\hdsi{\huge}\else
4 \def\chsize{\huge}\def\hdsi{\Huge}
5 \fi
6 \let\ttsi\LARGE
7 \let\ausi\large
8 \let\dasizelarge

```

---

```

9 \let\seclsize\Large
10 \let\subsize\large
11 \let\hdpos\raggedright
12 \newcounter{hddepth}
13 \let\fpind\relax
14 \def\ttfnt{}
15 \def\hdfnt{}
16 \def\fefnt{}
17 \def\thfnt{}
18 \def\pgfnt{}
19 \def\hmkfnt{}
20 \let\mkcse\uppercase
21 \def\hddot{}
22 \def\cpdot{:}
23 \def\nmidot{}
24 \ifx\secindent\undefined
25 \newdimen\secindent
26 \newskip\secpreskp
27 \newskip\secpstskp
28 \newdimen\subindent
29 \newskip\subpreskp
30 \newskip\subpstskp
31 \newskip\parpstskp
32 \newcount\c@hddepth
33 \fi

```

---

File 192 **lwarp-sectionbreak.sty**

§ 279 Package **sectionbreak**

*(Emulates or patches code by MICHAL HOFTICH.)*

Pkg sectionbreak sectionbreak is patched for use by lwarp.

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{sectionbreak}

2 \renewcommand\asterism{\HTMLUnicode{2042}}
3
4 \renewcommand\pre@sectionbreak{}
5 \renewcommand\post@sectionbreak{}
6
7 \renewcommand\print@sectionbreak[1]{%
8 \begin{center}
9 #1
10 \end{center}
11 }
12

```



---

File 193 **lwarp-sectsty.sty**

§ 280      Package **sectsty**

*(Emulates or patches code by ROWLAND McDONNELL.)*

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 194 **lwarp-setspace.sty**

§ 281      Package **setspace**

*(Emulates or patches code by ROBIN FAIRBAIRNS.)*

Pkg    setspace    setspace is not used during HTML conversion.

Discard all options for lwarp-setspace:

for HTML output:

---

```

1 \LWR@ProvidesPackageDrop{setSPACE}
2
3 \newcommand*{\setstretch}[1]{}
4 \newcommand*{\SetSingleSpace}[1]{}
5 \newcommand*{\singleSpacing}{}
6 \newcommand*{\onehalfSpacing}{}
7 \newcommand*{\doubleSpacing}{}
8
9 \newenvironment*{singleSpace}
10 {
11 \LWR@forcenewpage
12 \BlockClass{singleSpace}
13 }
14 {\endBlockClass}
15
16 \newenvironment*{singleSpace*}
17 {
18 \LWR@forcenewpage
19 \BlockClass{singleSpace}
20 }
21 {\endBlockClass}
22
23 \newenvironment*{spacing}[1]{
24
25 }{
26
27 }
28
29 \newenvironment*{onehalfSpace}
30 {
31 \LWR@forcenewpage
32 \BlockClass{onehalfSpace}
33 }
34 {\endBlockClass}
35
36 \newenvironment*{doubleSpace}
37 {
38 \LWR@forcenewpage
39 \BlockClass{doubleSpace}
40 }
41 {\endBlockClass}

```

---

File 195 **lwarp-shadow.sty**

§ 282 Package **shadow**

*(Emulates or patches code by MAURO ORLANDINI.)*

Pkg shadow shadow is emulated.

for HTML output: Discard all options for lwarp-shadow:

```
1 \LWR@ProvidesPackageDrop{shadow}

2 \newdimen\sboxsep
3 \newdimen\sboxrule
4 \newdimen\sdim
5
6 \newcommand{\shabox}[1]{%
7 \InlineClass{shabox}{#1}%
8 }
```

---

File 196 **lwarp-showidx.sty**

§ 283 Package **showidx**

Pkg showidx showidx is ignored.

for HTML output: Discard all options for lwarp-showidx:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

---

File 197 **lwarp-showkeys.sty**

§ 284 Package **showkeys**

*(Emulates or patches code by DAVID CARLISLE, MORTEN HØGHOLM.)*

Pkg showkeys showkeys is ignored.

for HTML output: Discard all options for lwarp-showkeys:

```
1 \LWR@ProvidesPackageDrop{showkeys}

2 \NewDocumentCommand{\showkeys}{s}{}
```

---

File 198 **lwarp-sidecap.sty**

§ 285 Package **sidecap**

*(Emulates or patches code by ROLF NIEPRASCHK, HUBERT GÄSSLEIN.)*

Pkg sidecap sidecap is emulated.

for HTML output: Discard all options for lwarp-sidecap.

```
1 \LWR@ProvidesPackageDrop{sidecap}
```

See:

<http://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment>  
regarding the creation of starred environments with xparse.

```
2 \NewDocumentEnvironment{SCtable}{soo}
3 {\IfValueTF{#3}{\table[#3]}\table}}
4 {\endtable}
5
6 \ExplSyntaxOn
7 \cs_new:cpn {SCtable*} {\SCtable*}
8 \cs_new_eq:cN {endSCtable*} \endSCtable
9 \ExplSyntaxOff
10
11
12 \NewDocumentEnvironment{SCfigure}{soo}
13 {\IfValueTF{#3}{\figure[#3]}\figure}}
14 {\endfigure}
15
16 \ExplSyntaxOn
17 \cs_new:cpn {SCfigure*} {\SCfigure*}
18 \cs_new_eq:cN {endSCfigure*} \endSCfigure
19 \ExplSyntaxOff
20
21
22 \newenvironment*{wide}{}{}
```

---

File 199 **lwarp-sidenotes.sty**

§ 286 Package **sidenotes**

*(Emulates or patches code by ANDY THOMAS, OLIVER SCHEBAUM.)*

Pkg sidenotes Patched for lwarp.

for HTML output: Load the original package:

```
1 \LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with lwarp:

```

\sidecaption * [<entry>] [<offset>] {<text>}

2 \RenewDocumentCommand \sidecaption {s o o +m}
3 {
4   \LWR@stoppars
5   \begingroup
6   \captionsetup{style=sidecaption}
7   \IfBooleanTF{#1}
8   { % starred
9     \begin{BlockClass}[border:none ; box-shadow:none]{marginblock}
10    \caption*{#4}
11    \end{BlockClass}
12  }
13  { % unstarred
14    \IfNoValueOrEmptyTF{#2}
15    {\def\@sidenotes@sidecaption@tof{#4}}
16    {\def\@sidenotes@sidecaption@tof{#2}}
17    \begin{BlockClass}[border:none ; box-shadow:none]{marginblock}
18    \caption[\@sidenotes@sidecaption@tof]{#4}
19    \end{BlockClass}
20  }
21  \endgroup
22  \LWR@startpars
23 }

```

Borrowed from the lwarp version of keyfloat:

```

24 \NewDocumentEnvironment{KFLT sidenotes@marginfloat}{0{-1.2ex} m}
25 {% start
26 \LWR@BlockClassWP{float:right; width:2in; margin:10pt}{\marginblock}%
27 \captionsetup{type=#2}%
28 }
29 {%
30 \endLWR@BlockClassWP%
31 }
32
33 \RenewDocumentEnvironment{marginfigure}{o}
34 {\begin{KFLT sidenotes@marginfloat}{figure}}
35 {\end{KFLT sidenotes@marginfloat}}
36
37 \RenewDocumentEnvironment{margintable}{o}
38 {\begin{KFLT sidenotes@marginfloat}{table}}
39 {\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  $\LaTeX$  2<sub>ε</sub> article.cls source:

```

40 \renewenvironment{figure*}

```

---

```

41          {\@dblfloat{figure}}
42          {\end@dblfloat}
43
44 \renewenvironment{table*}
45          {\@dblfloat{table}}
46          {\end@dblfloat}

```

---

File 200 **lwarp-siunitx.sty**

§ 287    Package **siunitx**

*(Emulates or patches code by JOSEPH WRIGHT.)*

Pkg    siunitx    siunitx is patched for use by lwarp.

**fractions**    Due to pdftolatex limitations, fraction output is replaced by symbol output for per-mode and quotient-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.

**for HTML output:**

```

1 \RequirePackage{xcolor}% for \convertcolorspec
2
3 \LWR@ProvidesPackagePass{siunitx}

```

```

4 \AtBeginDocument{% in case textcomp was not loaded
5 \DeclareSIUnit\bohr{\textit{a}\textsubscript{0}}
6 \DeclareSIUnit\clight{\textit{c}\textsubscript{0}}
7 \DeclareSIUnit\elementarycharge{\textit{e}}
8 \DeclareSIUnit\electronmass{\textit{m}\textsubscript{e}}
9 \DeclareSIUnit\hartree{\textit{E}\textsubscript{h}}
10 \DeclareSIUnit\planckbar{\LWR@siunitx@textplanckbar}
11 }% AtBeginDocument

```

\@ensuredmath is not supported inside an \hbox, so it must temporarily be restored to its original. Similar for \mbox. SVG math is created explicitly when necessary, using \LWR@subsingledollar.

```

12
13 \ExplSyntaxOn
14 %

```

Modified to set set HTML \textcolor if not black:

```

15 \cs_undefine:N \__siunitx_print_aux:
16 \cs_new_protected:Npn \__siunitx_print_aux:
17 {
18   \text
19   {
20     \__siunitx_ensure_ltr:n
21     {
22       \color@begingroup
23       \__siunitx_print_color:
24       \__siunitx_font_shape:
25       \__siunitx_font_weight:
26       \use:c
27       {
28         @@_ \l__siunitx_print_type_tl _
29         text \l__siunitx_font_family_tl :
30       }
31       \bool_if:NTF \l__siunitx_font_math_mode_bool
32       { \__siunitx_print_math: }
33       {
34         \LWR@findcurrenttextcolor% lwarp
35         \ifdefstring{\LWR@tempcolor}{000000}% lwarp
36         {\__siunitx_print_text:}% lwarp
37         {% lwarp
38           \LWR@textcurrentcolor{% lwarp
39             \__siunitx_print_text:
40            }% lwarp
41          }% lwarp
42         }
43       \color@endgroup
44     }
45   }
46 }
47
48
49 \cs_undefine:N \__siunitx_set_math_fam:n
50 \cs_new_protected:Npn \__siunitx_set_math_fam:n #1 {
51   \int_new:c { c__siunitx_math #1 _int }
52   \group_begin:% lwarp
53   \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
54   \LetLtxMacro\mbox\LWR@origmbox% lwarp
55   \hbox_set:Nn \l__siunitx_tmp_box
56   {
57     \ensuremath
58     {
59       \use:c { math #1 }
60       {
61         \int_gset:cn { c__siunitx_math #1 _int } { \fam }
62       }

```

```

63      }
64    }
65    \group_end:% lwarp
66 }
67
68 \cs_undefine:N \__siunitx_combined_output:n
69 \cs_new_protected:Npn \__siunitx_combined_output:n #1 {
70   \group_begin:% lwarp
71   \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
72   \LetLtxMacro\mbox\LWR@origmbox% lwarp
73   \bool_if:NTF \l__siunitx_number_parse_bool
74   {
75     \tl_clear:N \l__siunitx_number_out_tl
76     \bool_set_false:N \l__siunitx_number_compound_bool
77     \__siunitx_number_output_parse:n {#1}
78   }
79   {
80     \__siunitx_unit_output_pre_print:

```

For parse-numbers=false:

```

81%     \__siunitx_print:nn { number } { \ensuremath {#1} }
82     \LWR@subsingledollar{% lwarp
83       \textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)% lwarp
84     }{\__siunitx_print:nn { number } { #1 }}% lwarp
85
86     \__siunitx_unit_output_print:
87   }
88 \group_end:% lwarp
89 %

```

For quotients, the fraction code is replaced by the symbol code:

```

90 \cs_undefine:N \__siunitx_number_output_quotient_fraction:
91 \cs_new_protected:Npn \__siunitx_number_output_quotient_fraction: {
92   \bool_set_true:N \l__siunitx_number_compound_bool
93   \__siunitx_number_output_quotient_aux_i:
94   \tl_set_eq:NN \l__siunitx_number_out_tl
95     \l__siunitx_number_numerator_tl
96   \tl_put_right:NV \l__siunitx_number_out_tl \l__siunitx_output_quotient_tl
97   \tl_put_right:NV \l__siunitx_number_out_tl
98     \l__siunitx_number_denominator_tl
99   \__siunitx_number_output_single_aux:
100 }

```

For units, the fraction code is replaced by the symbol code:

```

101 \cs_undefine:N \__siunitx_unit_format_fraction_fraction:

```



```

102 \cs_new_protected:Npn \__siunitx_unit_format_fraction_fraction: {
103   \__siunitx_unit_format_fraction_symbol_aux:
104   \int_compare:nNnT { \l__siunitx_unit_denominator_int } > { 1 }
105   {
106     \bool_if:NT \l__siunitx_unit_denominator_bracket_bool
107     {
108       \tl_put_left:NV \l__siunitx_unit_denominator_tl \l__siunitx_bracket_open_tl
109       \tl_put_right:NV \l__siunitx_unit_denominator_tl \l__siunitx_bracket_close_tl
110     }
111   }
112   \tl_set_eq:NN \l__siunitx_unit_tl \l__siunitx_unit_numerator_tl
113   \tl_put_right:NV \l__siunitx_unit_tl \l__siunitx_per_symbol_tl
114   \tl_put_right:NV \l__siunitx_unit_tl \l__siunitx_unit_denominator_tl
115 }

116 \cs_undefine:N \__siunitx_angle_print_astronomy_aux:
117 \cs_new_protected:Npn \__siunitx_angle_print_astronomy_aux: {
118   \prop_get:NnNT \l__siunitx_number_out_prop { mantissa-integer }
119   \l__siunitx_tmpa_tl
120   { \__siunitx_print:nV { number } \l__siunitx_tmpa_tl }
121   \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}% lwarp
122   {% lateximage
123     \hbox_set:Nn \l__siunitx_angle_marker_box
124     {
125       \__siunitx_print:nn { number } { { \l__siunitx_output_decimal_tl } }
126     }
127     \hbox_set:Nn \l__siunitx_angle_unit_box
128     {
129       \__siunitx_print:nV { unit } \l__siunitx_unit_tl
130       \skip_horizontal:n { -\scriptspace }
131     }
132     \__siunitx_angle_print_astronomy_aux:n { marker }
133     \__siunitx_angle_print_astronomy_aux:n { unit }
134     \hbox_set:Nn \l__siunitx_angle_marker_box
135     {
136       \box_use:N \l__siunitx_angle_marker_box
137       \box_use:N \l__siunitx_angle_unit_box
138     }
139     \dim_compare:nNnTF
140     { \l__siunitx_angle_marker_dim } > { \l__siunitx_angle_unit_dim }
141     { \__siunitx_angle_print_astronomy_marker: }
142     { \__siunitx_angle_print_astronomy_unit: }
143   }% lateximage
144   {% not a lateximage
145     \__siunitx_print:nV { unit } \l__siunitx_unit_tl
146     \__siunitx_print:nn { number } { { \l__siunitx_output_decimal_tl } }
147   }% not a lateximage
148   \prop_get:NnNT \l__siunitx_number_out_prop { mantissa-decimal }
149   \l__siunitx_tmpa_tl

```

```

150   { \__siunitx_print:nV { number } \l__siunitx_tmpa_tl }
151 }

152 \RenewDocumentCommand \num { o m } {
153   \leavevmode
154   \group_begin:% lwarp
155   \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
156   \LetLtxMacro\mbox\LWR@origmbox% lwarp
157   \bool_set_false:N \l__siunitx_font_set_bool
158   \IfNoValueF {#1}
159   { \keys_set:nn { siunitx } {#1} }
160   \__siunitx_number_output:n {#2}
161   \group_end:% lwarp
162 }
163
164 \RenewDocumentCommand \numrange { o m m } {
165   \leavevmode
166   \group_begin:% lwarp
167   \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
168   \LetLtxMacro\mbox\LWR@origmbox% lwarp
169   \bool_set_false:N \l__siunitx_font_set_bool
170   \IfNoValueF {#1}
171   { \keys_set:nn { siunitx } {#1} }
172   \__siunitx_range_numbers:nn {#2} {#3}
173   \group_end:% lwarp
174 }
175
176 \RenewDocumentCommand \ang { o > { \SplitArgument { 2 } { ; } } m } {
177   \group_begin:% lwarp
178   \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
179   \LetLtxMacro\mbox\LWR@origmbox% lwarp
180   \IfNoValueF {#1}
181   { \keys_set:nn { siunitx } {#1} }
182   \__siunitx_angle_output:nnn #2
183   \group_end:% lwarp
184 }
185
186 \RenewDocumentCommand \si { o m } {
187   \leavevmode
188   \group_begin:% lwarp
189   \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
190   \LetLtxMacro\mbox\LWR@origmbox% lwarp
191   \bool_set_false:N \l__siunitx_font_set_bool
192   \IfNoValueTF {#1}
193   { \__siunitx_unit_output:nn {#2} { } }
194   {
195     \keys_set:nn { siunitx } {#1}
196     \__siunitx_unit_output:nn {#2} {#1}
197   }

```

---

```

198 \group_end:% lwarp
199 }
200
201
202 \RenewDocumentCommand{\SIrange}{o m m m}
203 {%
204 \leavevmode
205 \group_begin:% lwarp
206 \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
207 \LetLtxMacro\mbox\LWR@origmbox% lwarp
208 \bool_set_false:N \l__siunitx_font_set_bool
209 \IfNoValueTF {#1}
210 { \__siunitx_range_unit:nnnn {#4} { } {#2} {#3} }
211 {
212 \keys_set:nn { siunitx } {#1}
213 \__siunitx_range_unit:nnnn {#4} {#1} {#2} {#3}
214 }
215 \group_end:% lwarp
216 }
217
218 \ExplSyntaxOff

```

---

File 201 **lwarp-soul.sty**

§ 288 Package **soul**

*(Emulates or patches code by MELCHIOR FRANZ.)*

Pkg soul Emulated.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{soul}[2003/11/17]
2 \RequirePackage{xcolor}% for \convertcolorspec

```

Storage for the colors to use:

```

3 \newcommand*{\LWR@soululcolor}{}
4
5 \newcommand*{\LWR@soulstcolor}{}
6
7 % \definecolor{\LWR@soulhlcolordefault}{HTML}{F8E800}
8 % \newcommand*{\LWR@soulhlcolor}{\LWR@soulhlcolordefault}
9 \newcommand*{\LWR@soulhlcolor}{}

```

\so {<text>}

Basic markup with css:

```

10 \newcommand{\so}[1]{%
11 \LWR@HTMLtextstyle{letter-spacing:.2ex}{letterspacing}{#1}%
12 }

\caps  {\langle text\rangle}

13 \newcommand{\caps}[1]{%
14 \LWR@HTMLtextstyle%
15   {font-variant:small-caps;letter-spacing:.1ex}%
16   {capsspacing}{#1}%
17 }

\LWR@soulcolor  {\langle text\rangle} {\langle color\rangle} {\langle class\rangle} {\langle colorstyle\rangle} {\langle FormatWPstyle\rangle}
Add colors if not empty:

18 \newcommand{\LWR@soulcolor}[5]{%
19 \ifcsempy{#2}%
20 {\LWR@HTMLtextstyle{#5}{#3}{#1}}%
21 {%
22   \convertcolorspec{named}{\csuse{#2}}{HTML}\LWR@tempcolor%
23   \LWR@htmlspanclass[#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[1]{}
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 202 **lwarp-soulpos.sty**

§ 289      Package **soulpos**

*(Emulates or patches code by JAVIER BEZOS.)*

Pkg    soulpos    soulpos is emulated.

**for HTML output:**

```

1 \RequirePackage{soul}
2 \RequirePackage{soulutf8}
3 \LWR@ProvidesPackageDrop{soulpos}

4 \NewDocumentCommand{\ulposdef}{m o m}{}
5
6 \newdimen\ulwidth
7
8 \newcommand\ifulstarttype[1]{%
9 \expandafter\@secondoftwo%
10 }
11
12 \newcommand\ifulendtype[1]{%
13 \expandafter\@secondoftwo%
14 }
15
16 \newcommand{\ulstarttype}{0}
17 \newcommand{\ulendtype}{0}
18 \newcommand\ulpostolerance{0}%

```

---

File 203 **lwarp-soulutf8.sty**

§ 290     Package **soulutf8**

Pkg soulutf8 soulutf8 is emulated.

lwarp's HTML output naturally supports UTF-8 encoding.

for HTML output:    1 \LWR@ProvidesPackageDrop{soulutf8}

---

File 204 **lwarp-stabular.sty**

§ 291     Package **stabular**

*(Emulates or patches code by SIGITAS TOLUŠIS.)*

Pkg stabular stabular is emulated.

for HTML output:    1 \LWR@ProvidesPackageDrop{stabular}

Env stabular    [ $\langle vpos \rangle$ ] [ $\langle colspec \rangle$ ]

```
2 \newenvironment{stabular}[2][c]
3 {
4 \begin{tabular}[#1]{#2}
5 \renewcommand{\noalign}[1]{\}
6 }
7 {\end{tabular}}
```

Env stabular    [ $\langle width \rangle$ ] [ $\langle vpos \rangle$ ] [ $\langle colspec \rangle$ ]

```
8 \NewDocumentEnvironment{stabular*}{m o m}
9 {
10 \begin{tabular}[#2]{#3}
11 \renewcommand{\noalign}[1]{\}
12 }
13 {\end{tabular}}
```

---

File 205 **lwarp-stfloats.sty**

§ 292     Package **stfloats**

Pkg    stfloats    stfloats is ignored.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{stfloats}

2 \newcommand\*{\fnbelowfloat}{}  
 3 \newcommand\*{\fnunderfloat}{}  
 4 \newcommand\*{\setbaselinefloat}{}  
 5 \newcommand\*{\setbaselinefixed}{}  
 6 \newcommand\*{\setbaselinefixed}


---

File 206 **lwarp-subfig.sty**

§ 293     Package **subfig**

*(Emulates or patches code by STEVEN DOUGLAS COCHRAN.)*

Pkg    subfig    subfig is supported and patched by lwarp.

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

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

**for HTML output:**    Accept all options for lwarp-subfig:

1 \LWR@ProvidesPackagePass{subfig}

\sf@@@subfloat    {\langle 1 type \rangle} [\langle 2 lof entry \rangle] [\langle 3 caption \rangle] {\langle 4 contents \rangle}

The outer minipage allows side-by-side subfloats with \hfill between.

2 \long\def\sf@@@subfloat#1[#2][#3]#4{%  
 3 \begin{minipage}{\linewidth}% lwarp  
 4 \IfValueTF{#2}{%  
 5     \LWR@setlatestname{#2}%  
 6 }{%

```

7   \IfValueTF{#3}{%
8       \LWR@setlatestname{#3}%
9   }{}%
10}%
11 \LWR@stoppars% lwarp
12   \@ifundefined{FBsc@max}{}%
13       {\FB@readaux{\let\FBsuboheight\relax}}%
14   \@tempcnta=\@ne
15   \if@minipage
16       \@tempcnta=\z@
17   \else\ifdim \lastskip=\z@ \else
18       \@tempcnta=\tw@
19   \fi\fi
20   \ifmaincaptiontop
21       \sf@top=\sf@narskip
22       \sf@bottom=\sf@farskip
23   \else
24       \sf@top=\sf@farskip
25       \sf@bottom=\sf@narskip
26   \fi
27   \leavevmode

28 %   \setbox\@tempboxa \hbox{#4}%
29 %   \@tempdima=\wd\@tempboxa
30 %   \@ifundefined{FBsc@max}{}%
31 %       {\global\advance\Xhsize-\wd\@tempboxa
32 %           \dimen@=\ht\@tempboxa
33 %           \advance\dimen@\dp\@tempboxa
34 %           \ifdim\dimen@>\FBso@max
35 %               \global\FBso@max\dimen@
36 %           \fi}%

```

Do not use boxes, which interfere with lateximages:

```

37 %   \vtop%
38   \bgroup
39 %   \vbox%
40   \bgroup
41       \ifcase\@tempcnta
42           \@minipagefalse
43       \or
44 %           \vskip\sf@top
45       \or
46           \ifdim \lastskip=\z@ \else
47 %               \@tempskipb\sf@top\relax\@xaddvskip
48           \fi
49       \fi
50       \sf@ifpositiontop{%
51           \ifx \@empty#3\relax \else
52               \sf@subcaption{#1}{#2}{#3}%

```



```

53 %          \vskip\sf@capskip
54 %          \vskip\sf@captopadj
55          \fi\egroup
56 %          \hrule widthOpt heightOpt depthOpt
57          \LWR@startpars% lwarp
58 % \box\@tempboxa
59          #4
60          \LWR@stoppars% lwarp
61      }{%
62          \LWR@startpars% lwarp
63          \@ifundefined{FBsc@max}%
64          {
65 % \box\@tempboxa
66          #4
67          }%
68          {\ifx\FBsuboheight\relax
69 %          \box\@tempboxa
70          #4
71          \else
72 %          \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
73          #4
74          \fi}%
75      \LWR@stoppars% lwarp
76      \egroup
77      \ifx \@empty#3\relax \else
78 %          \vskip\sf@capskip
79 %          \hrule widthOpt heightOpt depthOpt
80 %          \sf@subcaption{#1}{#2}{#3}%
81      \fi
82      }%
83 %      \vskip\sf@bottom
84      \egroup
85      \@ifundefined{FBsc@max}{}%
86      {\addtocounter{FRobj}{-1}%
87      \ifnum\c@FRobj=0\else
88      \subfloatrowsep
89      \fi}%
90      \ifmaincaptiontop\else
91      \global\advance\@nameuse{c@\@captype}\m@ne
92      \fi
93 \end{minipage}% lwarp
94 \LWR@startpars% lwarp
95 \endgroup\ignorespaces%
96 }%

```

`\sf@subcaption`     $\{\langle 1 \text{ type} \rangle\} \{\langle 2 \text{ lof entry} \rangle\} \{\langle 3 \text{ caption} \rangle\}$

```

97 \long\def\sf@subcaption#1#2#3{%
98 \LWR@stoppars% lwarp

```

---

```

99 \ifx \relax#2\relax \else
100 \bgroup
101 \let\label=\@gobble
102 \let\protect=\string
103 \def\@subcaplabel{%
104 \caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
105 \sf@updatecaptionlist{#1}{#2}{\the\value{\@captype}}{\the\value{#1}}%
106 \egroup
107 \fi
108 \bgroup
109 \ifx \relax#3\relax
110 \let\captionlabelsep=\relax
111 \fi
112 % \setbox0\vbox{%
113 % \hb@xt@\the\@tempdima{%
114 %
115 % \hss
116 % \parbox[t]{\the\@tempdima}{%
117 % \caption@make
118 % {\@nameuse{sub\@captype name}}%
119 % {\@nameuse{thesub\@captype}}%
120 % {#3}
121 % }%
122 % \hss
123 % }
124 % }%
125 \@ifundefined{FBsc@max}%
126 % {\box0}%
127 % {
128 % \parbox[t]{\the\@tempdima}{%
129 \LWR@traceinfo{sfsubcap B1}% lwarp
130 \LWR@figcaption% lwarp
131 \caption@make
132 {\@nameuse{sub\@captype name}}%
133 {\@nameuse{thesub\@captype}}%
134 {#3}

135 \endLWR@figcaption% lwarp
136 \LWR@traceinfo{sfsubcap B2}% lwarp
137 % }%
138 % }%
139 {\dimen@ \ht0%
140 \advance\dimen@ \dp0%
141 \ifdim\dimen@ > \FBsc@max
142 \global\FBsc@max \dimen@
143 \fi
144 \FB@readaux{\let\FBsubcheight\relax}%
145 \ifx\FBsubcheight\relax
146 \def\next{
147 % \parbox[t]{\the\@tempdima}

```

```

148         }%
149     \else
150         \def\next{
151 %   \parbox[t][\FBsubheight][t]{\the\@tempdima}
152         }%
153     \fi
154 %       \vbox{%
155 %           \hb@xt@\the\@tempdima{%
156
157 %               \hss
158 %               \next{%
159 \LWR@traceinfo{sfsubcap C1}% lwarp
160             \caption@make
161                 {\@nameuse{sub\@capttype name}}}%
162                 {\@nameuse{thesub\@capttype}}}%
163                 {#3}
164 \LWR@traceinfo{sfsubcap C1}% lwarp
165 %   }%
166 %       \hss
167
168 %   }
169 %       }
170 %   }%
171 \egroup
172 \LWR@startpars% lwarp
173 }

```

`\subfloat@label` Patches for `\sf@sub@label`:

```

174 \def\subfloat@label{%
175 \LWR@ensuredoingapar% lwarp
176 \@ifnextchar(%   %) match left parenthesis
177   {\sf@sub@label}
178   {\sf@sub@label(Sub\@capttype\space
179                 \ifundefined{thechapter}{\@nameuse{thechapter}\space}%
180                 \@nameuse{p@sub\@capttype}%
181                 \@nameuse{thesub\@capttype}.)}}

```

Patches for `\subref`.

`\sf@subref` `{<label>}`

The unstarred version uses a `\ref` link whose printed text comes from the `sub@<label>`:

```

182 \renewcommand{\sf@subref}[1]{%
183 \LWR@subnewref{#1}{sub@#1}%
184 }

```

`\sf@@subref` `{<label>}`

The starred version uses the printed sub@<label> which is stored as if it were a page number:

```
185 \renewcommand{\sf@@subref}[1]{\LWR@origpageref{sub@#1}}
```

Defining new subfloats. The l@sub<type> for each is redefined.

```
\@newsubfloat    [{<keys/values>}] {<float name>}

186 \LetLtxMacro\LWR@orig@newsubfloat\@newsubfloat
187
188 \def\@newsubfloat[#1]#2{%
189 \LWR@orig@newsubfloat[#1]{#2}%
190 \renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}%
191 }
```

Pre-defined for figures and tables:

```
\l@subfigure    {<text>} {<pagenum>}

192 \renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2}}

\l@subtable     {<text>} {<pagenum>}

193 \renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}
```

---

File 207 **lwarp-subfigure.sty**

## \$ 294    Package **subfigure**

Pkg    subfigure    subfigure is emulated by subfig.

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{subfigure}
2 \RequirePackage{subfig}

3 \LetLtxMacro\subfigure\subfloat
4 \LetLtxMacro\subtable\subfloat
5 \LetLtxMacro\Subref\subref
6 \@ifundefined{figuretopcaptrue}{\newif\iffiguretopcap}{}
7 \newif\ifsubfiguretopcap
8 \newif\ifsubcaphang
9 \newif\ifsubcapcenter
10 \newif\ifsubcapcenterlast
11 \newif\ifsubcapnooneline
12 \newif\ifsubcapraggedright
13 \newskip\subfigtopskip
14 \newskip\subfigcapskip
```

---

```

15 \newdimen\subfigcaptopadj
16 \newskip\subfigbottomskip
17 \newdimen\subfigcapmargin
18 \newskip\subfiglabelskip
19 \newcommand*{\subcapsize}{}
20 \newcommand*{\subcaplabelfont}{}
21 \newcommand*{\subcapfont}{}

```

---

File 208 **lwarp-supertabular.sty**

§ 295     Package **supertabular**

*(Emulates or patches code by JOHANNES BRAAMS, THEO JURRIENS.)*

Pkg supertabular supertabular is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{supertabular}

⚠ misplaced alignment  
alignment tab character &

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

```

\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars

```

See section 8.9.

⚠ lateximage supertabular and xtab are not supported inside a lateximage.

```

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][1]{%
18   \long\gdef\LWRST@caption{\caption{#1}{#2}}%
19 }

```

---

```

20
21 \let\topcaption\tablecaption
22 \let\bottomcaption\tablecaption
23

24 \newcommand*{\LWRST@caption}{}
25
26 \newcommand*{\shrinkheight}[1]{}
27
28 \NewDocumentEnvironment{supertabular}{s o m}
29 {%
30 \LWR@traceinfo{supertabular}%
31 \table%
32 \LWRST@caption%
33 \begin{tabular}{#3}%
34 \TabularMacro\ifdefvoid{\LWRST@firsthead}%
35 {\LWR@getmynexttoken}%
36 {\expandafter\LWR@getmynexttoken\LWRST@firsthead}%
37 }%
38 {%
39 \ifdefvoid{\LWRST@lasttail}%
40 {}%
41 {%
42 \TabularMacro\ResumeTabular%
43 \LWRST@lasttail%
44 }%
45 \end{tabular}%
46 \endtable%
47 \LWR@traceinfo{supertabular done}%
48 }
49
50 \NewDocumentEnvironment{mpsupertabular}{s o m}
51 {\minipage{\linewidth}\supertabular{#3}}
52 {\endsupertabular\endminipage}

```

---

File 209 **lwarp-syntonly.sty**

§ 296      Package **syntonly**

*(Emulates or patches code by FRANK MITTELBACH, RAINER SCHÖPF.)*

Pkg    syntonly    Emulated.

**for HTML output:**    Discard all options for lwarp-syntonly:

```
1 \LWR@ProvidesPackageDrop{syntonly}
```

---

```

2 \newif\ifsyntax@
3 \syntax@false
4
5 \newcommand*{\syntaxonly}{\}
6
7 \@onlypreamble\syntaxonly

```

---

File 210 **lwarp-tables.sty**

§ 297     Package **tables**

*(Emulates or patches code by DONALD ARSENEAU.)*

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 211 **lwarp-tabularx.sty**

§ 298     Package **tabularx**

*(Emulates or patches code by DAVID CARLISLE.)*

Pkg    **tabularx**    **tabularx** is emulated by **lwarp**.

**for HTML output:**    Discard all options for **lwarp-tabularx**:

```

1 \LWR@ProvidesPackageDrop{tabularx}

2 \DeclareDocumentEnvironment{tabularx}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \DeclareDocumentEnvironment{tabularx*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}

```

---

File 212 **lwarp-tabulary.sty**

§ 299      Package **tabulary**

*(Emulates or patches code by DAVID CARLISLE.)*

Pkg    tabulary    tabulary is emulated by lwarp.

for HTML output:    Discard all options for lwarp-tabulary.

Column types L, C, R, and J are emulated by lwarp core code.

```

1 \LWR@ProvidesPackageDrop{tabulary}

2 \NewDocumentEnvironment{tabulary}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabulary*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}
9
10 \newdimen\tymin
11 \newdimen\tymax
12 \def\tyformat{}
```

---

File 213 **lwarp-textarea.sty**

§ 300      Package **textarea**

*(Emulates or patches code by ALEXANDER I. ROZHENKO.)*

Pkg    textarea    textarea is emulated.

for HTML output:    1 \LWR@ProvidesPackageDrop{textarea}

```

2 \newcommand\StartFromTextArea{}
3 \newcommand\StartFromHeaderArea{}
4 \newcommand*\RestoreTextArea{}
5 \newcommand*\ExpandTextArea[1][*]{}
6 \let\NCC@restoretextarea\@empty
```



---

File 214 **lwarp-textcomp.sty**


§ 301 Package **textcomp**

(Emulates or patches code by FRANK MITTELBACH, ROBIN FAIRBAIRNS, WERNER LEMBERG.)

Pkg textcomp textcomp is patched for use by lwarp.

§ 301.1 **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.

§ 301.2 **Package loading**

**for HTML output:** `1 \LWR@ProvidesPackagePass{textcomp}`

§ 301.3 **Remembering original definitions**

The following are restored for print when inside a lateximage:

```
2 \let\LWR@origtextdegree\textdegree
3 \let\LWR@origtextcelsius\textcelsius
4 \let\LWR@origtextohm\textohm
5 \let\LWR@origtextmu\textmu
6 \let\LWR@origtextlquill\textlquill
7 \let\LWR@origtexttrquill\texttrquill
8 \let\LWR@origtextcircledP\textcircledP
9 \let\LWR@origtexttwelveudash\texttwelveudash
10 \let\LWR@origtextthreequartersemdash\textthreequartersemdash
11 \let\LWR@origtextmho\textmho
12 \let\LWR@origtextnaira\textnaira
13 \let\LWR@origtextpeso\textpeso
14 \let\LWR@origtextrecipe\textrecipe
15 \let\LWR@origtextinterrobangdown\textinterrobangdown
16 \let\LWR@origtextpertenthousand\textpertenthousand
17 \let\LWR@origtextbaht\textbaht
18 \let\LWR@origtextdiscount\textdiscount
19 \let\LWR@origtextservicemark\textservicemark
20 \let\LtxMacro\LWR@origcapitalcedilla\capitalcedilla
21 \let\LtxMacro\LWR@origcapitalogonek\capitalogonek
22 \let\LtxMacro\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

```

### § 301.4 HTML symbols

For HTML, use HTML entities or direct Unicode, depending on the engine.

`\AtBeginDocument` improves support for Lua $\TeX$  and Xe $\TeX$ .

#### § 301.4.1 pdf $\TeX$ symbols

```

34 \AtBeginDocument{
35 \ifPDFTeX
36 \renewcommand*\textdegree{\HTMLentity{deg}}
37 \renewcommand*\textcelsius{\HTMLUnicode{2103}}
38 \renewcommand*\textohm{\HTMLUnicode{2126}}
39 \renewcommand*\textmu{\HTMLUnicode{00B5}}
40 \renewcommand*\textlquill{\HTMLUnicode{2045}}
41 \renewcommand*\textrquill{\HTMLUnicode{2046}}
42 \renewcommand*\textcircledP{\HTMLUnicode{2117}}
43 \renewcommand*\texttwelveudash{\HTMLUnicode{2014}}% emdash
44 \renewcommand*\textthreequartersemdash{\HTMLUnicode{2014}}% emdash
45 \renewcommand*\textmho{\HTMLUnicode{2127}}
46 \renewcommand*\textnaira{\HTMLUnicode{20A6}}
47 \renewcommand*\textpeso{\HTMLUnicode{20B1}}
48 \renewcommand*\textrecipe{\HTMLUnicode{211E}}
49 \renewcommand*\textinterrobangdown{\HTMLUnicode{2E18}}
50 \renewcommand*\textpertenthousand{\HTMLUnicode{2031}}
51 \renewcommand*\textbaht{\HTMLUnicode{0E3F}}
52 \renewcommand*\textdiscount{\}%
53 \renewcommand*\textservicemark{\HTMLUnicode{2120}}
54 \else

```

#### § 301.4.2 Xe $\TeX$ and Lua $\TeX$ symbols

NOTE: Some of the following do not print well in the listing. Consult the .dtx or .sty file for the actual characters.

```

55 \renewcommand*{\textdegree}{°}
56 \renewcommand*{\textcelsius}{°C}
57 \renewcommand*{\textohm}{Ω}
58 \renewcommand*{\textmu}{μ}
59 \renewcommand*{\textlquill}{{}}
60 \renewcommand*{\textrquill}{{}}
61 \renewcommand*{\textcircledP}{@}
62 \renewcommand*{\texttwelveudash}{--}% emdash
63 \renewcommand*{\textthreequartersemdash}{--}% emdash
64 \renewcommand*{\textmho}{Ω}
65 \renewcommand*{\textnaira}{₦}
66 \renewcommand*{\textpeso}{₱}
67 \renewcommand*{\textrecipe}{℞}
68 \renewcommand*{\textinterrobangdown}{‡}
69 \renewcommand*{\textpertenthousand}{‰}
70 \renewcommand*{\textbaht}{฿}
71 \renewcommand*{\textdiscount}{\%}
72 \renewcommand*{\textservicemark}{®}
73 \fi

```

### § 301.5 HTML diacritics

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

```

### § 301.6 Inside a lateximage

When a lateximage is begun:

```

91 \appto{\LWR@restoreorigformatting}{%
92 \let\textdegree\LWR@origtextdegree%
93 \let\textcelsius\LWR@origtextcelsius%
94 \let\textohm\LWR@origtextohm%
95 \let\textmu\LWR@origtextmu%
96 \let\textlquill\LWR@origtextlquill%
97 \let\texttrquill\LWR@origtexttrquill%
98 \let\textcircledP\LWR@origtextcircledP%
99 \let\texttwelveudash\LWR@origtexttwelveudash%
100 \let\textthreequartersemdash\LWR@origtextthreequartersemdash%
101 \let\textmho\LWR@origtextmho%
102 \let\textnaira\LWR@origtextnaira%
103 \let\textpeso\LWR@origtextpeso%
104 \let\textrecipe\LWR@origtextrecipe%
105 \let\textinterrobangdown\LWR@origtextinterrobangdown%
106 \let\textpertenthousand\LWR@origtextpertenthousand%
107 \let\textbaht\LWR@origtextbaht%
108 \let\textdiscount\LWR@origtextdiscount%
109 \let\textservicemark\LWR@origtextservicemark%
110 \LetLtxMacro\capitalcedilla\LWR@origcapitalcedilla%
111 \LetLtxMacro\capitalogonek\LWR@origcapitalogonek%
112 \LetLtxMacro\capitalgrave\LWR@origcapitalgrave%
113 \LetLtxMacro\capitalacute\LWR@origcapitalacute%
114 \LetLtxMacro\capitalcircumflex\LWR@origcapitalcircumflex%
115 \LetLtxMacro\capitaltilde\LWR@origcapitaltilde%
116 \LetLtxMacro\capitaldieresis\LWR@origcapitaldieresis%
117 \LetLtxMacro\capitalhungarumlaut\LWR@origcapitalhungarumlaut%
118 \LetLtxMacro\capitalring\LWR@origcapitalring%
119 \LetLtxMacro\capitalcaron\LWR@origcapitalcaron%
120 \LetLtxMacro\capitalbreve\LWR@origcapitalbreve%
121 \LetLtxMacro\capitalmacron\LWR@origcapitalmacron%
122 \LetLtxMacro\capitaldotaccent\LWR@origcapitaldotaccent%
123 \LetLtxMacro\textcircled\LWR@origtextcircled%
124 }
```

---

File 215 **lwarp-textfit.sty**

### § 302 Package **textfit**

Pkg **textfit** textfit is emulated.

Text is placed into a <span> of class textfit. Sizes are approximated, and also limited by browser min/max font-size settings.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{textfit}

2 \newsavebox{\LWR@textfitbox}
3
4 \newcommand*{\LWR@textfitscale}[2]{%
5 \setlength{\LWR@templengthone}{#1}%
6 \setlength{\LWR@templengthone}{%
7   1em*\ratio{\LWR@templengthone}{\LWR@templengthtwo}}%
8 }%
9 \InlineClass[font-size:\LWR@printlength{\LWR@templengthone}]{textfit}{#2}%
10 }
11
12 \newcommand*{\scaletowidth}[2]{%
13 \sbox{\LWR@textfitbox}{#2}%
14 \settowidth{\LWR@templengthtwo}{\usebox{\LWR@textfitbox}}%
15 \LWR@textfitscale{#1}{#2}%
16 }
17
18 \newcommand*{\scaletoheight}[2]{%
19 \sbox{\LWR@textfitbox}{#2}%
20 \settoheight{\LWR@templengthtwo}{\usebox{\LWR@textfitbox}}%
21 \LWR@textfitscale{#1}{#2}%
22 }

```

---

File 216 **lwarp-textpos.sty**

§ 303      Package **textpos**

*(Emulates or patches code by NORMAN GRAY.)*

Pkg    textpos    textpos is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{textpos}

2 \NewDocumentEnvironment{textblock}{m r()}{ }{}
3 \NewDocumentEnvironment{textblock*}{m o r()}{ }{}
4 \newcommand*{\TPGrid}[3][ ]{}
5 \NewDocumentCommand{\TPMargin}{s o}{}
6 \newcommand*{\textblockcolour}[1]{}
7 \newcommand*{\textblockrulecolour}[1]{}
8 \newcommand*{\textblockcolor}[1]{}
9 \newcommand*{\textblockrulecolor}[1]{}
10 \newcommand*{\tekstblokkulur}[1]{}
11 \newcommand*{\tekstblokrulekulur}[1]{}
12 \newlength{\TPHorizModule}
13 \newlength{\TPVertModule}
14 \newlength{\TPboxrulesize}

```

---

```

15 \newcommand{\textblocklabel}[1]{%
16 \newcommand*{\showtextsize}{%
17 \newcommand{\textblockorigin}[2]{%

```

---

File 217 **lwarp-theorem.sty**

§ 304     Package **theorem**

*(Emulates or patches code by FRANK MITTELBACH.)*

Pkg    theorem    theorem is patched for use by lwarp.

---

Table 14: Theorem package — CSS styling of theorems and proofs

---

**Theorem:** <div> of class theorembody<theoremstyle>

**Theorem Header:** <span> of class theoremheader

where <theoremstyle> is plain, break, etc.

---

for HTML output:    1 \LWR@ProvidesPackagePass{theorem}

### § 304.1    **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}% lwarp
8     }%
9   {%
10    \theorem@style{#1}%
11    \renewcommand{\LWR@newtheoremstyle}{#1}% lwarp
12  }%
13  \begingroup
14    \csname th@the\theorem@style \endcsname
15  \endgroup}

```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

16 \gdef\@xnthm#1#2[#3]{%
17   \expandafter\@ifdefinable\csname #1\endcsname
18   {%
19     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
20     \@definecounter{#1}\@newctr{#1}[#3]%
21     \expandafter\xdef\csname the#1\endcsname
22     {\expandafter \noexpand \csname the#3\endcsname
23      \@thmcountersep \@thmcounter{#1}}%
24     \def\@tempa{\global\@namedef{#1}}%
25     \expandafter \@tempa \expandafter{%
26       \csname th@the \theoremstyle
27       \expandafter \endcsname \the \theorembodyfont
28       \@thm{#1}{#2}}%
29     \global \expandafter \let \csname end#1\endcsname \@endtheorem
30     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% lwarp
31   }}
32
33 \gdef\@ynthm#1#2{%
34   \expandafter\@ifdefinable\csname #1\endcsname
35   {
36     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
37     \@definecounter{#1}%
38     \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
39     \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
40     \expandafter{\csname th@the \theoremstyle \expandafter
41       \endcsname \the\theorembodyfont \@thm{#1}{#2}}%
42     \global \expandafter \let \csname end#1\endcsname \@endtheorem
43     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% lwarp
44   }}
45
46 \gdef\@othm#1[#2]#3{%
47   \expandafter\ifx\csname c@#2\endcsname\relax
48     \@nocounterr{#2}%
49   \else
50     \expandafter\@ifdefinable\csname #1\endcsname
51     {
52       \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
53       \expandafter \xdef \csname the#1\endcsname
54       {\expandafter \noexpand \csname the#2\endcsname}%
55       \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
56       \expandafter{\csname th@the \theoremstyle \expandafter
57         \endcsname \the\theorembodyfont \@thm{#2}{#3}}%
58       \global \expandafter \let \csname end#1\endcsname \@endtheorem
59       \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% lwarp
60     }%
61   \fi}

```

## § 304.2 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{%
63   \def\@begintheorem##1##2{%
64     \item[
65       \InlineClass{theoremheader}{##1\ ##2}
66     ]
67   }%
68 \def\@opargbegintheorem##1##2##3{%
69   \item[
70     \InlineClass{theoremheader}{##1\ ##2\ (##3)}
71   ]
72   }
73 }
74
75 \gdef\th@break{%
76   \def\@begintheorem##1##2{%
77     \item[
78       \InlineClass{theoremheader}{##1\ ##2}\newline%
79     ]
80   }%
81 \def\@opargbegintheorem##1##2##3{%
82   \item[
83     \InlineClass{theoremheader}{##1\ ##2\ (##3)}\newline
84   ]
85   }
86 }
87
88 \gdef\th@marginbreak{%
89   \def\@begintheorem##1##2{
90     \item[
91       \InlineClass{theoremheader}{##2 \quad ##1}\newline
92     ]
93   }%
94 \def\@opargbegintheorem##1##2##3{%
95   \item[
96     \InlineClass{theoremheader}{##2 \quad ##1\ %
97     (##3)}\newline
98   ]
99   }
100 }
```



```

101
102 \gdef\th@changebreak{%
103   \def\@begintheorem##1##2{
104     \item[
105       \InlineClass{theoremheader}{##2\ ##1}\newline
106     ]
107   }%
108 \def\@opargbegintheorem##1##2##3{%
109   \item[
110     \InlineClass{theoremheader}{ ##2\ ##1\ %
111       (##3)}\newline
112   ]
113   }
114 }
115
116 \gdef\th@change{%
117   \def\@begintheorem##1##2{
118     \item[
119       \InlineClass{theoremheader}{##2\ ##1}
120     ]
121   }%
122 \def\@opargbegintheorem##1##2##3{%
123   \item[
124     \InlineClass{theoremheader}{##2\ ##1\ (##3)}
125   ]
126   }
127 }
128
129 \gdef\th@margin{%
130   \def\@begintheorem##1##2{
131     \item[
132       \InlineClass{theoremheader}{##2 \quad ##1}
133     ]
134   }%
135 \def\@opargbegintheorem##1##2##3{%
136   \item[
137     \InlineClass{theoremheader}{##2 \quad ##1\ (##3)}
138   ]
139   }
140 }

```

Patched for CSS:

```

141 \gdef\@thm#1#2{\refstepcounter{#1}%
142 \LWR@forcenewpage% lwarp
143   \BlockClass{theorembody\LWR@thisthmstyle}% lwarp
144   \trivlist
145   \@topsep \theorempreskipamount           % used by first \item
146   \@topsepadd \theorempostskipamount        % used by \@endparenv

```

---

```

147 \@ifnextchar [%
148 {\@ythm{#1}{#2}}}%
149 {\@begintheorem{#2}{\csname the#1\endcsname}\ignorespaces}}
150
151 \gdef\@endtheorem{%
152 \endtrivlist
153 \endBlockClass
154 }

```

---

File 218 **lwarp-threeparttable.sty**

\$ 305     Package **threeparttable**

*(Emulates or patches code by DONALD ARSENEAU.)*

Pkg    **threeparttable**    threeparttable is emulated.

Table note are contained inside a css <div> of class tnotes. If enumitem is used, the note item labels are also individually highlighted with an additional css <span> of class tnoteitemheader, otherwise they are plain text.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{threeparttable}

\LWR@printtablenote    {\<text>}

Prints the table note item header inside a css class of tnoteitemheader.

```
2 \newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}
```

Env    **threeparttable**    [*<alignment>*] To emulate threeparttable:

```
3 \newenvironment*{threeparttable}[1][b]{-}{-}
```

Env    **tablenotes**    [*<options>*]

```

4 \newenvironment*{tablenotes}[1][
5 {%
6 \LWR@forcenewpage
7 \BlockClass{tnotes}%
8 \ltx@ifpackageloaded{enumitem}{%
9 \setlist[description]{format=\LWR@printtablenote}%
10 }{}%
11 \description%
12 }
13 {%
14 \enddescription%
15 \endBlockClass%

```

---

```
16 }
```

```
\tnote {⟨text⟩}
```

```
17 \newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}
```

---

File 219 **lwarp-tikz.sty**

§ 306 Package **tikz**

*(Emulates or patches code by TILL TANTAU.)*

Pkg tikz tikz is supported.

Accept all options for lwarp-tikz:

```
1 \LWR@ProvidesPackagePass{tikz}
```

**catcodes** lwarp changes the catcode of \$ for its own use. The Tikz babel library temporarily changes catcodes back to normal for Tikz's use. tikz v3.0.0 introduced the babel library which handles catcode changes. For older versions, lwarp must change \$'s catcode itself.

Also see:

<https://tex.stackexchange.com/questions/16199/test-if-a-package-or-package-option-is-loaded>

```
2 \newboolean{LWR@tikzbabel}
3
4 \@ifpackagelater{tikz}{2013/12/20}% Test for Tikz version v3.0.0
5 {\usetikzlibrary{babel}\booltrue{LWR@tikzbabel}}
6 {\boolfalse{LWR@tikzbabel}}
```

Env tikzpicture tikzpicture environment is enclosed inside a \lateximage. May be used as-is, and its contents will be converted to an image.

```
7 \BeforeBeginEnvironment{tikzpicture}{%
8 \begin{lateximage}%
9 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
10 {}%
11 {\catcode'\$=3}% dollar sign is math shift
12 }
13
14 \AfterEndEnvironment{tikzpicture}{%
15 \end{lateximage}%
```

---

```

16 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
17 {}%
18 {\catcode'\$=\active}%
19 }

```

---

File 220 **lwarp-titles.sty**

## § 307 Package **titles**

*(Emulates or patches code by JAVIER BEZOS.)*

Pkg titles titles is loaded and used by lwarp during HTML output. All user options and macros are ignored and disabled.

Discard all options for lwarp-titles:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{titles}

\pagestyle and \thispagestyle are already disabled in the lwarp code.

```

\newpagestyle {\langle name \rangle} [\langle style \rangle] {\langle commands \rangle}
2 \NewDocumentCommand{\newpagestyle}{m o m}{}

```

```

\renewpagestyle {\langle name \rangle} [\langle style \rangle] {\langle commands \rangle}
3 \NewDocumentCommand{\renewpagestyle}{m o m}{}

```

```

\sethead [\langle el \rangle] [\langle ec \rangle] [\langle er \rangle] {\langle ol \rangle} {\langle oc \rangle} {\langle or \rangle}
4 \NewDocumentCommand{\sethead}{o o o m m m}{}

```

```

\setfoot [\langle el \rangle] [\langle ec \rangle] [\langle er \rangle] {\langle ol \rangle} {\langle oc \rangle} {\langle or \rangle}
5 \NewDocumentCommand{\setfoot}{o o o m m m}{}

```

```

\settitledmarks * {\langle names \rangle}
6 \NewDocumentCommand{\settitledmarks}{s m}{}

```

```

\headrule
7 \newcommand*{\headrule}{}

```

---

<code>\footrule</code>	<code>8 \newcommand*{\footrule}{}</code>
<code>\setheadrule</code>	<code>{\langle length \rangle}</code> <code>9 \newcommand*{\setheadrule}[1]{} </code>
<code>\setfootrule</code>	<code>{\langle length \rangle}</code> <code>10 \newcommand*{\setfootrule}[1]{} </code>
<code>\makeheadrule</code>	<code>11 \newcommand*{\makeheadrule}{} </code>
<code>\makefootrule</code>	<code>12 \newcommand*{\makefootrule}{} </code>
<code>\setmarkboth</code>	<code>{\langle code \rangle}</code> <code>13 \newcommand{\setmarkboth}[1]{} </code>
<code>\widenhead</code>	<code>14 \NewDocumentCommand{\widenhead}{s o o m m}{} </code>
<code>\bottitlemarks</code>	<code>15 \newcommand*{\bottitlemarks}{} </code>
<code>\toptitlemarks</code>	<code>16 \newcommand*{\toptitlemarks}{} </code>
<code>\firsttitlemarks</code>	<code>17 \newcommand*{\firsttitlemarks}{} </code>
<code>\nexttitlemarks</code>	<code>18 \newcommand*{\nexttoptitlemarks}{} </code>
<code>\outertitlemarks</code>	<code>19 \newcommand*{\outertitlemarks}{} </code>

\innertitlemarks

20 \newcommand\*\innertitlemarks{}\}

\newtitlemark \* {\langle name \rangle}

21 \NewDocumentCommand{\newtitlemark}{s m}{}\}

\pretitlemark \* {\langle section \rangle} {\langle text \rangle}

22 \NewDocumentCommand{\pretitlemark}{s m m}{}\}

\ifsamemark {\langle group \rangle} {\langle command \rangle} {\langle true \rangle} {\langle false \rangle}

23 \newcommand{\ifsamemark}[4]{}\}

\setfloathead \* [\langle. \rangle] [\langle. \rangle] [\langle. \rangle] {\langle. \rangle} {\langle. \rangle} {\langle. \rangle} {\langle extra \rangle} [\langle which \rangle]

24 \NewDocumentCommand{\setfloathead}{s o o o m m m m m}{}\}

\setfloatfoot \* [\langle. \rangle] [\langle. \rangle] [\langle. \rangle] {\langle. \rangle} {\langle. \rangle} {\langle. \rangle} {\langle extra \rangle} [\langle which \rangle]

25 \NewDocumentCommand{\setfloatfoot}{s o o o m m m m m}{}\}

\nextfloathead \* [\langle. \rangle] [\langle. \rangle] [\langle. \rangle] {\langle. \rangle} {\langle. \rangle} {\langle. \rangle} {\langle extra \rangle} [\langle which \rangle]

26 \NewDocumentCommand{\nextfloathead}{s o o o m m m m m}{}\}

\nextfloatfoot \* [\langle. \rangle] [\langle. \rangle] [\langle. \rangle] {\langle. \rangle} {\langle. \rangle} {\langle. \rangle} {\langle extra \rangle} [\langle which \rangle]

27 \NewDocumentCommand{\nextfloatfoot}{s o o o m 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 221 **lwarp-titleref.sty**

§ 308     Package **titleref**

Pkg    titleref    titleref is emulated.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{titleref}
2
3 \LetLtxMacro\titleref\nameref
4
5 \providecounter{LWR@currenttitle}
6
7 \newcommand*{\currenttitle}{%
8   \addtocounter{LWR@currenttitle}{1}%
9   \label{currenttitle\arabic{LWR@currenttitle}}%
10  \nameref{currenttitle\arabic{LWR@currenttitle}}%
11 }
12
13 \newcommand*{\theTitleReference}[2]{}

```

---

File 222 **lwarp-titlesec.sty**

§ 309     Package **titlesec**

*(Emulates or patches code by JAVIER BEZOS.)*

Pkg titlesec titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titlesec:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{titlesec}

\titlelabel {\langle label-format \rangle}

2 \newcommand\*{\titlelabel}[1]{}

\titleformat\* {\langle command \rangle} {\langle format \rangle}

\titleformat {\langle command \rangle} [\langle shape \rangle] {\langle format \rangle} {\langle label \rangle} {\langle sep \rangle} {\langle before \rangle} [\langle after \rangle]

3 \newcommand\titleformat{%

4 \ifstar{\ttl@format@s}%

5 {\ttl@format@i}}

6 \newcommand{\ttl@format@s}[1]{}

7 \NewDocumentCommand{\ttl@format@i}{m o m m m o}{}

\chaptertitlename

8 \@ifundefined{@chapapp}{\let\@chapapp\chaptername}{}%

9 \newcommand\chaptertitlename{\@chapapp}

\titlespacing \* {\langle command \rangle} {\langle left \rangle} {\langle before \rangle} {\langle after \rangle} [\langle right \rangle]

10 \NewDocumentCommand{\titlespacing}{s m m m m o}{}

\filright

11 \newcommand\*{\filright}{}%

\filcenter

12 \newcommand\*{\filcenter}{}%

\filleft

13 \newcommand\*{\filleft}{}%

\fillast

14 \newcommand\*{\fillast}{}%

\filinner



---

```

15 \newcommand*\filinner{}

\filouter

16 \newcommand*\filouter{}

\wordsep

17 \newcommand\wordsep{\fontdimen\tw@font \@plus
18 \fontdimen\thr@@font \@minus \fontdimen4font}

\titeline * [\langle align \rangle] {\langle material \rangle}

19 \NewDocumentCommand{\titeline}{s o m}{}

\titlerule [\langle height \rangle]

20 \providecommand*\titlerule{\@ifstar{\ttl@row}{\ttl@rule}}
21 \newcommand*\ttl@rule[1] [] {}
22 \newcommand*\ttl@row[2] [] {}

\iftitlemeasuring {\langle true \rangle} {\langle false \rangle}

23 \newcommand{\iftitlemeasuring}[2] {\#2}

\assignpagestyle {\langle command \rangle} {\langle pagestyle \rangle}

24 \newcommand{\assignpagestyle}[2] {\#2}

\titleclass {\langle name \rangle} [\langle startlevel \rangle] {\langle class \rangle} [\langle cmd \rangle]

25 \NewDocumentCommand{\titleclass}{m o m o}

```

---

File 223 **lwarp-titletoc.sty**

§ 310 Package **titletoc**

*(Emulates or patches code by JAVIER BEZOS.)*

Pkg titletoc titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titletoc:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{titletoc}

---

```

\dottedcontents  {\<section>} [\<left>] {\<above>} {\<label>} {\<leader>}
                  2 \NewDocumentCommand{\dottedcontents}{m o m m m}{\}

\titlecontents  * {\<section>} [\<left>] {\<above>} {\<numbered>} {\<numberless>} {\<filler>} [\<below
or begin>] [\<separator>] [\<end>]
                  3 \newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
                  4 \NewDocumentCommand{\ttl@tcstar}{m o m m m m o o}{\}
                  5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m m o}{\}

\contentsmargin  [\<correction>] {\<right>}
                  6 \newcommand{\contentsmargin}[2] [] {\}

\thecontentslabel
                  7 \newcommand*{\thecontentslabel}{thecontentslabel}

\thecontentspage
                  8 \newcommand*{\thecontentspage}{thecontentspage}

\contentslabel   [\<format>] {\<space>}
                  9 \newcommand{\contentslabel}[2] [] {\thecontentslabel}

\contentspage    [\<format>]
                  10 \newcommand{\contentspage}[1] [] {\thecontentspage}

\contentspush    {\<text>}
                  11 \newcommand{\contentspush}[1] {\}

\contentsuse     {\<name>} {\<text>}
                  12 \newcommand{\contentsuse}[2] {\}

\startcontents   [\<name>]
                  13 \newcommand*{\startcontents}[1] [] {\}

\stopcontents    [\<name>]
                  14 \newcommand*{\stopcontents}[1] [] {\}

```

---

```

\resumecontents  [\langle name \rangle]
                  15 \newcommand*{\resumecontents}[1] [] {}

\printcontents  [\langle name \rangle] {\langle prefix \rangle} {\langle start \rangle} {\langle code \rangle}
                  16 \newcommand{\printcontents}[4] [] {}

\startlist      [\langle name \rangle] {\langle list \rangle}
                  17 \newcommand{\startlist}[2] [] {}

\stoplist       [\langle name \rangle] {\langle list \rangle}
                  18 \newcommand{\stoplist}[2] [] {}

\resumelist     [\langle name \rangle] {\langle list \rangle}
                  19 \newcommand{\resumelist}[2] [] {}

\printlist      [\langle name \rangle] {\langle list \rangle} {\langle prefix \rangle} {\langle code \rangle}
                  20 \newcommand{\printlist}[4] [] {}

```

---

File 224 **lwarp-titling.sty**

§ 311     Package **titling**

*(Emulates or patches code by PETER WILSON.)*

Pkg    titling

**package support**    lwarp supports the native  $\LaTeX$  titling commands, and also supports the packages  
**load order**        authblk and titling. If both are used, authblk should be loaded before titling.

**\published and \subtitle**    If using the titling package, additional titlepage fields for \published and \subtitle  
                                  may be added by using \AddSubtitlePublished in the preamble. See section 58.8.

The various titling footnote restyling commands have no effect.

Pass all options to lwarp-titling:

**for HTML output:**    1 \LWR@ProvidesPackagePass{titling}

**\@bsmtitleempty**    Patch \@bsmtitleempty:

```

2 \let\LWR@orig@bsmtitleempty\@bsmtitleempty
3 \renewcommand*{\@bsmtitleempty}{%
4 \LWR@orig@bsmtitleempty%
5 }

```

\keepthetitle Patch \keepthetitle:

```

6 \let\LWR@origkeepthetitle\keepthetitle
7 \renewcommand*{\keepthetitle}{%
8 \LWR@orig@keepthetitle%
9 }

```

\killtitle Patch \killtitle:

```

10 \let\LWR@origkilltitle\killtitle
11 \renewcommand*{\killtitle}{%
12 \LWR@orig@killtitle%
13 }

```

Env titlingpage

```

14 \renewenvironment*{titlingpage}
15 {%

```

Start an HTML titlepage div:

```

16 \LWR@printpendingfootnotes
17 \begin{titlepage}

```

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

```

18 \LWR@maketitlesetup
19 \let\maketitle\LWR@titlingmaketitle
20 }
21 {

```

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

```

22 \end{titlepage}
23 }

```

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

```

24
25 \pretitle{}
26 \posttitle{}
27

```

```

28 \preauthor{}
29 \postauthor{}
30
31 \predate{}
32 \postdate{}

```

`\LWR@maketitlesetup` Patches `\thanks` macros.

```

33 \renewcommand*{\LWR@maketitlesetup}{%

```

Redefine the footnote mark:

```

34 \def\@makefnmark{\textsuperscript{\@thefnmark}}

```

```

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

```

Redefine the footnote text:

```

35 \long\def\@makefntext##1{%

```

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

```

36 \makethanksmark~%

```

```

\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒
\@thefnmark ⇒ \itshape a (or similar)

```

Print the text:

```

37 ##1%
38 }% \@makefntext
39 }

```

`\maketitle` HTML mode. Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

```

40 \renewcommand*{\maketitle}{%

```

An HTML titlepage `<div>` is used for all classes.

```

41 \begin{titlepage}

```

Select which kind of footnote marks to use:

```
42 \@bsmarkseries
```

Set up special patches:

```
43 \LWR@maketitlesetup
```

Typeset the title, etc:

```
44 \@maketitle
```

Immediately generate any \thanks footnotes:

```
45 \@thanks
```

Close the HTML titlepage div:

```
46 \end{titlepage}
```

Reset the footnote counter:

```
47 \@bscontmark
48 }
```

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

```
49 \DeclareDocumentCommand{\@maketitle}{-}{%
50   \maketitlehooka
51   {
52     \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
53     \@bsprettitle \@title \@bsposttitle
54     \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
55   }
56   \maketitlehookb
57   {
58     \begin{BlockClass}{author}
59     \renewcommand{\and}{\and}
60     \end{BlockClass}
61     \begin{BlockClass}{oneauthor}
62     }
63     \begin{BlockClass}{oneauthor}
64     \@bspreauthor \@author \@bspostauthor
65     \end{BlockClass}
66     \end{BlockClass}
67   }
68   \maketitlehookc
```

```

69   {
70       \begin{BlockClass}{titledat}
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 \@thanks
```

Select which kind of footnote marks to use:

```
78 \@bsmarkseries
```

Set up special patches:

```
79 \LWR@maketitlesetup
```

Typeset the title, etc:

```
80 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
81 \@thanks
```

Reset the footnote counter:

```
82 \@bscontmark
83 }
```

`\thanksmarkseries` `{\langle series \rangle}`

Sets the type of footnote marks used by `\thanks`, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc.

```

84 \renewcommand{\thanksmarkseries}[1]{%
85 \def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
86 }

```

Set default titlepage thanks footnote marks. See section 58.7.

```

87 \@ifclassloaded{memoir}{
88   \thanksmarkseries{arabic}
89 }{% not memoir
90 \if@titlepage
91   \thanksmarkseries{arabic}
92 \else
93   \thanksmarkseries{fnsymbol}
94 \fi
95 }% not memoir

```

---

File 225 **lwarp-tocbasic.sty**

§ 312     Package **tocbasic**

*(Emulates or patches code by MARKUS KOHM.)*

Pkg    tocbasic    tocbasic is patched for use by lwarp.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. `\DeclareDocumentCommand` is used to overwrite the koma-script definitions.

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{tocbasic}

2 \DeclareDocumentCommand{\usetocbasicnumberline}{o}{\}
3 \DeclareDocumentCommand{\DeclareTOCStyleEntry}{o m m}{\}
4 \DeclareDocumentCommand{\DeclareTOCEntryStyle}{m o m}{\}
5 \DeclareDocumentCommand{\DefineTOCEntryOption}{m o m}{\}
6 \DeclareDocumentCommand{\DefineTOCEntryBooleanOption}{m o m m m}{\}
7 \DeclareDocumentCommand{\DefineTOCEntryCommandOption}{m o m m m}{\}
8 \DeclareDocumentCommand{\DefineTOCEntryIfOption}{m o m m m}{\}
9 \DeclareDocumentCommand{\DefineTOCEntryLengthOption}{m o m m m}{\}
10 \DeclareDocumentCommand{\DefineTOCEntryNumberOption}{m o m m m}{\}
11 \DeclareDocumentCommand{\CloneTOCEntryStyle}{m m}{\}
12 \DeclareDocumentCommand{\TOCEntryStyleInitCode}{m m}{\}
13 \DeclareDocumentCommand{\TOCEntryStyleStartInitCode}{m m}{\}

```

---

File 226 **lwarp-tocbibind.sty**


§ 313     Package **tocbibind**

*(Emulates or patches code by PETER WILSON.)*



Pkg `tocbibind` `tocbibind` is patched for use by `lwarp`.

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

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

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

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

A commonly-used method to introduce an index in a  $\TeX$  document:

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

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

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

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

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

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

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

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

Opt `tocbibind` `numindex` Use the `tocbibind` `numindex` option to generate a numbered index. Without this

numbered index section 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 \DeclareDocumentCommand{\simplechapter}{0{\@empty}}{%
34     \def\@chapcntformat##1{%
35         #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
36     }%
37 }
38
39 \DeclareDocumentCommand{\restorechapter}{1}{%
40 \let\@chapcntformat\@seccntformat%
41 }

```


---

File 227 **lwarp-tocloft.sty**

§ 314 Package **tocloft**

*(Emulates or patches code by PETER WILSON.)*

Pkg tocloft tocloft is emulated. Most user options and macros are ignored and disabled. \newlistof and \cftchapterprecis are supported.

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

Discard all options for lwarp-tocloft:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{tocloft}

\tocloftpagestyle {\langle style \rangle}  
2 \newcommand{\tocloftpagestyle}[1]{}  
3 \newcommand\*{\cftmarktoc}{}

\cftmarktoc  
3 \newcommand\*{\cftmarktoc}{}

\cfttoctitlefont  
4 \newcommand\*{\cfttoctitlefont}{}

\cftaftertoctitle  
5 \newcommand\*{\cftaftertoctitle}{}  
6 \newlength{\cftbeforetoctitleskip}  
7 \newlength{\cftaftertoctitleskip}

\cftmarklof  
8 \newcommand\*{\cftmarklof}{}

\cftlofttitlefont  
9 \newcommand\*{\cftlofttitlefont}{}

`\cftafterlofttitle`

```
10 \newcommand*{\cftafterlofttitle}{}

11 \newlength{\cftbeforelofttitleskip}
12 \newlength{\cftafterlofttitleskip}
```

`\cftmarklot`

```
13 \newcommand*{\cftmarklot}{}{}
```

`\cftlottitlefont`

```
14 \newcommand*{\cftlottitlefont}{}{}
```

`\cftafterlotttitle`

```
15 \newcommand*{\cftafterlotttitle}{}

16 \newlength{\cftbeforelotttitleskip}
17 \newlength{\cftafterlotttitleskip}
```

`\cftdot`

```
18 \providecommand*{\cftdot}{.}
```

`\cftdotsep`

```
19 \providecommand*{\cftdotsep}{1}
```

`\cftnodots`

```
20 \providecommand*{\cftnodots}{5000}
```

`\cftdotfill`     $\{\langle sep \rangle\}$

```
21 \providecommand{\cftdotfill}[1]{}{}
```

`\cftsetpnumwidth`     $\{\langle length \rangle\}$

```
22 \DeclareDocumentCommand{\cftsetpnumwidth}{m}{}{}
```

`\cftsetrmarg`     $\{\langle length \rangle\}$

```
23 \DeclareDocumentCommand{\cftsetrmarg}{m}{}{}
```

`\cftpnumalign`     $\{\langle alignment \rangle\}$

```
24 \DeclareDocumentCommand{\cftpnumalign}{m}{}

```

```
25 \LWR@providelength{\cftparskip}

```

The part-related items are also provided by memoir:

```
26 \LWR@providelength{\cftbeforepartskip}
27 \LWR@providelength{\cftpartincent}
28 \LWR@providelength{\cftpnumwidth}
29 \providecommand*\cftpfont{}
30 \providecommand*\cftpresnum{}
31 \providecommand*\cftpartaftersnum{}
32 \providecommand*\cftpartaftersnumb{}
33 \providecommand*\cftpartleader{}
34 \providecommand*\cftpartdotsep{1}
35 \providecommand*\cftpartpagefont{}
36 \providecommand*\cftpartafterpnum{}

```

memoir uses the full name “chapter” instead of “chap”:

```
37 \LWR@providelength{\cftbeforechapskip}
38 \LWR@providelength{\cftchapindent}
39 \LWR@providelength{\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{}

```

The following do not appear in memoir:

```
48 \LWR@providelength{\cftbeforesecskip}
49 \LWR@providelength{\cftsecindent}
50 \LWR@providelength{\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 \LWR@providelength{\cftbeforesubsecskip}
60 \LWR@providelength{\cftsubsecindent}
61 \LWR@providelength{\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 \LWR@providelength{\cftbeforesubsubsecskip}
71 \LWR@providelength{\cftsubsubsecindent}
72 \LWR@providelength{\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 \LWR@providelength{\cftbeforeparaskip}
82 \LWR@providelength{\cftparaindent}
83 \LWR@providelength{\cftparanumwidth}
84 \newcommand*{\cftparafont}{}
85 \newcommand*{\cftparapresnum}{}
86 \newcommand*{\cftparaaftersnum}{}
87 \newcommand*{\cftparaaftersnumb}{}
88 \newcommand*{\cftparaleader}{}
89 \newcommand*{\cftparadotsep}{1}
90 \newcommand*{\cftparapagefont}{}
91 \newcommand*{\cftparaafterpnum}{}

92 \LWR@providelength{\cftbeforesubparaskip}
93 \LWR@providelength{\cftsubparaindent}
94 \LWR@providelength{\cftsubparanumwidth}
95 \newcommand*{\cftsubparafont}{}
96 \newcommand*{\cftsubparapresnum}{}
97 \newcommand*{\cftsubparaaftersnum}{}
98 \newcommand*{\cftsubparaaftersnumb}{}
99 \newcommand*{\cftsubparaleader}{}
100 \newcommand*{\cftsubparadotsep}{1}
101 \newcommand*{\cftsubparapagefont}{}
102 \newcommand*{\cftsubparaafterpnum}{}

103 \LWR@providelength{\cftbeforefigskip}
```

```
104 \LWR@providelength{\cftfigindent}  
105 \LWR@providelength{\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 \LWR@providelength{\cftbeforesubfigskip}  
115 \LWR@providelength{\cftsubfigindent}  
116 \LWR@providelength{\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 \LWR@providelength{\cftbeforetabskip}  
126 \LWR@providelength{\cfttabindent}  
127 \LWR@providelength{\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 \LWR@providelength{\cftbeforesubtabskip}  
137 \LWR@providelength{\cftsubtabindent}  
138 \LWR@providelength{\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 \DeclareDocumentCommand{\cftsetindents}{m m m}{}  

```

```

148 \newcommand{\pagenumbersoff}[1]{%
149 \newcommand{\pagenumberson}[1]{%

```

`\newlistentry` [*<within>*] [*<counter>*] [*<ext>*] [*<level-1>*]

```

150 \DeclareDocumentCommand{\newlistentry}{o m m m}
151 {%
152 \LWR@traceinfo{newlistentry #2 #3 #4}%
153 \IfValueTF{#1}%
154 {%
155     \@ifundefined{c@#2}{%
156         \newcounter{#2}[#1]%
157         \expandafter\edef\csname the#2\endcsname{%
158             \expandafter\noexpand\csname the#1\endcsname.\noexpand\arabic{#2}%
159         }%
160     }{}%
161 }%
162 {%
163     \@ifundefined{c@#2}{%
164         \newcounter{#2}%
165     }{}%
166 }%
167 \@namedef{l@#2}##1##2{%
168     \hypertocfloat{1}{#2}{#3}{##1}{##2}%
169     \def\cftwhatismyname{#2}% from memoir
170 }%
171 \expandafter\newlength\csname cftbefore#2skip\endcsname%
172 \expandafter\newlength\csname cft#2indent\endcsname%
173 \expandafter\newlength\csname cft#2numwidth\endcsname%
174 \@namedef{cft#2font}{}%
175 \@namedef{cft#2presnum}{}%
176 \@namedef{cft#2aftersnum}{}%
177 \@namedef{cft#2aftersnumb}{}%
178 \@namedef{cft#2leader}{}%
179 \@namedef{cft#2dotsep}{1}%
180 \@namedef{cft#2pagefont}{}%
181 \@namedef{cft#2afterpnum}{}%
182 \@namedef{toclevel@#2}{#4}%
183 \@namedef{cft#2fillnum}##1{%
184 \LWR@traceinfo{newlistentry done}%
185 }

```

`\newlistof` [*<within>*] [*<type>*] [*<ext>*] [*<listofname>*]

Emulated through the `\newfloat` mechanism.

```

186 \DeclareDocumentCommand{\newlistof}{o m m m}
187 {%
188 \IfValueTF{#1}

```



```

189 {\newlistentry[#1]{#2}{#3}{0}}
190 {\newlistentry{#2}{#3}{0}}
191 \@namedef{ext@#2}{#3}
192 \@ifundefined{c@#3depth}{\newcounter{#3depth}}{}
193 \setcounter{#3depth}{1}
194 \@namedef{cftmark#3}{}
195 \@namedef{listof#2}{\listof{#2}{#4}}
196 \@namedef{@cftmake#3title}{}
197 \expandafter\newlength\csname cftbefore#3titleskip\endcsname
198 \expandafter\newlength\csname cftafter#3titleskip\endcsname
199 \@namedef{cft#3titlefont}{}
200 \@namedef{cftafter#3title}{}
201 \@namedef{cft#3prehook}{}
202 \@namedef{cft#3posthook}{}
203 }

```

`\cftchapterprecis`     $\langle text \rangle$

```

204 \newcommand{\cftchapterprecis}[1]{%
205   \cftchapterprecishere{#1}
206   \cftchapterprecistoc{#1}}
207 \newcommand{\cftchapterprecishere}[1]{%
208   \begin{quote}\textit{#1}\end{quote}}
209 \newcommand{\cftchapterprecistoc}[1]{
210   \addtocontents{toc}{%
211     {
212       \protect\begin{quote}#1\protect\end{quote}}
213   }
214 }

```

---

File 228    **lwarp-tocstyle.sty**

§ 315    Package    **tocstyle**

Pkg    `tocstyle`    `tocstyle` is ignored.

 **Not fully tested!**    [Please send bug reports!](#)

**for HTML output:**    `1 \LWR@ProvidesPackageDrop{tocstyle}`

```

2 \newcommand*{\usetocstyle}[2] [] {}
3 \newcommand*{\deactivatetocstyle}[1] [] {}
4 \newcommand*{\reactivatetocstyle}[1] [] {}
5 \NewDocumentCommand{\settocfeature}{o o m m}{}
6 \NewDocumentCommand{\settocstylefeature}{o m m}{}
7 \NewDocumentCommand{\newtocstyle}{o o m m}{}
8 \newcommand*{\aliastoc}[2]{}

```

```

9 \newcommand*\showtoc}[2] [] {}
10 \newcommand{\iftochasdepth}[4] {}

```

File 229 **lwarp-todo.sty**

§ 316 Package **todo**

*(Emulates or patches code by FEDERICO GARCIA.)*

Pkg todo todo is patched for use by lwarp.

for HTML output:

```

1 \LWR@ProvidesPackagePass{todo}

2 \renewcommand\todoitem[2] {%
3   \refstepcounter{todo}%
4   \item[%
5     \HTMLUnicode{2610} \quad
6     \ref{todopage:\thetodo}
7   ] : {\todoformat\ifx#1\todomark\else\textbf{#1} \fi}#2%
8   \label{todolbl:\thetodo}%
9 }%
10
11 \renewcommand\doneitem[2] {%
12   \stepcounter{todo}%
13   \item[%
14     \HTMLUnicode{2611} \quad
15     \ref{todopage:\thetodo}
16   ] \@nameuse{@done\the\c@todo}:
17     {\todoformat\ifx#1\todomark\else\textbf{#1} \fi}#2%
18 }
19
20 \xpatchcmd{\@displaytodo}
21   {\todoformat #1}{\todoformat \textbf{#1}}{}
22   {\PackageWarning{lwarp-todo}{Unable to patch @displaytodo.}}
23
24 \xpatchcmd{\@displayfulltodo}
25   {\todoformat #1}{\todoformat \textbf{#1}}{}
26   {\PackageWarning{lwarp-todo}{Unable to patch @displayfulltodo.}}
27
28 \patchcmd{\todoenv}{\itshape see text.}{\textit{see text.}}{}
29   {\PackageWarning{lwarp-todo}{Unable to patch todoenv.}}
30
31 \patchcmd{\astodos}{\todoformat #1}{\todoformat \textbf{#1}}{}
32   {\PackageWarning{lwarp-todo}{Unable to patch astodos.}}
33
34 \AtBeginDocument{
35 \crefname{todo}{todo}{todos}

```

---

```

36 \Crefname{todo}{Todo}{Todos}
37 }

```

---

File 230 **lwarp-todonotes.sty**

§ 317     Package **todonotes**

*(Emulates or patches code by HENRIK SKOV MIDTIBY.)*

Pkg    todonotes    todonotes is emulated.

The documentation for todonotes and luatodonotes have an example with a todo inside a caption. If this example does not work it will be necessary to move the todo outside of the caption.

**for HTML output:**

```

1 \LWR@ProvidesPackagePass{todonotes}

2 \if@todonotes@disabled
3 \else
4
5 \newcommand{\ext@todo}{tdo}
6
7 \renewcommand{\l@todo}[2]{\hypertocfloat{1}{\todo}{ldo}{#1}{#2}}

8 \let\LWRTODONOTES@orig@todototoc\todototoc
9
10 \renewcommand*{\todototoc}{%
11 \phantomsection%
12 \LWRTODONOTES@orig@todototoc%
13 }
14
15 \renewcommand{\@todonotes@drawMarginNoteWithLine}{%
16 \fcolorbox
17   {\@todonotes@currentbordercolor}
18   {\@todonotes@currentbackgroundcolor}
19   {\arabic{\@todonotes@numberoftodonotes}}
20 \marginpar{\@todonotes@drawMarginNote}
21 }
22
23 \renewcommand{\@todonotes@drawInlineNote}{%
24 \fcolorboxBlock%
25   {\@todonotes@currentbordercolor}%
26   {\@todonotes@currentbackgroundcolor}%
27   {%
28     \if@todonotes@authorgiven%
29     {\@todonotes@author:\,}%

```

```

30      \fi%
31      \@todonotes@text%
32  }%
33 }
34
35 \renewcommand{\@todonotes@drawMarginNote}{%
36   \if@todonotes@authorgiven%
37     \@todonotes@author\par%
38   \fi%
39   \arabic{\@todonotes@numberoftodonotes}: %
40   \fcolorbox%
41     {\@todonotes@currentbordercolor}%
42     {\@todonotes@currentbackgroundcolor}%
43     {%
44       \@todonotes@sizecommand%
45       \@todonotes@text %
46     }%
47 }%
48
49 \renewcommand{\@todonotes@drawLineToRightMargin}{}
50
51 \renewcommand{\@todonotes@drawLineToLeftMargin}{}
52
53 \renewcommand{\missingfigure}[2] [] {%
54   \setkeys{todonotes}{#1}%
55   \addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
56   \fcolorboxBlock%
57     {\@todonotes@currentbordercolor}%
58     {\@todonotes@currentfigcolor}%
59     {%
60       \setlength{\fboxrule}{4pt}%
61       \fcolorbox{red}{white}{Missing figure} \quad #2%
62     }
63 }
64
65 \LetLtxMacro\LWRTODONOTES@orig@todo\@todo
66
67 \RenewDocumentCommand{\@todo}{o m}{%
68   \begingroup%
69   \renewcommand*\phantomsection{}%
70   \IfValueTF{#1}{%
71     \LWRTODONOTES@orig@todo[#1]{#2}%
72   }{%
73     \LWRTODONOTES@orig@todo{#2}%
74   }
75   \endgroup%
76 }
77
78 \fi% \if@todonotes@disabled

```

---

File 231 **lwarp-transparent.sty**

§ 318     Package **transparent**

*(Emulates or patches code by HEIKO OBERDIEK.)*

Pkg   transparent    Emulated. `\texttransparent` works for inline objects. `\transparent` only works for `\includegraphics`.

 **Not  $\LaTeX$ !** Note that transparent does not work with  $\LaTeX$ .

**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 232 **lwarp-trivfloat.sty**

§ 319     Package **trivfloat**

*(Emulates or patches code by JOSEPH WRIGHT.)*

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.

To create a new float type and change its name:

---

```
\trivfloat{example}
\renewcommand{\examplename}{Example Name}
\crefname{example}{example}{examples}
\Crefname{example}{Example}{Examples}
```

---

```
1 \LWR@ProvidesPackageDrop{trivfloat}
2 \LWR@origRequirePackage{trivfloat}
```

`\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:** `3 \begin{warpHTML}`

```
4 \AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}
```

```
5 \end{warpHTML}
```

### § 319.1 Combining `\newfloat`, `\trivfloat`, and `algorithmicx`

**for HTML & PRINT:** `6 \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
```

---

```
7 \end{warpall}
```

---

File 233 `lwarp-turnthepage.sty`

### § 320 Package **turnthepage**

Pkg `turnthepage` `turnthepage` is ignored.

**for HTML output:** `1 \LWR@ProvidesPackageDrop{turnthepage}`

---

```
2 \newcommand{\turnthepage}{}

```

---

File 234 **lwarp-typearea.sty**

§ 321 Package **typearea**

*(Emulates or patches code by MARKUS KOHM.)*

Pkg typearea typearea is emulated.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. `\DeclareDocumentCommand` is used to overwrite the koma-script definitions.

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{typearea}

2 \DeclareDocumentCommand{\typearea}{o m}{}
3 \DeclareDocumentCommand{\recalctypearea}{}{}
4 \@ifundefined{footheight}{\newlength{footheight}}{}
5 \DeclareDocumentCommand{\areaset}{o m m}{}
6 \DeclareDocumentCommand{\activateareas}{}{}
7 \DeclareDocumentCommand{\storeareas}{m}{}
8 \DeclareDocumentCommand{\BeforeRestoreareas}{s m}{}
9 \DeclareDocumentCommand{\AfterRestoreareas}{s m}{}
10 \DeclareDocumentCommand{\AfterCalculatingTypearea}{s m}{}
11 \DeclareDocumentCommand{\AfterSettingArea}{s m}{}

```

---

File 235 **lwarp-ulem.sty**

§ 322 Package **ulem**

*(Emulates or patches code by DONALD ARSENEAU.)*

Pkg ulem Emulated.

**for HTML output:** Emulate the original package:

```
1 \ProvidesPackage{lwarp-ulem}

```

Original lwarp definitions:

```
2 \LetLtxMacro\LWR@ulemorigemph\emph
3 \LetLtxMacro\LWR@ulemorigtextbf\textbf

```

Basic markup commands, using CSS:

```
4 \NewDocumentCommand{\uline}{+m}{%
5 \LWR@HTMLtextstyle%
6   {text-decoration:underline;text-decoration-skip}%
7   {uline}{#1}%
8 }
9
10 \NewDocumentCommand{\uuline}{+m}{%
11 \LWR@HTMLtextstyle%
12   {%
13     text-decoration:underline;text-decoration-skip;%
14     text-decoration-style:double%
15   }%
16   {uuline}{#1}%
17 }
18
19 \NewDocumentCommand{\uwave}{+m}{%
20 \LWR@HTMLtextstyle%
21   {%
22     text-decoration:underline;text-decoration-skip;%
23     text-decoration-style:wavy%
24   }%
25   {uwave}{#1}%
26 }
27
28 \NewDocumentCommand{\sout}{+m}{%
29 \LWR@HTMLtextstyle%
30   {text-decoration:line-through}%
31   {sout}{#1}%
32 }
33
34 \NewDocumentCommand{\xout}{+m}{%
35 \LWR@HTMLtextstyle%
36   {text-decoration:line-through}%
37   {xout}{#1}%
38 }
39
40 \NewDocumentCommand{\dashuline}{+m}{%
41 \LWR@HTMLtextstyle%
42   {%
43     text-decoration:underline;%
44     text-decoration-skip;%
45     text-decoration-style:dashed%
46   }%
47   {dashuline}{#1}%
48 }
49
50 \NewDocumentCommand{\dotuline}{+m}{%
51 \LWR@HTMLtextstyle%
```



```

52    {%
53        text-decoration:underline;%
54        text-decoration-skip;%
55        text-decoration-style:dotted%
56    }%
57    {dotuline}{#1}%
58 }

```

Nullified parameters:

```

59 \NewDocumentCommand{\ULthickness}{-}{-}
60 \newlength{\ULdepth}

```

Nullified/emulated macros:

```

61 \NewDocumentCommand{\markoverwith}{m}{-}
62 \NewDocumentCommand{\ULon}{+m}{\uline{#1}\egroup}

```

\useunder only works with \textbf, etc, but not \bfseries, etc.

```

63 \NewDocumentCommand{\useunder}{m m m}{%
64 \relax%
65 \ifx\relax#3\relax\else % argumentative command
66     \def#3{#1}\MakeRobust{#3}\fi
67 }

```

Triggered by package options, also available for the users:

```

68 \newcommand*{\normalem}{\LetLtxMacro\emph\LWR@ulemorigemph}
69 \newcommand*{\ULforem}{\LetLtxMacro\emph\uline}
70 \ULforem% default

```

Package options:

```

71 \DeclareOption{normalem}{\normalem}
72 \DeclareOption{ULforem}{\ULforem}
73 \DeclareOption{normalbf}{-}
74 \DeclareOption{UWforbf}{\useunder{\uwave}{\bf}\textbf}}
75
76 \DeclareOption*{}
77 \ProcessOptions\relax% original LaTeX code

```

---

File 236 **lwarp-upref.sty**

§ 323    Package **upref**

Pkg upref Ignored.

for HTML output: Discard all options for lwarp-upref:

```
1 \LWR@ProvidesPackageDrop{upref}
```

---

File 237 **lwarp-verse.sty**

§ 324    Package **verse**

*(Emulates or patches code by PETER WILSON.)*

Pkg verse verse is supported and patched by lwarp.

for HTML output: Pass all options for lwarp-verse:

```
1 \LWR@ProvidesPackagePass{verse}
```

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


---

```
\begin{warpHTML}
\let\attrib\attribution
\end{warpHTML}
```

---

Len `\leftskip` These lengths are used by verse and memoir to control the left margin, and they may already be set by the user for print output. New lengths `\HTMLvleftskip` and `\HTMLleftmargini` are provided to control the margins in HTML output. These new lengths may be set by the user before any verse environment, and persist until they are manually changed again. One reason to change `\HTMLleftmargini` is if there is a wide `\flagverse` in use, such as the word “Chorus”, in which case the value of

\HTMLleftmargini should be set to a wide enough length to contain “Chorus”. The default is wide enough for a stanza number.

 **spacing** 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 {%
```

Use the original list environment inside a <pre> to attempt to preserve formatting.

```
6 \LWR@restoreoriglists%
```

Pkg **verse** The verse or memoir packages can place stanza numbers to the left with their  
Cls **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**

```
7 \ifdef{\vleftskip}{%
8 \setlength{\vleftskip}{\HTMLvleftskip}
9 \setlength{\leftmargini}{\HTMLleftmargini}
10 }{}
11 \LWR@forcenewpage
12 \LWR@atbeginverbatim{verse}
13 \unskip\LWR@origvspace{-\baselineskip}
14 }
```

After the end of the verse environment, which places the <pre> tag at the regular left margin:

```
15 \AtEndEnvironment{verse}{
16 \LWR@afterendverbatim
17 }
```

Patch to place poemtitle inside an HTML <span> of class poemtitle:

```
18 \ifdef{\poemtitle}{
19 \DeclareDocumentCommand{\@vstypeptitle}{m}{%}
```

---

```

20 \vspace{\beforepoemtitleskip}%
21 {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
22 \vspace{\afterpoemtitleskip}%
23 }
24 }{}
25
26 \LWR@traceinfo{Finished patching verse.}
27 }% AfterEndPreamble

```

---

File 238 **lwarp-vertbars.sty**

§ 325     Package **vertbars**

*(Emulates or patches code by PETER WILSON.)*

Pkg    vertbars    vertbars is emulated.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{vertbars}

2 \newlength{\barwidth}
3 \setlength{\barwidth}{0.4pt}
4 \newlength{\barspace}
5 \setlength{\barspace}{1em}
6
7 \newenvironment{vertbar}{
8   \LWR@forcenewpage
9   \LWR@forceminwidth{\barwidth}
10  \begin{BlockClass}[%
11    border-left: \LWR@printlength{\LWR@atleastonept} solid black ; %
12    padding-left: \LWR@printlength{\barspace}%
13  ]{vertbar}
14 }{
15   \end{BlockClass}
16 }

```

---

File 239 **lwarp-vmargin.sty**

§ 326     Package **vmargin**

Pkg    vmargin    vmargin is ignored.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{vmargin}

```

---

```

2 \newcommand*{\LWRVM@customsize}[2]{}
3 \newcommand*{\setpapersize}[2][\ifstrequal{#2}{custom}{\LWRVM@customsize}{}]{}
4 \newcommand*{\setmargins}[8]{}
5 \newcommand*{\setmarginsrb}[8]{}
6 \newcommand*{\setmargnohf}[4]{}
7 \newcommand*{\setmargnohfrb}[4]{}
8 \newcommand*{\setmarg}[4]{}
9 \newcommand*{\setmargrb}[4]{}
10 \newlength{\PaperWidth}
11 \setlength{\PaperWidth}{8.5in}
12 \newlength{\PaperHeight}
13 \setlength{\PaperHeight}{11in}
14 \newif\ifLandscape

```

---

File 240 **lwarp-vwcol.sty**

§ 327      Package **vwcol**

*(Emulates or patches code by WILL ROBERTSON.)*

Pkg    vwcol    vwcol is patched for use with lwarp.

The width option is ignored. All vwcol environments adjust to 1–3 equal-width columns, depening on the width of the browser window.

The remaining options are supported, except for lines and maxrecursion.

**for HTML output:**    1 \LWR@ProvidesPackagePass{vwcol}

Factored from \vwcol. Each is given a style tag to append to the final style.

```

\LWR@vwcol@addrule    {\style tag}

2 \newcommand*{\LWR@vwcol@addrule}[1]{%
3     \appto{\LWR@vwcolstyle}{%
4         #1: %
5         \LWR@printlength{\vwcol@rule} solid \#\LWR@vwcol@rulecolor ; %
6     }%
7 }

\LWR@vwcol@addrule    {\style tag}

8 \newcommand*{\LWR@vwcol@addgap}[1]{%
9     \appto{\LWR@vwcolstyle}{%
10        #1: %
11        \LWR@printlength{\vwcol@sep} ; %

```

```

12    }%
13 }

```

Env `vwcol` `{\key/values}`

Redefine the environment to add a HTML style. The style is built depending on the required options.

```

14 \renewenvironment*{vwcol}[1][ ]{%

```

New paragraph, and process the options:

```

15 \par\noindent%
16 \vwcolsetup{#1}%

```

Begin with no style:

```

17 \newcommand*{\LWR@vwcolstyle}{}

```

presep and postsep are created with HTML margins:

```

18 \if@vwcol@presep
19   \appto{\LWR@vwcolstyle}{margin-left: 1em ; padding-left: .5em ; }
20 \fi
21 \if@vwcol@postsep
22   \appto{\LWR@vwcolstyle}{margin-right: 1em ; padding-right: .5em ; }
23 \fi

```

sep becomes column-gap:

```

24 \ifdimgreater{\vwcol@sep}{1sp}{
25   \LWR@vwcol@addgap{column-gap}
26   \LWR@vwcol@addgap{-moz-column-gap}
27   \LWR@vwcol@addgap{-webkit-column-gap}
28 }{}

```

rule become column-rule, while prerule and postrule become HTML borders:

```

29 \convertcolorspec{named}{\vwcol@rulecol}{HTML}\LWR@vwcol@rulecolor%
30 \ifdimgreater{\vwcol@rule}{0pt}{
31   \ifdimless{\vwcol@rule}{1pt}{
32     \setlength{\vwcol@rule}{1pt}
33   }{}
34   \LWR@vwcol@addrule{column-rule}
35   \LWR@vwcol@addrule{-moz-column-rule}
36   \LWR@vwcol@addrule{-webkit-column-rule}
37   \if@vwcol@prerule\LWR@vwcol@addrule{border-left}\fi
38   \if@vwcol@postrule\LWR@vwcol@addrule{border-right}\fi
39 }{}

```

Each of the justify options becomes a text-align. Indentation is added where appropriate.

```

40 \ifdefequal{\vwcol@justify}{\RaggedRight}{
41   \appto{\LWR@vwcolstyle}{text-align: left ; }

```

```

42 \ifdimgreater{\vwcol@parindent}{Opt}{
43     \appto{\LWR@vwcolstyle}{%
44         text-indent: \LWR@printlength{\vwcol@parindent} ; %
45     }
46 }{}
47 {}

48 \ifdefequal{\vwcol@justify}{\RaggedLeft}{
49     \appto{\LWR@vwcolstyle}{text-align: right ; }
50 }{}

51 \ifdefequal{\vwcol@justify}{\Centering}{
52     \appto{\LWR@vwcolstyle}{text-align: center ; }
53 }{}

54 \ifdefequal{\vwcol@justify}{\justifying}{
55     \appto{\LWR@vwcolstyle}{text-align: justify ; }
56     \ifdimgreater{\vwcol@parindent}{Opt}{
57         \appto{\LWR@vwcolstyle}{%
58             text-indent: \LWR@printlength{\vwcol@parindent} ; %
59         }
60     }{}
61 }{}

```

Create the <div> with the assembled style:

```

62 \BlockClass[\LWR@vwcolstyle]{multicols}
63 }

```

When the environment ends:

```

64 {
65 \endBlockClass
66 }

```

---

File 241 **lwarp-wallpaper.sty**

§ 328     Package **wallpaper**

*(Emulates or patches code by MICHAEL H.F. WILKINSON.)*

Pkg wallpaper wallpaper is emulated.

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[1]{}
17 \newlength\wpXoffset
18 \newlength\wpYoffset

```

---

File 242 **lwarp-watermark.sty**

§ 329      Package **watermark**

*(Emulates or patches code by ALEXANDER I. ROZHENKO.)*

Pkg    watermark    watermark is emulated.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{watermark}

```

2 \newcommand\watermark[1]{}
3 \newcommand\leftwatermark[1]{}
4 \newcommand\rightwatermark[1]{}
5 \newcommand\thiswatermark[1]{}
6 \newcommand\thispageheading[1]{}

```

---

File 243 **lwarp-wrapfig.sty**

§ 330      Package **wrapfig**

*(Emulates or patches code by DONALD ARSENEAU.)*

Pkg    wrapfig    wrapfig is emulated.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{wrapfig}

```

2 \newcommand*\LWR@wrapposition[1]{}
3
4 \newcommand*\LWR@subwrapfigure[2]{}
5 \renewcommand*\LWR@wrapposition[1]{}
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 \LWR@BlockClassWP{%
14     width:\LWR@printlength{\LWR@templengthone}; \LWR@wrapposition; %
15     margin:10pt%
16}%
17 {%
18     width:\LWR@printlength{\LWR@templengthone}; \LWR@wrapposition%
19}%
20 {marginblock}%
21 }
22
23
24 \NewDocumentEnvironment{wrapfigure}{o m o m}
25 {%
26 \LWR@subwrapfigure{#2}{#4}%
27 \captionsetup{type=figure}%
28 }
29 {%
30 \endLWR@BlockClassWP%
31 }
32
33
34 \NewDocumentEnvironment{wraptable}{o m o m}
35 {%
36 \LWR@subwrapfigure{#2}{#4}%
37 \captionsetup{type=table}%
38 }
39 {%
40 \endLWR@BlockClassWP%
41 }
42
43
44 \NewDocumentEnvironment{wrapfloat}{m o m o m}
45 {%
46 \LWR@subwrapfigure{#3}{#5}%
47 \captionsetup{type=#1}%
48 }
49 {%
50 \endLWR@BlockClassWP%
51 }
52
53 \newlength{\wrapoverhang}

```

---

File 244 **lwarp-xcolor.sty**

§ 331 Package **xcolor**

*(Emulates or patches code by DR. UWE KERN.)*

Pkg xcolor xcolor is supported by lwarp.

### § 331.1 Limitations

`\colorboxBlock` and `\fcolorboxBlock` are provided for increased HTML compatibility, and they are identical to `\colorbox` and `\fcolorbox` in print mode. In HTML mode they place their contents into a `<div>` instead of a `<span>`. These `<div>`s are set to `display: inline-block` so adjacent `\colorboxBlock`s appear side-by-side in HTML, although text is placed before or after each.

Print-mode definitions for `\colorboxBlock` and `\fcolorboxBlock` are created by lwarp's core if xcolor is loaded.

`background: none` `\fcolorbox` and `\fcolorboxBlock` allow a background color of `none`, in which case only the frame is drawn, which can be useful for HTML.

`color support` Color definitions, models, and mixing are fully supported without any changes required.

`tables` `\rowcolors` is supported, except that the optional argument is ignored so far.

`colored text and boxes` `\textcolor`, `\colorbox`, and `\fcolorbox` are supported.

`\color` and `\pagecolor` `\color` and `\pagecolor` are ignored. Use CSS or `\textcolor` where possible.

### § 331.2 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 77. 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 409.

**\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 743.

**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 743.

**HTML lateximage:** Remembers and reuses the print version.

**\colorboxBlock:**

**Print:** Becomes `\colorbox`.

**HTML:** Newly defined by `lwarp-xcolor` to use a `<div>`, page 744.

**HTML lateximage:** Remembers and reuses the print version `\colorbox`.

**\fcolorbox:**

**Print:** Modified to allow a background of none.

`\LWRprint@fcolorbox` at section 77

**HTML:** Redefined by `lwarp-xcolor`, page 744.

**HTML lateximage:** Remembers and reuses the print version.

**\fcolorboxBlock:**

**Print:** Becomes `\fcolorbox`. Section 77

**HTML:** Newly defined by `lwarp-xcolor` to use a `<div>`, page 745.

**HTML lateximage:** Remembers and reuses the print version `\fcolorbox`.

**fcolorminipage:**

**Print:** Newly defined in the `lwarp` core.

`LWRprint@fcolorminipage` at section 77

**HTML:** Newly defined by `lwarp-xcolor`, page 745.

**HTML lateximage:** Uses the print version.

**\boxframe:**

**Print:** Used as-is.

**HTML:** Redefined by `lwarp-xcolor`, page 746.

**HTML lateximage:** Remembers and reuses the print version.

### § 331.3 Package loading

for HTML output:   1 \LWR@ProvidesPackagePass{xcolor}

                      2 \begin{warpHTML}

### § 331.4 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 }
```

### § 331.5 HTML color style

`\LWR@tempcolor` The color converted to HTML colorspace.

```
19 \newcommand*{\LWR@tempcolor}{}
20 \newcommand*{\LWR@tempcolortwo}{}%
```

Sets `\LWR@tempcolor` to the current color.

`\LWR@findcurrenttextcolor`

```
21 \newcommand*{\LWR@findcurrenttextcolor}{%
22 \protect\colorlet{\LWR@current@color}{.}%
23 \protect\convertcolourspec{named}{\LWR@current@color}{HTML}\LWR@tempcolor%
24 }
```

Prints a color style for the current color.

`\LWR@currenttextcolorstyle`

```
25 \newcommand*{\LWR@currenttextcolorstyle}{%
26 \LWR@findcurrenttextcolor%
27 \ifdefstring{\LWR@tempcolor}{000000}%
28 {}%
29 {color: \#\LWR@tempcolor ; }%
30 }
```

`\LWR@textcurrentcolor` `{<text>}` Like `\textcolor` but uses the current `\color` instead.

```

31 \newcommand*{\LWR@textcurrentcolor}[1]{%
32 \begingroup%
33 \LWR@FBcancel%
34 \LWR@findcurrenttextcolor%
35 \InlineClass[color:\#\LWR@tempcolor]{textcolor}{%
36     \renewcommand*{\LWR@currenttextcolor}{\#\LWR@tempcolor}%
37     #1%
38 }%
39 \endgroup%
40 }

```

`\LWR@colorstyle`    `{\langle 2: model \rangle}{\langle 3: color \rangle}`

For a color style, prints the color converted to HTML colors.

```

41 \NewDocumentCommand{\LWR@colorstyle}{m m}{%
42 \begingroup%
43 \LWR@FBcancel%

```

Use the xcolor package to convert to an HTML color space:

```

44 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%

```

Print the converted color:

```

45 \#\LWR@tempcolor%
46 \endgroup%
47 }

```

`\LWR@backgroundcolor`    `[\langle model \rangle]{\langle color \rangle}{\langle text \rangle}`

Similar to `\textcolor`, but prints black text against a color background.

Converted into an HTML hex color span.

```

48 \NewDocumentCommand{\LWR@backgroundcolor}{O{named} m m}{%
49 \begingroup%
50 \LWR@FBcancel%
51 \InlineClass[background:\LWR@colorstyle{#1}{#2}]{backgroundcolor}{%
52 #3%
53 }%
54 \endgroup%
55 }

```

### § 331.6 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.

```
56 \newcommand*\LWR@borderpadding[2]{%
57 border:\LWR@printlength{\LWR@atleastonept} solid \LWR@colorstyle{#1}{#2} ; %
58 padding:\LWR@printlength{\fboxsep}%
59 }
```

### § 331.7 High-level macros

`\color` `\color` appears in the  $\TeX$  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.

```
60 \RenewDocumentCommand{\textcolor}{0{named} m m}{%
61 \begingroup%
62 \LWR@FBcancel%
63 \InlineClass[color:\LWR@colorstyle{#1}{#2}]{textcolor}{%
64 \renewcommand*\LWR@currenttextcolor{\#\LWR@tempcolor}%
65 #3%
66 }%
67 \endgroup%
68 }
```

`\pagecolor` `[\model] \color`

Ignored. Use css instead.

```
69 \renewcommand*\pagecolor[2][named]{}
```

`\nopagecolor` Ignored.

```
70 \renewcommand*\nopagecolor{}
```

`\colorbox` `[\model] \color \text`

Converted into an HTML hex background color `<span>`.

```

71 \RenewDocumentCommand{\colorbox}{0{named} m +m}{%
72 \begingroup%
73 \LWR@FBcancel%
74 \InlineClass[%
75 background:\LWR@colorstyle{#1}{#2} ; %
76 padding:\LWR@printlength{\fboxsep}%
77 ]{\colorbox}{#3}%
78 \endgroup%
79 }

```

`\colorboxBlock` [*⟨model⟩*] {*⟨color⟩*} {*⟨text⟩*}

Converted into an HTML hex background color <div>.

```

80 \NewDocumentCommand{\colorboxBlock}{0{named} m +m}{%
81 \begingroup%
82 \LWR@FBcancel%
83 \begin{BlockClass}[%
84 background:\LWR@colorstyle{#1}{#2} ; %
85 padding:\LWR@printlength{\fboxsep}%
86 ]{\colorboxBlock}
87 #3
88 \end{BlockClass}%
89 \endgroup%
90 }

```

`\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.

```

91 \RenewDocumentCommand{\fcolorbox}{0{named} m 0{named} m +m}{%
92 \LWR@traceinfo{HTML fcolorbox #2 #4}%
93 \begingroup%
94 \LWR@FBcancel%
95 \LWR@forceminwidth{\fboxrule}%
96 \ifthenelse{\equal{#4}{none}}{%
97 {% no background color
98   \InlineClass[%
99   \LWR@borderpadding{#1}{#2}%
100 ]{\fcolorbox}{#5}%
101 }%
102 {% yes background color
103   \InlineClass[%
104   \LWR@borderpadding{#1}{#2} ; %
105   background:\LWR@colorstyle{#3}{#4}%

```



```

106    ]{fcolorbox}{#5}%
107 }%
108 \endgroup%
109 }

```

`\fcolorboxBlock` [*framemodel*] [*framecolor*] [*boxmodel*] [*boxcolor*] [*text*]

Converted into a framed HTML hex background color span.

A background color of none creates a colored frame without a background color.

```

110 \NewDocumentCommand{\fcolorboxBlock}{0{named} m 0{named} m +m}{%
111 \LWR@traceinfo{HTML fcolorboxBlock #2 #4}%
112 \begingroup%
113 \LWR@FBcancel%
114 \LWR@forceminwidth{\fboxrule}%
115 \ifthenelse{\equal{#4}{none}}{%
116 {% no background color
117   \begin{BlockClass}[%
118     \LWR@borderpadding{#1}{#2}%
119   ]{fcolorboxBlock}
120   #5
121   \end{BlockClass}%
122 }%
123 {% yes background color
124   \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
125   \begin{BlockClass}[%
126     background:\#\LWR@tempcolortwo\ ; %
127     \LWR@borderpadding{#1}{#2}%
128   ]{fcolorboxBlock}
129   #5
130   \end{BlockClass}%
131 }%
132 \endgroup%
133 \LWR@traceinfo{HTML fcolorboxBlock done}%
134 }

```

Creates a framed HTML <div> around its contents.

A print-output version is defined in the lwarp core: section [77](#)

`\LWR@subfcolorminipage` [*framemodel*] [*framecolor*] [*background tag*] [*height*]

```

135 \NewDocumentCommand{\LWR@subfcolorminipage}{m m m m}{%
136 \begin{BlockClass}[%
137 #3%
138 \LWR@borderpadding{#1}{#2} ; %

```

```

139 \IfValueT{#4}{height:\LWR@printlength{\LWR@tempheight} ; }%
140 width:\LWR@printlength{\LWR@tempwidth}%
141 ]{fcolorminipage}%
142 }

```

Env fcolorminipage [*⟨1:frame model⟩*] [*⟨2:frame color⟩*] [*⟨3:box model⟩*] [*⟨4:box color⟩*] [*⟨5:align⟩*] [*⟨6:height⟩*] [*⟨7:inner-align⟩*] [*⟨8:width⟩*]

```

143 \NewDocumentEnvironment{fcolorminipage}{0{named} m 0{named} m 0{c} o o m}
144 {%
145 \LWR@FBcancel%
146 \setlength{\LWR@tempwidth}{#8}%
147 \IfValueT{#6}{\setlength{\LWR@tempheight}{#6}}%
148 \LWR@forceminwidth{\fboxrule}%
149 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
150 \ifthenelse{\equal{#4}{none}}%
151 {\LWR@subfcolorminipage{#1}{#2}{#6}}%
152 {%
153   \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
154   \LWR@subfcolorminipage{#1}{#2}{background:\#\LWR@tempcolortwo\ ;}{#6}%
155 }%
156 }
157 {\end{BlockClass}}

```

`\boxframe` [*⟨width⟩*] [*⟨height⟩*] [*⟨depth⟩*]

The depth is added to the height, but the box is not decended below by the depth.  
`\textcolor` is honored.

```

158 \renewcommand*{\boxframe}[3]{%
159 {%
160 \setlength{\LWR@tempwidth}{#1}%
161 \setlength{\LWR@tempheight}{#2}%
162 \addtolength{\LWR@tempheight}{#3}%
163 \LWR@forceminwidth{\fboxrule}%
164 \InlineClass[%
165 display:inline-block ; %
166 border:\LWR@printlength{\LWR@atleastonept} solid \LWR@currenttextcolor{} ; %
167 width:\LWR@printlength{\LWR@tempwidth} ; %
168 height:\LWR@printlength{\LWR@tempheight}%
169 ]{boxframe}{}%
170 }%
171 }

```

### § 331.8 Row colors

```

\rowc@l@rs    [{\cmds}] {\startrow} {\odd color} {\even color}

172 \newcommand*{\LWR@xcolortempcolor}{}
173
174 \def\rowc@l@rs[#1]#2#3#4%
175 {
176 \global\rownum=1
177 \global\@rowcolorstrue
178 \@ifxempty{#3}%
179   {\def\@oddrowcolor{\@norowcolor}}%
180   {%
181     \convertcolorspec{named}{#3}{HTML}\LWR@xcolortempcolor%
182     \edef\@oddrowcolor{%
183       \csdef{\LWR@xcolorrowHTMLcolor}{\LWR@xcolortempcolor}%
184     }%
185   }%
186 \@ifxempty{#4}%
187   {\def\@evenrowcolor{\@norowcolor}}%
188   {%
189     \convertcolorspec{named}{#4}{HTML}\LWR@xcolortempcolor%
190     \edef\@evenrowcolor{%
191       \csdef{\LWR@xcolorrowHTMLcolor}{\LWR@xcolortempcolor}%
192     }%
193   }%
194 \if@rowcmd
195   \def\@rowcolors
196   {%
197 %     #1%
198     \if@rowcolors
199 %       \noalign{%
200 %         \relax\ifnum\rownum<#2\@norowcolor\else
201 %         \ifodd\rownum\@oddrowcolor\else\@evenrowcolor\fi\fi%
202 %       }%
203     \fi%
204   }%
205 \else
206   \def\@rowcolors
207   {%
208     \if@rowcolors
209 %       \ifnum\rownum<#2%
210 %       \noalign{%
211 %         \@norowcolor
212 %       }
213 %     \else
214 %     #1%
215 %     \noalign{%
216 %       \ifodd\rownum\@oddrowcolor\else\@evenrowcolor\fi%
217 %     }%

```

```

218          \fi
219          \fi%
220      }%
221      \fi
222      \ignorespaces%
223 }

```

`\@norowcolor` Turns off color for this row.

```

224 \def\@norowcolor{%
225 \renewcommand{\LWR@xcolorrowHTMLcolor}{}}%
226 }

```

`\@rowc@lors` Executed at the end of each row.

```

227 \def\@rowc@lors{%
228 % \noalign{%
229 % \global\advance\rownum\@ne%
230 % }%
231 % \@rowcolors%
232 }

```

```

233 \end{warpHTML}

```

---

File 245 **lwarp-xfrac.sty**

§ 332 Package **xfrac**

*(Emulates or patches code by THE LATEX3 PROJECT.)*

Pkg **xfrac** Supported by adding xfrac instances.

for HTML output: 1 \LWR@ProvidesPackagePass{xfrac}

 **font size**

In the user's document preamble, lwarp should be loaded after font-related setup. During HTML conversion, this font is used by lwarp to generate its initial PDF output containing HTML tags, later to be converted by pdftotext to a plain text file. While the text may be in any font which pdftotext can read, the math is directly converted into SVG images using this same user-selected font. xfrac below is set for the Latin Modern (lmr) font. If another font is used, it may be desirable to redefine `\xfracHTMLfontsize` with a different em size.

`\sfrac` [*instance*] {*num*} [*sep*] {*denom*}

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. `\scalebox` is

made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a `lateximage`, no adjustments are necessary.

**for HTML & PRINT:**    `2 \begin{warpall}`

`\xfracHTMLfontsize`    User-redefinable macro which controls the font size of the fraction.

```
3 \newcommand*{\xfracHTMLfontsize}{.6em}
```

```
4 \end{warpall}
```

**for HTML output:**    `5 \begin{warpHTML}`

**font size**    A span for a small font, used in the numerator and denominator:

```
6 \newcommand*{\LWR@htmlsmallfontstart}{%
7 \LWR@htmltagc{span style="font-size:\xfracHTMLfontsize"}%
8 \LWR@nestspan%
9 %
10 }
11
12 \newcommand*{\LWR@htmlsmallfontend}{%
13 \LWR@htmltagc{/span}%
14 \endLWR@nestspan%
15 }
```

`\scalebox`    A nullified `\scalebox` command, to avoid introducing HTML scaling tags:

```
16 \NewDocumentCommand{\LWR@noscalebox}{m o m}{#3}
```

**instances**    Instances of `xfrac` for various font choices:

Produce HTML tags for a small superscript numerator and a small (non-subscript) denominator.

Scaling is turned off so that `pdftotext` correctly reads the result.

```
17 \DeclareInstance{xfrac}{default}{text}{
18 numerator-format = {%
19 \begingroup%
20 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
21 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend%
22 \endgroup%
23 },
24 denominator-format = {%
25 \begingroup%
```

```

26 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
27 \LWR@htmlsmallfontstart{}\, #1\LWR@htmlsmallfontend%
28 \endgroup%
29 },

```

For pdftotext, do not scale the text:

```

30 scaling = false
31 }
32
33 \DeclareInstance{xfrac}{lmr}{text}{
34 numerator-format = {%
35 \begingroup%
36 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
37 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend%
38 \endgroup%
39 },
40 denominator-format = {%
41 \begingroup%
42 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
43 \LWR@htmlsmallfontstart{}\, #1\LWR@htmlsmallfontend%
44 \endgroup%
45 },

```

For pdftotext, do not scale the text:

```

46 scaling = false
47 }
48
49 \DeclareInstance{xfrac}{lms}{text}{
50 numerator-format = {%
51 \begingroup%
52 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
53 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend%
54 \endgroup%
55 },
56 denominator-format = {%
57 \begingroup%
58 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
59 \LWR@htmlsmallfontstart{}\, #1\LWR@htmlsmallfontend%
60 \endgroup%
61 },

```

For pdftotext, do not scale the text:

```

62 scaling = false
63 }
64
65 \DeclareInstance{xfrac}{lmtt}{text}{

```

```

66 numerator-format = {%
67 \begingroup%
68 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
69 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend%
70 \endgroup%
71 },
72 denominator-format = {%
73 \begingroup%
74 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
75 \LWR@htmlsmallfontstart{\,\,#1\LWR@htmlsmallfontend%
76 \endgroup%
77 },

```

For pdftotext, do not scale the text:

```

78 scaling = false
79 }

```

```

80 \end{warpHTML}

```

---

File 246 **lwarp-xltxtra.sty**

§ 333      Package **xltxtra**

*(Emulates or patches code by WILL ROBERTSON, JONATHAN KEW.)*

Pkg    xltxtra    xltxtra is emulated.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{xltxtra}

2 \RequirePackage{realscripts}
3 \RequirePackage{metalogo}
4 \newcommand*\TeX@logo@spacing[6]{%
5
6 \newcommand*\vfrac[2]{%
7 \textsuperscript{#1}/\textsubscript{#2}%
8 }
9
10 \newcommand\namedglyph[1]{%
11 \@tempcnta=\XeTeXglyphindex " #1"\relax
12 \ifnum\@tempcnta>0
13 \XeTeXglyph\@tempcnta
14 \else
15 \xxt@namedglyph@fallback{#1}%
16 \fi}

```

---

```

17
18 \newcommand\xxt@namedglyph@fallback[1]{[#1]}
19
20 \DeclareDocumentCommand{\showhyphens}{m}{}
```

---

File 247 **lwarp-xmpincl.sty**

§ 334     Package **xmpincl**

*(Emulates or patches code by MAARTEN SNEEP.)*

Pkg    xmpincl    Emulated.

**for HTML output:**    Discard all options for lwarp-xmpincl:

```

1 \LWR@ProvidesPackageDrop{xmpincl}

2 \newcommand*{\includexmp}[1]{}
```

---

File 248 **lwarp-xtab.sty**

§ 335     Package **xtab**

*(Emulates or patches code by PETER WILSON.)*

Pkg    xtab    xtab is emulated.

**for HTML output:**    1 \LWR@ProvidesPackageDrop{xtab}

⚠ misplaced alignment    For \tablefirsthead, etc., enclose them as follows:  
alignment tab character &

```

\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars
```

See section 8.9.

⚠ **lateximage**    supertabular and xtab are not supported inside a lateximage.

```

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 \newcommand*{\LWRXT@caption}{}
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 249 **lwarp-xurl.sty**

§ 336 Package **xurl**

Pkg xurl xurl is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{xurl}


---

File 250 **lwarp-xy.sty**

§ 337 Package **xy**

*(Emulates or patches code by KRISTOFFER H. ROSE, ROSS MOORE.)*

Pkg xy xy is patched for use by lwarp.

 \xypolygon must be used inside the xy environment, or inside \xy ... \endxy.

for HTML output: 1 \LWR@ProvidesPackagePass{xy}

```

2 \AtBeginDocument{
3
4 \pretol{\xy}{\begin{lateximage}[xy image]}
5 \apptol{\endxy}{\end{lateximage}}
6
7 \@ifundefined{xymatrix}{}{
8 \LetLtxMacro\LWR@origxymatrix\xymatrix
9
10 \renewcommand{\xymatrix}[1]{%
11 \begin{lateximage}[xymatrix]
12 \LWR@origxymatrix{#1}
13 \end{lateximage}
14 }
15 }
16
17 \@ifundefined{xygraph}{}{
18 \LetLtxMacro\LWR@origxygraph\xygraph

```

---

```

19
20 \renewcommand{\xygraph}[1]{%
21 \begin{lateximage}[xygraph]
22 \LWR@origxygraph{#1}
23 \end{lateximage}
24 }
25 }
26
27 }

```

---

File 251 **lwarp-zwpagelayout.sty**

§ 338      Package **zwpagelayout**

*(Emulates or patches code by ZDENĚK WAGNER.)*

Pkg    zwpagelayout    zwpagelayout is ignored.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{zwpagelayout}

2 \def\noBboxes{}
3 \@onlypreamble\noBboxes
4
5 \expandafter\ifx\cename definecolor\endcsname\relax \else
6   \definecolor{cmykblack}{cmyk}{0,0,0,1}
7   \definecolor{grblack}{gray}{0}
8%   \ifzwpl@redefineblack
9%     \definecolor{black}{cmyk}{0,0,0,1}\color{black}
10%   \fi
11   \definecolor{cmykred}{cmyk}{0,1,1,0}
12   \definecolor{cmykgreen}{cmyk}{1,0,1,0}
13   \definecolor{cmykblue}{cmyk}{1,1,0,0}
14   \definecolor{rgbred}{rgb}{1,0,0}
15   \definecolor{rgbgreen}{rgb}{0,1,0}
16   \definecolor{rgbbblue}{rgb}{0,0,1}
17%   \ifzwpl@redefinetcmyk
18%     \definecolor{red}{cmyk}{0,1,1,0}
19%     \definecolor{green}{cmyk}{1,0,1,0}
20%     \definecolor{blue}{cmyk}{1,1,0,0}
21%   \fi
22 \fi
23
24 \let\OverprintXeTeXExtGState\relax
25
26 \DeclareRobustCommand\SetOverprint{\ignorespaces}
27 \DeclareRobustCommand\SetKnockout{\ignorespaces}
28 \DeclareRobustCommand\textoverprint[1]{\SetOverprint#1}

```

---

```

29 \DeclareRobustCommand\textknockout[1]{\SetKnockout#1}}
30
31 \def\SetPDFminorversion#1{}
32 \@onlypreamble\SetPDFminorversion
33
34 \newcommand*\Vcorr{}
35
36 \DeclareRobustCommand\vb[1][]{\SetKnockout#1}}
37 \NewDocumentCommand{\NewOddPage}{* o}{}
38 \NewDocumentCommand{\NewEvenPage}{* o}{}
39 \def\SetOddPageMessage#1{\gdef\ZW@oddwarning#1}
40 \def\SetEvenPageMessage#1{\gdef\ZW@evenwarning#1}
41 \def\ZW@oddwarning{Empty page inserted}\let\ZW@evenwarning\ZW@oddwarning
42
43 \def\clap#1{#1}
44
45 \def\CropFlap{2in}
46 \def\CropSpine{1in}
47 \def\CropXSpine{1in}
48 \def\CropXtrim{.25in}
49 \def\CropYtrim{.25in}
50 \def\UserWidth{5in}
51 \def\UserLeftMargin{1in}
52 \def\UserRightMargin{1in}
53 \def\UserTopMargin{1in}
54 \def\UserBotMargin{1in}
55 \def\thePageNumber{\#\,\arabic{page}}
56 \ifXeTeX
57 \def\ifcaseZWdriver{\ifcase2}
58 \else
59 \def\ifcaseZWdriver{\ifcase1}
60 \fi
61 \DeclareRobustCommand\ZWifdriver[2]{}

```

---

File 252 **lwarp-patch-komascript.sty**

§ 339     Package **patch-komascript**

Pkg    lwarp-patch-komascript    Patches for komascript classes.

lwarp loads this package when scrbook, scrartcl, or screprt classes are detected.

Many features are ignored during the HTML conversion. The goal is source-level compatibility.

\titlehead, \subject, \captionformat, \figureformat, and \tableformat are not yet emulated.

⚠ Not fully tested! [Please send bug reports!](#)

Some features have not yet been tested. Please contact the author with any bug reports.

**for HTML output:** `1 \ProvidesPackage{lwarp-patch-komascript}`

typearea is emulated.

`2 \RequirePackage{lwarp-typearea}`

tocbasic is emulated.

`3 \RequirePackage{lwarp-tocbasic}`

scrextend patches most of the new macros.

`4 \RequirePackage{lwarp-scrextend}`

The `\minisec` is placed inside a `<div>` of class `minisec`.

```
5 \renewcommand*{\minisec}[1]{
6 \begin{BlockClass}{minisec}
7 #1
8 \end{BlockClass}
9 }
```

The part and chapter preambles are placed as plain text just after each heading.

```
10 \@ifundefined{setpartpreamble}{}{
11 \RenewDocumentCommand{\setpartpreamble}{o o +m}{%
12 \renewcommand{\part@preamble}{#3}%
13 }
14 }
15
16 \@ifundefined{setchapterpreamble}{}{
17 \RenewDocumentCommand{\setchapterpreamble}{o o +m}{%
18 \renewcommand{\chapter@preamble}{#3}%
19 }
20 }
```

Simple captions are used in all cases.

```
21 \LetLtxMacro\captionbelow\caption
22 \LetLtxMacro\captionabove\caption
23
24 \LetLtxMacro\captionofbelow\captionof
25 \LetLtxMacro\captionofabove\captionof
```

---

```

26
27 \RenewDocumentEnvironment{captionbeside}{o m o o o s}
28 {}
29 {%
30 \IfValueTF{#1}%
31 {\caption[#1]{#2}}%
32 {\caption{#2}}%
33 }
34
35 \RenewDocumentEnvironment{captionofbeside}{m o m o o o s}
36 {}
37 {%
38 \IfValueTF{#2}%
39 {\captionof{#1}[#2]{#3}}%
40 {\captionof{#1}{#3}}%
41 }
42
43 \RenewDocumentCommand{\setcapindent}{s m}{}
44 \renewcommand*\setcaphanging{}
45 \renewcommand*\setcapwidth[2][{}]{ }
46 \renewcommand*\setcapdynwidth[2][{}]{ }
47 \RenewDocumentCommand{\setcapmargin}{s o m}{}

```

---

File 253 **lwarp-patch-memoir.sty**


§ 340      Package **patch-memoir**

*(Emulates or patches code by PETER WILSON.)*

Pkg lwarp-patch-memoir Patches for memoir class.

 **Not fully tested!** [Please send bug reports!](#)

lwarp loads this package when the memoir class is detected.

 **options clash** While emulating memoir, lwarp pre-loads a number of packages (section 340.1). This can cause an options clash when the user's document later loads the same packages with options. To fix this problem, specify the options before loading lwarp:

```

\documentclass{memoir}
...
\PassOptionsToPackage{options_list}{package_name}
...
\usepackage{lwarp}
...
\usepackage{package_name}

```

\verbfootnote is not supported.

\newfootnoteseries, etc. are not supported.

lwarp loads pagenote to perform memoir's pagenote functions, but there are minor differences in \pagenotesubhead and related macros.

Poem numbering is not supported.

The verbatim environment does not yet support the memoir enhancements. It is currently recommended to load and use fancyvrb instead.

The memoir glossary system is not yet supported by lwarpmk. The glossaries package may be used instead, but does require the glossary entries be changed from the memoir syntax to the glossaries syntax.

for HTML output:    1 \ProvidesPackage{lwarp-patch-memoir}

## § 340.1 Packages

These are pre-loaded to provide emulation for many of memoir's functions. memoir pretends that abstract, etc. are already loaded, via its “emulated” package mechanism, but lwarp is directly loading the “lwarp-” version of each, which happens to avoid memoir's emulation system.

```

2 \RequirePackage{lwarp-abstract}% req'd
3 \RequirePackage{lwarp-array}% req'd
4 \RequirePackage{lwarp-booktabs}% req'd
5 % \RequirePackage{lwarp-ccaption}% to do
6 \RequirePackage{lwarp-changepage}% req'd
7 \RequirePackage{lwarp-crop}
8 \RequirePackage{lwarp-dcolumn}% req'd
9 \RequirePackage{lwarp-enumerate}% req'd
10 \RequirePackage{lwarp-epigraph}% req'd
11 \RequirePackage{lwarp-fancyvrb}% req'd
12 \RequirePackage{lwarp-footmisc}% req'd
13 \RequirePackage{lwarp-framed}% req'd
14 \RequirePackage{lwarp-hanging}% req'd
15 \DisemulatePackage{moreverb}
16 \RequirePackage{lwarp-moreverb}
17 \RequirePackage{lwarp-mparhack}
18 \RequirePackage{lwarp-needspace}% req'd
19 \RequirePackage{lwarp-nextpage}% req'd
20 \RequirePackage{lwarp-pagenote}% req'd
21 \RequirePackage{lwarp-parskip}
22 \RequirePackage{lwarp-setspace}% req'd
23 \RequirePackage{lwarp-showidx}
24 \RequirePackage{lwarp-subfigure}% red'q

```

subfigure is emulated via subfig, which pre-defines subfigure and subtable, but memoir does not, so they must be tested for here:

```

25 \LetLtxMacro\LWR@memorignewsubfloat\nnewsbfloat
26 \RenewDocumentCommand{\newsbfloat}{0{} m}{%
27   \ifundefined{c@sub#2}{%
28     \LWR@memorignewsubfloat[#1]{#2}%
29   }{}%
30 }
31
32 \RequirePackage{lwarp-tabularx}% req'd
33 \RequirePackage{lwarp-titling}% req'd
34 % \RequirePackage{lwarp-tocbibind}% not emulated by memoir
35 \RequirePackage{lwarp-tocloft}% req'd
36 \RequirePackage{lwarp-verse}% req'd

```

## § 340.2 Preliminary setup

Bypass the memoir package mechanism:

```

37 \LetLtxMacro\LWR@origlabel\@mem@old@label

```

Redefined to write the LWR@autoindex counter instead of page

```

38 \AtBeginDocument{
39   \def\@wrindexhyp#1||\{\%
40     \addtocounter{LWR@autoindex}{1}%
41     \LWR@newlabel{LWRindex-\arabic{LWR@autoindex}}%
42     \ifshowindexmark\@showidx{#1}\fi
43     \protected@write\@auxout{%
44       {\string\@wrindexm@m{\@idxfile}{#1}{\arabic{LWR@autoindex}}}%
45     \endgroup
46     \@esphack}%
47 }

```

memoir already set the page size to a default, so it must be forced large for lwarp's use, to avoid tag overflows off the page.

```

48 \setstocksize{190in}{20in}
49 \setlrmarginsandblock{2in}{2in}{*}
50 \setulmarginsandblock{1in}{1in}{*}

```



### § 340.3 Laying out the page

```

51 \renewcommand*{\stockavi}{}
52 \renewcommand*{\stockav}{}
53 \renewcommand*{\stockaiv}{}
54 \renewcommand*{\stockaiii}{}
55 \renewcommand*{\stockbvi}{}
56 \renewcommand*{\stockbv}{}
57 \renewcommand*{\stockbiv}{}
58 \renewcommand*{\stockbiii}{}
59 % \renewcommand*{\stockmetriccrownvo}{}% in docs but not in the package
60 \renewcommand*{\stockmlargecrownvo}{}
61 \renewcommand*{\stockmdemyvo}{}
62 \renewcommand*{\stockmsmallroyalvo}{}
63 \renewcommand*{\pageavi}{}
64 \renewcommand*{\pageav}{}
65 \renewcommand*{\pageaiv}{}
66 \renewcommand*{\pageaiii}{}
67 \renewcommand*{\pagebvi}{}
68 \renewcommand*{\pagebv}{}
69 \renewcommand*{\pagebiv}{}
70 \renewcommand*{\pagebiii}{}
71 % \renewcommand*{\pagemetriccrownvo}{}% in docs but not in the package
72 \renewcommand*{\pagemlargecrownvo}{}
73 \renewcommand*{\pagemdemyvo}{}
74 \renewcommand*{\pagemsmallroyalvo}{}
75
76 \renewcommand*{\stockdbill}{}
77 \renewcommand*{\stockstatement}{}
78 \renewcommand*{\stockexecutive}{}
79 \renewcommand*{\stockletter}{}
80 \renewcommand*{\stockold}{}
81 \renewcommand*{\stocklegal}{}
82 \renewcommand*{\stockledger}{}
83 \renewcommand*{\stockbroadsheet}{}
84 \renewcommand*{\pagedbill}{}
85 \renewcommand*{\pagestatement}{}
86 \renewcommand*{\pageexecutive}{}
87 \renewcommand*{\pageletter}{}
88 \renewcommand*{\pageold}{}
89 \renewcommand*{\pagelegal}{}
90 \renewcommand*{\pageledger}{}
91 \renewcommand*{\pagebroadsheet}{}
92
93 \renewcommand*{\stockpottvo}{}
94 \renewcommand*{\stockfoolscapvo}{}
95 \renewcommand*{\stockcrownvo}{}
96 \renewcommand*{\stockpostvo}{}
97 \renewcommand*{\stocklargecrownvo}{}
98 \renewcommand*{\stocklargepostvo}{}

```

```
99 \renewcommand*{\stocksmallldemyvo}{}
100 \renewcommand*{\stockdemyvo}{}
101 \renewcommand*{\stockmediumvo}{}
102 \renewcommand*{\stocksmallroyalvo}{}
103 \renewcommand*{\stockroyalvo}{}
104 \renewcommand*{\stocksuperroyalvo}{}
105 \renewcommand*{\stockimperialvo}{}
106 \renewcommand*{\pagepottvo}{}
107 \renewcommand*{\pagefoolscapvo}{}
108 \renewcommand*{\pagecrownvo}{}
109 \renewcommand*{\pagepostvo}{}
110 \renewcommand*{\pagelargecrownvo}{}
111 \renewcommand*{\pagelargepostvo}{}
112 \renewcommand*{\pagesmallldemyvo}{}
113 \renewcommand*{\pagedemyvo}{}
114 \renewcommand*{\pagemediumvo}{}
115 \renewcommand*{\pagesmallroyalvo}{}
116 \renewcommand*{\pageroyalvo}{}
117 \renewcommand*{\pagesuperroyalvo}{}
118 \renewcommand*{\pageimperialvo}{}
119
120 \renewcommand*{\memfontfamily}{}
121 \renewcommand*{\memfontenc}{}
122 \renewcommand*{\memfontpack}{}
123
124 \renewcommand*{\anyptfilebase}{}
125 \renewcommand*{\anyptsizesize}{10}
126
127 \renewcommand*{\setstocksize}[2]{}
128 \renewcommand*{\settrimmedsize}[3]{}
129 \renewcommand*{\settrims}[2]{}
130
131 % \newlength{\lxvchars}
132 % \setlength{\lxvchars}{305pt}
133 % \newlength{\xlvchars}
134 % \setlength{\xlvchars}{190pt}
135 \renewcommand*{\setxlvchars}[1]{}
136 \renewcommand*{\setlxvchars}[1]{}
137
138 \renewcommand*{\settypeblocksize}[3]{}
139 \renewcommand*{\setlrmargins}[3]{}
140 \renewcommand*{\setlrmarginsandblock}[3]{}
141 \renewcommand*{\setbinding}[1]{}
142 \renewcommand*{\setulmargins}[3]{}
143 \renewcommand*{\setulmarginsandblock}[3]{}
144 \renewcommand*{\setcolsepandrule}[2]{}
145
146 \renewcommand*{\setheadfoot}[2]{}
147 \renewcommand*{\setheaderspaces}[3]{}
148 \renewcommand*{\setmarginnotes}[3]{}

```

```

149 \renewcommand*{\setfootins}[2]{}
150 \renewcommand*{\checkandfixthelayout}[1] [] {}
151 \renewcommand*{\checkthelayout}[1] {}
152 \renewcommand*{\fixthelayout}{}
153 %
154 % \newlength{\stockheight}
155 % \newlength{\trimtop}
156 % \newlength{\trimedge}
157 % \newlength{\stockwidth}
158 % \newlength{\spinemargin}
159 % \newlength{\foremargin}
160 % \newlength{\uppermargin}
161 % \newlength{\headmargin}
162 %
163 \renewcommand*{\typeoutlayout}{}
164 \renewcommand*{\typeoutstandardlayout}{}
165 \renewcommand*{\settypeoutlayoutunit}[1] {}
166 \renewcommand*{\fixpdflayout}{}
167 \renewcommand*{\fixdvipslayout}{}
168
169 \renewcommand*{\medievalpage}[1] [] {}
170 \renewcommand*{\isopage}[1] [] {}
171 \renewcommand*{\semiisopage}[1] [] {}
172
173 \renewcommand{\setpagebl}[3] {}
174 \renewcommand{\setpageml}[3] {}
175 \renewcommand{\setpagetl}[3] {}
176 \renewcommand{\setpagetm}[3] {}
177 \renewcommand{\setpagetr}[3] {}
178 \renewcommand{\setpagemr}[3] {}
179 \renewcommand{\setpagebr}[3] {}
180 \renewcommand{\setpagebm}[3] {}
181 \renewcommand{\setpagecc}[3] {}

```

#### § 340.4 Text and fonts

```

182 \let\miniscule\tiny
183 \let\HUGE\Huge
184
185 \renewcommand*{\abnormalparskip}[1] {}
186 \renewcommand*{\nonzeroparskip}{}
187 \renewcommand*{\traditionalparskip}{}
188
189 \let\onelineskip\baselineskip
190
191 \let\OnehalfSpacing\onehalfspacing
192 \let\DoubleSpacing\doublespacing
193 \renewcommand*{\setPagenoteSpacing}[1] {}
194 \renewcommand*{\setFloatSpacing}[1] {}

```

```

195 \let\SingleSpacing\singlespacing
196 \let\setSingleSpace\SetSinglespace
197 \let\SingleSpace\singlespace
198 \let\endSingleSpace\endsinglespace
199 \let\Spacing\spacing
200 \let\endSpacing\endspacing
201 \let\OnehalfSpace\onehalfspace
202 \let\endOnehalfSpace\endonehalfspace
203 \csletcs{OnehalfSpace*}{onehalfspace}
204 \csletcs{endOnehalfSpace*}{endonehalfspace}
205 \let\DoubleSpace\doublespace
206 \let\endDoubleSpace\enddoublespace
207 \csletcs{DoubleSpace*}{doublespace}
208 \csletcs{endDoubleSpace*}{enddoublespace}
209 \renewcommand*{\setDisplayskipStretch}[1]{
210 \renewcommand*{\memdskipstretch}{
211 \renewcommand*{\noDisplayskipStretch}{
212 \renewcommand*{\memdskips}{
213
214 \renewcommand*{\midsloppy}{
215 \renewenvironment*{midsloppypar}{
216
217 \renewcommand*{\sloppybottom}{

```

### § 340.5 Titles

```

218 \csletcs{titlingpage*}{titlingpage}
219 \csletcs{endtitlingpage*}{endtitlingpage}
220 \let\andnext\and
221 \renewcommand*{\thanksmarkstyle}[1]{
222 \renewcommand{\thanksfootmark}{%
223 \thanksscript{\tmark}%
224 }
225
226 % \newlength{\thanksmarksep}

```

### § 340.6 Abstracts

```

227 \renewcommand*{\abstractcol}{
228 \renewcommand*{\abstractintoc}{
229 \renewcommand*{\abstractnum}{
230 \renewcommand*{\abstractrunin}{

```

### § 340.7 Document divisions

```

231
232 \def\@apppage{%
233     \part*{\appendixpagename}
234 }

```

```

235 \renewcommand\mempreaddappagetotochook{}
236 \renewcommand\mempostaddappagetotochook{}
237
238 \def\@sappage{%
239     \part*\appendixpagename}
240 }
241
242 \csletcs{frontmatter*}{frontmatter}
243 \csletcs{mainmatter*}{mainmatter}
244 \renewcommand*\raggedbottomsection{}
245 \renewcommand*\normalbottomsection{}
246 \renewcommand*\bottomsectionskip{}
247 \renewcommand*\bottomsectionpenalty{}
248 \csletcs{appendixpage*}{appendixpage}
249 \renewcommand*\namedsubappendices{}
250 \renewcommand*\unnamedsubappendices{}
251 \renewcommand*\setsecnumdepth[1]{}% todo tocvsec2
252 \renewcommand*\maxsecnumdepth[1]{}% todo tocvsec2
253 \renewcommand*\beforebookskip{}
254 \renewcommand*\afterbookskip{}
255 \renewcommand*\beforepartskip{}
256 \renewcommand*\afterpartskip{}
257 \renewcommand*\midbookskip{}
258 \renewcommand*\midpartskip{}
259 \renewcommand*\printbookname{}
260 \renewcommand*\booknamefont{}
261 \renewcommand*\booknamenenum{}
262 \renewcommand*\printbooknum{}
263 \renewcommand*\booknumfont{}
264 \renewcommand*\printpartname{}
265 \renewcommand*\partnamefont{}
266 \renewcommand*\partnamenenum{}
267 \renewcommand*\printpartnum{}
268 \renewcommand*\partnumfont{}
269 \renewcommand*\printbooktitle[1]{}
270 \renewcommand*\booktitlefont{}
271 \renewcommand*\printparttitle[1]{}
272 \renewcommand*\parttitlefont{}
273 \renewcommand*\bookpageend{}
274 \renewcommand*\bookblankpage{}
275 \renewcommand*\nobookblankpage{}
276 \renewcommand*\partpageend{}
277 \renewcommand*\partblankpage{}
278 \renewcommand*\nopartblankpage{}
279 \RenewDocumentCommand{\newleadpage}{s o m m}{}% todo
280 \RenewDocumentCommand{\renewleadpage}{s o m m}{}% todo
281 \renewcommand*\leadpagetoclevel{chapter}
282
283 \renewcommand*\openright{}
284 \renewcommand*\openleft{}

```

```

285 \renewcommand*{\openany}{}
286 \renewcommand*{\clearforchapter}{}
287 \renewcommand*{\memendofchapterhook}{}
288 \renewcommand*{\chapterheadstart}{}
289 % \newlength{\beforechapskip}
290 \renewcommand*{\afterchapternum}{}
291 % \newlength{\midchapskip}
292 \renewcommand*{\afterchaptertitle}{}
293 % \newlength{\afterchapskip}
294 \renewcommand*{\printchaptername}{}
295 \renewcommand*{\chapnamefont}{}
296 \renewcommand*{\chapternamenum}{}
297 \renewcommand*{\printchapternum}{}
298 \renewcommand*{\chapnumfont}{}
299 \renewcommand{\printchaptertitle}[1]{}
300 \renewcommand*{\chaptitelfont}{}
301 \renewcommand*{\printchapternonum}{}
302 \renewcommand*{\indentafterchapter}{}
303 \renewcommand*{\noindentafterchapter}{}
304 \renewcommand*{\insertchapterspace}{}
305
306 \renewcommand*{\chapterstyle}[1]{}
307 \renewcommand{\makechapterstyle}[2]{}
308 \renewcommand*{\chapindent}{}
309 \let\chapterprecis\cftchapterprecis
310 \let\chapterprecishere\cftchapterprecishere
311 \let\chapterprecistoc\cftchapterprecistoc
312 \renewcommand*{\precisfont}{}
313 \renewcommand*{\prechapterprecis}{}
314 \renewcommand*{\postchapterprecis}{}
315 \renewcommand{\precistocetext}[1]{}
316 \renewcommand*{\precistocfont}{}
317 \renewcommand*{\precistocformat}{}
318 % \newlength{\prechapterprecisshift}
319
320 \renewcommand*{\setbeforesecskip}[1]{}
321 \renewcommand*{\setaftersecskip}[1]{}
322 \renewcommand*{\setsecindent}[1]{}
323 \renewcommand*{\setsecheadstyle}[1]{}
324 \renewcommand*{\setbeforesubsecskip}[1]{}
325 \renewcommand*{\setaftersubsecskip}[1]{}
326 \renewcommand*{\setsubsecindent}[1]{}
327 \renewcommand*{\setsubsecheadstyle}[1]{}
328 \renewcommand*{\setbeforesubsubsecskip}[1]{}
329 \renewcommand*{\setaftersubsubsecskip}[1]{}
330 \renewcommand*{\setsubsubsecindent}[1]{}
331 \renewcommand*{\setsubsubsecheadstyle}[1]{}
332 \renewcommand*{\setbeforeparaskip}[1]{}
333 \renewcommand*{\setafterparaskip}[1]{}
334 \renewcommand*{\setparaindent}[1]{}

```

```

335 \renewcommand*\setparaheadstyle}[1]{}
336 \renewcommand*\setbeforesubparaskip}[1]{}
337 \renewcommand*\setaftersubparaskip}[1]{}
338 \renewcommand*\setsubparaindent}[1]{}
339 \renewcommand*\setsubparaheadstyle}[1]{}
340 \renewcommand{\@hangfrom}[1]{#1}
341 \renewcommand{\sethangfrom}[1]{}
342 \renewcommand{\setsecnumformat}[1]{}
343
344 \renewcommand*\hangsecnum{}
345 \renewcommand*\defaultsecnum{}
346
347 \renewcommand*\sechook{}
348 \renewcommand{\setsechook}[1]{}
349 \renewcommand*\subsechook{}
350 \renewcommand{\setsubsechook}[1]{}
351 \renewcommand*\subsubsechook{}
352 \renewcommand{\setsubsubsechook}[1]{}
353 \renewcommand*\parahook{}
354 \renewcommand{\setparahook}[1]{}
355 \renewcommand*\subparahook{}
356 \renewcommand{\setsubparahook}[1]{}
357
358 \RenewDocumentCommand{\plainbreak}{s m}{\begin{center}~\end{center}}
359
360 \RenewDocumentCommand{\fancybreak}{s +m}{%
361 \begin{center}#2\end{center}%
362 }
363
364 \RenewDocumentCommand{\plainfancybreak}{s m m +m}{%
365 \begin{center}#4\end{center}%
366 }
367
368 \RenewDocumentCommand{\pfbreak}{s}{%
369 \begin{center}
370 \pfbreakdisplay
371 \end{center}
372 }
373
374 % \newlength{\pfbreakskip}
375 \renewcommand{\pfbreakdisplay}{*\quad*\quad*}
376
377 \renewcommand{\makeheadstyles}[2]{}
378 \renewcommand*\headstyles[1]{}

```

## § 340.8 Pagination and headers

```

379 \renewcommand*\savepagenumber{}
380 \renewcommand*\restorepagenumber{}

```

```
381 \renewcommand*\uppercaseheads{}\{}
382 \renewcommand*\nouppercaseheads{}\{}
383
384 \renewcommand*\bookpagemark}[1]\{}
385 \renewcommand*\partmark}[1]\{}
386 \renewcommand*\bibmark{}\{}
387 \renewcommand*\indexmark{}\{}
388 \renewcommand*\glossarymark{}\{}
389
390 \LWR@origpagestyle{empty}
391 \renewcommand*\ps@empty{}\{}
392 \renewcommand*\makepagestyle}[1]\{}
393 \renewcommand*\emptyshook{}\{}%
394 % \renewcommand*\empty@oddhead{}\{}
395 % \renewcommand*\empty@oddfoot{}\{}
396 % \renewcommand*\empty@evenhead{}\{}
397 % \renewcommand*\empty@evenfoot{}\{}
398 \renewcommand*\@oddhead{}\{}
399 \renewcommand*\@oddfoot{}\{}
400 \renewcommand*\@evenhead{}\{}
401 \renewcommand*\@evenfoot{}\{}
402 \renewcommand*\aliaspagestyle}[2]\{}
403 \renewcommand*\coppagestyle}[2]\{}
404
405 \renewcommand*\makeevenhead}[4]\{}
406 \renewcommand*\makeoddhead}[4]\{}
407 \renewcommand*\makeevenfoot}[4]\{}
408 \renewcommand*\makeoddfoot}[4]\{}
409 \renewcommand*\makerunningwidth}[3]\{}
410 % \newlength{headwidth}
411 \renewcommand*\makeheadrule}[3]\{}
412 \renewcommand*\makefootrule}[3]\{}
413 \renewcommand*\makeheadfootruleprefix}[3]\{}
414 % \newlength{normalrulethickness}
415 % \setlength{normalrulethickness}{.4pt}
416 % \newlength{footruleheight}
417 % \newlength{footruleskip}
418 \renewcommand*\makeheadposition}[5]\{}
419 \renewcommand*\makepsmarks}[2]\{}
420 \renewcommand*\makeheadfootstrut}[3]\{}
421
422 \renewcommand*\createplainmark}[3]\{}
423 \renewcommand*\memUHead}[1]\{}
424 \renewcommand*\createmark}[5]\{}
425 \renewcommand*\clearplainmark}[1]\{}
426 \renewcommand*\clearmark}[1]\{}
427 \renewcommand*\addtopsmarks}[3]\{}
428 \renewcommand*\ifonlyfloats}[2]{#2}
429 \renewcommand*\mergepagefloatstyle}[3]\{}
430
```



```

431 \renewcommand*\framepichead{}
432 \renewcommand*\framepictextfoot{}
433 \renewcommand*\framepichook{}
434 \renewcommand*\showheadfootlocoff{}
435 \renewcommand*\showtextblocklocoff{}

```

### § 340.9 Paragraphs and lists

```

436 \renewcommand\hangfrom[1]{#1}
437 \let\centerfloat\centering
438 \renewcommand*\raggedyright[1][]{ }
439 % \newlength\ragrparindent
440 \renewcommand\sourceatright[2][]{\attribution{#2}}
441 \let\memorigdb\LWR@endofline
442 \let\memorigpar\par
443 \let\atcentercr\LWR@endofline
444 \renewcommand*\flushleftright{}
445 \renewcommand*\linenottooshort[1][]{ }
446 \renewcommand*\russianpar{}
447 \renewcommand*\lastlinerulefill{}
448 \renewcommand*\lastlineparrule{}
449 \renewcommand*\justlastraggedleft{}
450 \renewcommand*\raggedrightthenleft{}
451 \renewcommand*\leftcenterright{}
452
453 \renewcommand\leftspringright[4]{%
454 \begin{minipage}{#1\linewidth}#3\end{minipage}\quad%
455 \begin{minipage}{#2\linewidth}\begin{flushright}#4\end{flushright}\end{minipage}%
456 }
457
458 \renewenvironment*{blockdescription}
459 {\LWR@descriptionstart\LWR@origdescription}
460 {\enddescription}
461 \renewcommand*\blockdescriptionlabel[1]{\textbf{#1}}
462 \renewenvironment*{labelled}[1]{\begin{description}}{\end{description}}
463 \renewenvironment*{flexlabelled}[6]{\begin{description}}{\end{description}}
464 \renewcommand*\tightlists{}
465 \renewcommand*\defaultlists{}
466 \RenewDocumentCommand\firmlists{s}{}
467 \renewcommand*\firmlist{}
468 \renewcommand*\tightlist{}
469 \renewcommand*\zerotrivseps{}
470 \renewcommand*\savetrivseps{}
471 \renewcommand*\restoretrivseps{}

```

### § 340.10 Contents lists

```

472 \csletcs{tableofcontents*}{tableofcontents}
473 \csletcs{listoffigures*}{listoffigures}

```

```

474 \csletcs{listoftables*}{listoftables}
475 \renewenvironment{KeepFromToc}{}{}
476 \renewcommand*{\onecoltocetc}{}
477 \renewcommand*{\twocoltocetc}{}
478 \renewcommand*{\ensureonecol}{}
479 \renewcommand*{\restorefromonecol}{}
480 \renewcommand*{\doccoltocetc}{}
481 \renewcommand*{\maxtocdepth}[1]{}% tocvsec2
482 \renewcommand*{\settocdepth}[1]{}% tocvsec2
483
484 \renewcommand{\toheadstart}{}
485 \renewcommand{\printtoctitle}[1]{}
486 \renewcommand{\tocmark}{}
487 \renewcommand{\aftertoctitle}{}
488 \renewcommand{\lofheadstart}{}
489 \renewcommand{\printloftitle}[1]{}
490 \renewcommand{\lofmark}{}
491 \renewcommand{\afterloftitle}{}
492 \renewcommand{\lotheadstart}{}
493 \renewcommand{\printlottitle}[1]{}
494 \renewcommand{\lotmark}{}
495 \renewcommand{\afterlottitle}{}
496
497 \renewcommand*{\setpnumwidth}[1]{}
498 \renewcommand*{\setrmarg}[1]{}
499 \renewcommand*{\cftbookbreak}{}
500 \renewcommand*{\cftpartbreak}{}
501 \renewcommand*{\cftchapterbreak}{}

502 % \newlength{\cftbeforebookskip}
503 % \newlength{\cftbookindent}
504 % \newlength{\cftbooknumwidth}
505 \renewcommand*{\cftbookfont}{}
506 \renewcommand*{\cftbookname}{}
507 \renewcommand*{\cftbookpresnum}{}
508 \renewcommand*{\cftbookaftersnum}{}
509 \renewcommand*{\cftbookaftersnumb}{}
510 \renewcommand*{\cftbookleader}{}
511 \renewcommand*{\cftbookdotsep}{1}
512 \renewcommand*{\cftbookpagefont}{}
513 \renewcommand*{\cftbookafterpnum}{}
514 \renewcommand*{\cftbookformatpnum}[1]{}
515 \renewcommand*{\cftbookformatpnumhook}[1]{}

```

Part is already defined by tocloft.

```

516 % \newlength{\cftbeforechapterskip}
517 % \newlength{\cftchapterindent}
518 % \newlength{\cftchapternumwidth}
519 \renewcommand*{\cftchapterfont}{}

```

```

520 \renewcommand*{\cftchaptername}{}
521 \renewcommand*{\cftchapterpresnum}{}
522 \renewcommand*{\cftchapteraftersnum}{}
523 \renewcommand*{\cftchapteraftersnumb}{}
524 \renewcommand*{\cftchapterleader}{}
525 \renewcommand*{\cftchapterdotsep}{1}
526 \renewcommand*{\cftchapterpagefont}{}
527 \renewcommand*{\cftchapterafterpnum}{}
528 \renewcommand*{\cftchapterformatpnum}[1]{}
529 \renewcommand*{\cftchapterformatpnumhook}[1]{}

530 % \newlength{\cftbeforesections skip}
531 % \newlength{\cftsectionindent}
532 % \newlength{\cftsectionnumwidth}
533 \renewcommand*{\cftsectionfont}{}
534 \renewcommand*{\cftsectionname}{}
535 \renewcommand*{\cftsectionpresnum}{}
536 \renewcommand*{\cftsectionaftersnum}{}
537 \renewcommand*{\cftsectionaftersnumb}{}
538 \renewcommand*{\cftsectionleader}{}
539 \renewcommand*{\cftsectiondotsep}{1}
540 \renewcommand*{\cftsectionpagefont}{}
541 \renewcommand*{\cftsectionafterpnum}{}
542 \renewcommand*{\cftsectionformatpnum}[1]{}
543 \renewcommand*{\cftsectionformatpnumhook}[1]{}

544 % \newlength{\cftbeforesubsection skip}
545 % \newlength{\cftsubsectionindent}
546 % \newlength{\cftsubsectionnumwidth}
547 \renewcommand*{\cftsubsectionfont}{}
548 \renewcommand*{\cftsubsectionname}{}
549 \renewcommand*{\cftsubsectionpresnum}{}
550 \renewcommand*{\cftsubsectionaftersnum}{}
551 \renewcommand*{\cftsubsectionaftersnumb}{}
552 \renewcommand*{\cftsubsectionleader}{}
553 \renewcommand*{\cftsubsectiondotsep}{1}
554 \renewcommand*{\cftsubsectionpagefont}{}
555 \renewcommand*{\cftsubsectionafterpnum}{}
556 \renewcommand*{\cftsubsectionformatpnum}[1]{}
557 \renewcommand*{\cftsubsectionformatpnumhook}[1]{}

558 % \newlength{\cftbeforesubsubsection skip}
559 % \newlength{\cftsubsubsectionindent}
560 % \newlength{\cftsubsubsectionnumwidth}
561 \renewcommand*{\cftsubsubsectionfont}{}
562 \renewcommand*{\cftsubsubsectionname}{}
563 \renewcommand*{\cftsubsubsectionpresnum}{}
564 \renewcommand*{\cftsubsubsectionaftersnum}{}
565 \renewcommand*{\cftsubsubsectionaftersnumb}{}
566 \renewcommand*{\cftsubsubsectionleader}{}

```

```
567 \renewcommand*{\cftsubsubsectiondotsep}{1}
568 \renewcommand*{\cftsubsubsectionpagefont}{}
569 \renewcommand*{\cftsubsubsectionafterpnum}{}
570 \renewcommand*{\cftsubsubsectionformatpnum}[1]{}
571 \renewcommand*{\cftsubsubsectionformatpnumhook}[1]{}

572 % \newlength{\cftbeforeparagraphskip}
573 % \newlength{\cftparagraphindent}
574 % \newlength{\cftparagraphnumwidth}
575 \renewcommand*{\cftparagraphfont}{}
576 \renewcommand*{\cftparagraphname}{}
577 \renewcommand*{\cftparagraphpresnum}{}
578 \renewcommand*{\cftparagraphaftersnum}{}
579 \renewcommand*{\cftparagraphaftersnumb}{}
580 \renewcommand*{\cftparagraphleader}{}
581 \renewcommand*{\cftparagraphdotsep}{1}
582 \renewcommand*{\cftparagraphpagefont}{}
583 \renewcommand*{\cftparagraphafterpnum}{}
584 \renewcommand*{\cftparagraphformatpnum}[1]{}
585 \renewcommand*{\cftparagraphformatpnumhook}[1]{}

586 % \newlength{\cftbeforesubparagraphskip}
587 % \newlength{\cftsubparagraphindent}
588 % \newlength{\cftsubparagraphnumwidth}
589 \renewcommand*{\cftsubparagraphfont}{}
590 \renewcommand*{\cftsubparagraphname}{}
591 \renewcommand*{\cftsubparagraphpresnum}{}
592 \renewcommand*{\cftsubparagraphaftersnum}{}
593 \renewcommand*{\cftsubparagraphaftersnumb}{}
594 \renewcommand*{\cftsubparagraphleader}{}
595 \renewcommand*{\cftsubparagraphdotsep}{1}
596 \renewcommand*{\cftsubparagraphpagefont}{}
597 \renewcommand*{\cftsubparagraphafterpnum}{}
598 \renewcommand*{\cftsubparagraphformatpnum}[1]{}
599 \renewcommand*{\cftsubparagraphformatpnumhook}[1]{}

600 % \newlength{\cftbeforefigureskip}
601 % \newlength{\cftfigureindent}
602 % \newlength{\cftfigurenumwidth}
603 \renewcommand*{\cftfigurefont}{}
604 \renewcommand*{\cftfigurename}{}
605 \renewcommand*{\cftfigurepresnum}{}
606 \renewcommand*{\cftfigureaftersnum}{}
607 \renewcommand*{\cftfigureaftersnumb}{}
608 \renewcommand*{\cftfigureleader}{}
609 \renewcommand*{\cftfiguredotsep}{1}
610 \renewcommand*{\cftfigurepagefont}{}
611 \renewcommand*{\cftfigureafterpnum}{}
612 \renewcommand*{\cftfigureformatpnum}[1]{}
613 \renewcommand*{\cftfigureformatpnumhook}[1]{}

```

```

614 % \newlength{\cftbeforesubfigureskip}
615 % \newlength{\cftsubfigureindent}
616 % \newlength{\cftsubfigureenumwidth}
617 \newcommand*{\cftsubfigurefont}{}
618 \newcommand*{\cftsubfigurename}{}
619 \newcommand*{\cftsubfigurepresnum}{}
620 \newcommand*{\cftsubfigureaftersnum}{}
621 \newcommand*{\cftsubfigureaftersnumb}{}
622 \newcommand*{\cftsubfigureleader}{}
623 \newcommand*{\cftsubfiguredotsep}{1}
624 \newcommand*{\cftsubfigurepagefont}{}
625 \newcommand*{\cftsubfigureafterpnum}{}
626 \newcommand*{\cftsubfigureformatpnum}[1]{}
627 \newcommand*{\cftsubfigureformatpnumhook}[1]{}

628 % \newlength{\cftbeforetableskip}
629 % \newlength{\cfttableindent}
630 % \newlength{\cfttableenumwidth}
631 \renewcommand*{\cfttablefont}{}
632 \renewcommand*{\cfttablename}{}
633 \renewcommand*{\cfttablepresnum}{}
634 \renewcommand*{\cfttableaftersnum}{}
635 \renewcommand*{\cfttableaftersnumb}{}
636 \renewcommand*{\cfttableleader}{}
637 \renewcommand*{\cfttabledotsep}{1}
638 \renewcommand*{\cfttablepagefont}{}
639 \renewcommand*{\cfttableafterpnum}{}
640 \renewcommand*{\cfttableformatpnum}[1]{}
641 \renewcommand*{\cfttableformatpnumhook}[1]{}

642 % \newlength{\cftbeforesubtableskip}
643 % \newlength{\cftsubtableindent}
644 % \newlength{\cftsubtableenumwidth}
645 \newcommand*{\cftsubtablefont}{}
646 \newcommand*{\cftsubtablename}{}
647 \newcommand*{\cftsubtablepresnum}{}
648 \newcommand*{\cftsubtableaftersnum}{}
649 \newcommand*{\cftsubtableaftersnumb}{}
650 \newcommand*{\cftsubtableleader}{}
651 \newcommand*{\cftsubtabledotsep}{1}
652 \newcommand*{\cftsubtablepagefont}{}
653 \newcommand*{\cftsubtableafterpnum}{}
654 \newcommand*{\cftsubtableformatpnum}[1]{}
655 \newcommand*{\cftsubtableformatpnumhook}[1]{}

656 \renewcommand*{\booknumberline}[1]{}
657 \renewcommand*{\partnumberline}[1]{}
658 \renewcommand*{\chapternumberline}[1]{}
659 \renewcommand*{\numberlinehook}[1]{}
660 % \renewcommand*{\cftwhatismyname}{}%
```

```

661 \renewcommand*{\booknumberlinehook}[1]{}
662 \renewcommand*{\partnumberlinehook}[1]{}
663 \renewcommand*{\chapternumberlinehook}[1]{}
664 \renewcommand{\numberlinebox}[2]{}
665 \renewcommand{\booknumberlinebox}[2]{}
666 \renewcommand{\partnumberlinebox}[2]{}
667 \renewcommand{\chapternumberlinebox}[2]{}
668 %
669 % \newlength{\cftparfillskip}
670 \renewcommand*{\cftpagenumbersoff}[1]{}
671 \renewcommand*{\cftpagenumberon}[1]{}
672 \renewcommand*{\cftlocalchange}[3]{}
673 \renewcommand*{\cftaddtitleline}[4]{}
674 \renewcommand*{\cftaddnumtitleline}[4]{}
675 \renewcommand{\cftinsertcode}[2]{}
676 \renewcommand{\cftinserthook}[2]{}
677 \renewcommand{\settocpreprocessor}[2]{}
678 \DeclareRobustCommand{\cftpagenumbersoff}[1]{}
679 \DeclareRobustCommand{\cftpagenumberon}[1]{}

```

### § 340.11 Floats and captions

`\newfloat` [*<1: within>*] [*<2: type>*] [*<3: ext>*] [*<4: capname>*]

```

680 \RenewDocumentCommand{\newfloat}{o m m m}{%
681 \IfValueTF{#1}%
682 {\DeclareFloatingEnvironment[fileext=#3,within=#1,name={#4}]{#2}}%
683 {\DeclareFloatingEnvironment[fileext=#3,name={#4}]{#2}}}%

```

`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.

```

684 \cslet{listof#2s}\relax%
685 \cslet{listof#2es}\relax%
686 }

```

`\newlistof` [*<within>*] [*<type>*] [*<ext>*] [*<listofname>*]

Emulated through the `\newfloat` mechanism. Note that `memoir` uses a different syntax than `tocloft` for the name.

```

687 \RenewDocumentCommand{\newlistof}{o m m m}{
688 {%
689 \IfValueTF{#1}
690 {\newlistentry[#1]{#2}{#3}{0}}
691 {\newlistentry[#2]{#3}{0}}
692 \@namedef{ext@#2}{#3}%
693 \@ifundefined{c@#3depth}{\newcounter{#3depth}}{}%
694 \setcounter{#3depth}{1}%
695 \@namedef{#3mark}{}%
696 \@namedef{#2}{\listof{#2}{#4}}

```

```

697 \@namedef{@cftmake#3title}{}
698 \@ifundefined{cftbefore#3titleskip}{
699     \expandafter\newlength\csname cftbefore#3titleskip\endcsname
700     \expandafter\newlength\csname cftafter#3titleskip\endcsname
701 }{}
702 \@namedef{cft#3titlefont}{}
703 \@namedef{cftafter#3title}{}
704 \@namedef{cft#3prehook}{}
705 \@namedef{cft#3posthook}{}
706 }

```

```

707 \renewcommand{\setfloatadjustment}[2]{}

```

Borrowed from the lwarp version of keyfloat:

```

708 \NewDocumentEnvironment{KFLTmemoir@marginfloat}{0}{-1.2ex}{m}
709 {% start
710 \LWR@BlockClassWP{float:right; width:2in; margin:10pt}{}{marginblock}%
711 \captionsetup{type=#2}%
712 }
713 {%
714 \endLWR@BlockClassWP%
715 }
716
717 \DeclareDocumentEnvironment{marginfigure}{o}{}
718 {\begin{KFLTmemoir@marginfloat}{figure}}
719 {\end{KFLTmemoir@marginfloat}}
720
721 \DeclareDocumentEnvironment{margintable}{o}{}
722 {\begin{KFLTmemoir@marginfloat}{table}}
723 {\end{KFLTmemoir@marginfloat}}
724
724 \renewcommand{\setmarginfloatcaptionadjustment}[2]{}
725 \renewcommand{\setmpjustification}[2]{}
726 \renewcommand*{\mpjustification}{}
727 \renewcommand*{\setfloatlocations}[2]{}
728 \DeclareDocumentCommand{\suppressfloats}{o}{}
729 \renewcommand*{\FloatBlock}{}
730 \renewcommand*{\FloatBlockAllowAbove}{}
731 \renewcommand*{\FloatBlockAllowBelow}{}
732 \renewcommand*{\setFloatBlockFor}{}
733 \renewcommand*{\captiondelim}[1]{\renewcommand*{\CaptionSeparator}{#1}}
734 \renewcommand*{\captionnamefont}[1]{}
735 \renewcommand*{\captiontitlefont}[1]{}
736 \renewcommand*{\captionstyle}[2][{}
737 \renewcommand*{\centerlastline}{}
738 \renewcommand*{\hangcaption}{}
739 \renewcommand*{\indentcaption}[1]{}
740 \renewcommand*{\normalcaption}{}

```

```

741 \renewcommand*{\changecaptionwidth}{%
742 \DeclareDocumentCommand{\captionwidth}{m}{%
743 \renewcommand*{\normalcaptionwidth}{%
744 \renewcommand{\precaption}[1]{%
745 \renewcommand{\captiontitlefinal}[1]{%
746 \renewcommand{\postcaption}[1]{%
747
748 \renewcommand{\contcaption}[1]{%
749 % \ContinuedFloat%
750 % \caption{#1}%
751 \begin{LWR@figcaption}% later becomes \caption*
752 \csuse{\@capttype name} \thechapter.\the\value{\@capttype}\CaptionSeparator #1
753 \end{LWR@figcaption}
754 }

```

The extra \\ here forces a <br> in HTML when \legend is used in a \marginpar.

```

755 \renewcommand{\legend}[1]{\begin{center}#1\\end{center}}
756
757 \renewcommand{\namedlegend}[2][{}{
758 \begin{center}
759 \csuse{fleg\@capttype}\CaptionSeparator#2\\
760 \end{center}
761 \csuse{flegtoc\@capttype}{#1}
762 }
763
764 \renewcommand{\flegtable}{\tablename}
765 \renewcommand{\flegfigure}{\figurename}
766 \renewcommand{\flegtocfigure}{%
767 \renewcommand{\flegtocfigure}{%
768
769 \renewcommand{\newfixedcaption}[3][\caption]{%
770 \renewcommand{#2}{\def\@capttype{#3}#1}}
771 \renewcommand{\renewfixedcaption}[3][\caption]{%
772 \renewcommand{#2}{\def\@capttype{#3}#1}}
773 \renewcommand{\providefixedcaption}[3][\caption]{%
774 \providecommand{#2}{\def\@capttype{#3}#1}}
775
776 \renewcommand{\bitwonumcaption}[6][{}{
777 \ifblank{#2}{\caption{#3}}{\caption{#2}{#3}}%
778 \addtocounter{\@capttype}{-1}%
779 \begingroup%
780 \csdef{\@capttype name}{#4}%
781 \ifblank{#5}{\caption{#6}}{\caption{#5}{#6}}%
782 \endgroup%
783 \ifblank{#1}{\label{#1}}%
784 }
785
786 \LetLtxMacro\bionenumcaption\bitwonumcaption% todo

```



```

787
788 \renewcommand{\bication}[5][{}]{%
789 \ifblank{#2}{\caption{#3}}{\caption[#2]{#3}}%
790 \begin{LWR@figcaption}% later becomes \caption*
791 #4 \thechapter.\the\value{\@capttype}\CaptionSeparator #5
792 \end{LWR@figcaption}
793 \ifblank{#1}{\label{#1}}%
794 }
795
796 \renewcommand{\bicontcaption}[3]{%
797 \contcaption{#1}%
798 \begingroup%
799 \csdef{\@capttype name}{#2}%
800 \contcaption{#3}%
801 \endgroup%
802 }
803
804 \renewcommand{\midbication}[1]{%
805
806 \renewcommand{\subcaption}[2][{}]{%
807 \ifblank{#1}{\subfloat[#2]}{\subfloat[#1][#2]}%
808 }
809
810 \RenewDocumentCommand{\subtop}{O{} O{} m}{%
811 \subfloat[#1][#2]{#3}%
812 }
813
814 \RenewDocumentCommand{\subbottom}{O{} O{} m}{%
815 \subfloat[#1][#2]{#3}%
816 }
817
818 \renewcommand{\contsubcaption}{\ContinuedFloat\subcaption}
819
820 \renewcommand{\contsubtop}{%
821 \ContinuedFloat\addtocounter{\@capttype}{1}%
822 \subtop}
823
824 \renewcommand{\contsubbottom}{%
825 \ContinuedFloat\addtocounter{\@capttype}{1}%
826 \subbottom}
827
828 \renewcommand{\subconcluded}{}
829
830 \LetLtxMacro\subcaptionref\subref
831
832 \renewcommand*{\tightsubcaptions}{}
833 \renewcommand*{\loosesubcaptions}{}
834
835 \renewcommand*{\subcaptionsize}[1]{}
836 \renewcommand*{\subcaptionlabelfont}[1]{}

```

```

837 \renewcommand*{\subcaptionfont}[1]{}
838 \renewcommand*{\subcaptionstyle}[1]{}
839
840 \renewcommand*{\hangsubcaption}{}
841 \renewcommand*{\shortsubcaption}{}
842 \renewcommand*{\normalsubcaption}{}
843
844 \RenewDocumentEnvironment{sidecaption}{o m o}
845 {}
846 {
847 \IfValueTF{#1}{\caption[#1]{#2}}{\caption{#2}}%
848 \IfValueT{#3}{\label{#3}}%
849 }
850
851 % \newlength{\sidecapwidth}
852 % \newlength{\sidecapsep}
853 \renewcommand*{\setsidecaps}[2]{}
854 \renewcommand*{\sidecapmargin}[1]{}
855 % \newif\ifscapmargleft
856 \scapmargleftfalse
857 \renewcommand*{\setsidecappos}[1]{}
858
859 \RenewDocumentEnvironment{sidecontcaption}{m o}
860 {}
861 {%
862 \ContinuedFloat%
863 \caption{#1}%

```

Without \@capttype, the section is referred to instead.

```

864 \IfValueT{#2}{\label[\@capttype]{#2}}%
865 }

```

\sidenamedlegend does not appear to use the TOC argument.

```

866 \renewenvironment{sidenamedlegend}[2][ ]{
867 \begin{center}
868 \csuse{\@capttype name}\CaptionSeparator#2
869 \end{center}
870 }
871 {}
872
873 \renewenvironment{sidelegend}[1]
874 {\begin{center}
875 #1
876
877 }
878 {\end{center}}
879

```

```

880 \renewcommand*{\sidecapstyle}{}
881 \renewcommand*{\overridescapmargin}[1]{}
882 % \newlength{\sidecapraise}
883 \renewcommand*{\sidecapfloatwidth}{\linewidth}
884
885 \LetLtxMacro\ctabular\tabular
886 \LetLtxMacro\endctabular\endtabular
887
888 \renewcommand{\autorows}[5][\%
889 #5
890 }
891
892 \renewcommand{\autocol}[5][\%
893 #5
894 }

```

### § 340.12 Page notes

```

895 \renewcommand*{\feetabovelfloat}{}
896 \renewcommand*{\feetbelowfloat}{}
897 \renewcommand*{\feetatbottom}{}
898
899 \renewcommand*{\verbfootnote}[2][\{
900 \PackageError{lwarp, memoir}
901 {Verbatim footnotes are not yet supported by lwarp.}
902 {This may be improved some day.}
903 }
904
905 \renewcommand*{\plainfootnotes}{}
906 \renewcommand*{\twocolumnfootnotes}{}
907 \renewcommand*{\threecolumnfootnotes}{}
908 \renewcommand*{\paragraphfootnotes}{}
909 \renewcommand*{\footfudgefiddle}{}
910
911 \renewcommand*{\newfootnoteseries}[1]{
912 \PackageError{lwarp, memoir}
913 {Memoir footnote series are not yet supported by lwarp.}
914 {This may be improved some day.}
915 }
916
917 \renewcommand*{\plainfootstyle}[1]{}
918 \renewcommand*{\twocolumnfootstyle}[1]{}
919 \renewcommand*{\threecolumnfootstyle}[1]{}
920 \renewcommand*{\paragraphfootstyle}[1]{}
921
922 \renewcommand*{\footfootmark}{}
923 \renewcommand*{\footmarkstyle}[1]{}
924
925 % \newlength{\footmarkwidth}

```

---

```

926 % \newlength{\footmarksep}
927 % \newlength{\footparindent}
928
929 \renewcommand*{\foottextfont}{}
930
931 \renewcommand*{\marginparmargin}[1]{}
932 \renewcommand*{\sideparmargin}[1]{}
933
934 \LetLtxMacro\sidepar\marginpar
935 \renewcommand*{\sideparfont}{}
936 \renewcommand*{\sideparform}{}
937 \LWR@providelength{\sideparvshift}
938
939 \renewcommand*{\parnopar}{}
940
941 \renewcommand{\sidebar}[1]{\begin{quote}#1\end{quote}}
942 \renewcommand*{\sidebarmargin}[1]{}
943 \renewcommand*{\sidebarfont}{}
944 \renewcommand*{\sidebarform}{}
945 % \newlength{\sidebarhsep}
946 % \newlength{\sidebarvsep}
947 % \newlength{\sidebarwidth}
948 % \newlength{\sidebartopsep}
949 \renewcommand{\setsidebarheight}[1]{}
950 \renewcommand*{\setsidebars}[6]{}
951 \renewcommand*{\footnotesatfoot}{}
952 \renewcommand*{\footnotesinmargin}{}
953
954 \LetLtxMacro\sidefootnote\footnote
955 \LetLtxMacro\sidefootnotemark\footnotemark
956 \LetLtxMacro\sidefootnotetext\footnotetext
957
958 \renewcommand*{\sidefootmargin}[1]{}
959 % \newlength{\sidefoothsep}
960 % \newlength{\sidefootvsep}
961 % \newlength{\sidefootwidth}
962 % \newlength{\sidefootadjust}
963 % \newlength{\sidefootheight}
964 \renewcommand*{\setsidefootheight}[1]{}
965 % \renewcommand*{\sidefootfont}{}% in docs but not in the package
966 \renewcommand*{\setsidefeet}[6]{}
967 \renewcommand*{\sidefootmarkstyle}[1]{}
968 \renewcommand*{\sidefoottextfont}{}
969 \renewcommand*{\sidefootform}{}
970
971 \renewcommand*{\continuousnotenums}{\pncontopttrue}% from pagenote
972 \renewcommand*{\notepageref}{}
973 \renewcommand*{\prenotetext}{}
974 \renewcommand*{\postnotetext}{}
975 \renewcommand*{\idtextinnotes}[1]{}

```

```

976 \renewcommand*{\printpageinnotes}[1]{ }
977 \renewcommand*{\printpageinnoteshyperref}[1]{ }
978 \renewcommand*{\foottopagenote}{ }
979 \renewcommand*{\pagetofootnote}{ }

```

### § 340.13 Decorative text

```

980 \renewcommand*{\epigraphposition}[1]{ }
981 \renewcommand*{\epigraphtextposition}[1]{ }
982 \renewcommand*{\epigraphsourceposition}[1]{ }
983 \renewcommand*{\epigraphfontsize}[1]{ }
984 \renewcommand*{\epigraphforheader}[2][ ]{ }
985 \renewcommand*{\epigraphpicture}{ }

```

### § 340.14 Poetry

```

986 \renewcommand*{\vinphantom}{ }
987 \renewcommand*{\vleftofline}[1]{#1}
988 % \let\linenumberfrequency\poemlines
989 % \renewcommand*{\linenumberfont}[1]{ }
990
991 \DeclareDocumentCommand{\PoemTitle}{s o m}{%
992 \IfValueTF{#2}%
993 {\poemtitle[#2]{#4}}%
994 {\poemtitle{#4}}%
995 }
996
997 \renewcommand*{\NumberPoemTitle}{ }
998 \renewcommand*{\PlainPoemTitle}{ }
999 \renewcommand*{\poemtitlestyle}{ }
1000 \renewcommand*{\poemtitlestarmark}[1]{ }
1001 \renewcommand*{\poemtitlestarpstyle}{ }
1002 \renewcommand*{\PoemTitleheadstart}{ }
1003 \renewcommand*{\printPoemTitlenonum}{ }
1004 \renewcommand*{\printPoemTitlenum}{ }
1005 \renewcommand*{\afterPoemTitlenum}{ }
1006 \renewcommand*{\printPoemTitletitle}[1]{ }
1007 \renewcommand*{\afterPoemTitle}{ }
1008 \newlength{\midpoemtitleskip}
1009 \renewcommand*{\PoemTitlenumfont}{ }
1010 \renewcommand*{\PoemTitlefont}{ }

```

### § 340.15 Boxes, verbatims and files

```

1011 \renewenvironment{qframe}{\framed}{\endframed}
1012 \renewenvironment{qshade}{\shaded}{\endshaded}

```

Use the comment package:

```

1013 \renewcommand*{\commentsoff}[1]{\includecomment{#1}}
1014 \renewcommand*{\commentson}[1]{\excludecomment{#1}}
1015 \LetLtxMacro\renewcomment\commentson
1016
1017 \renewcommand*{\setverbatimfont}[1]{}
1018 \renewcommand*{\tabson}[1]{}
1019 \renewcommand*{\tabsoff}{}
1020 \renewcommand*{\wrappingon}{}
1021 \renewcommand*{\wrappingoff}{}
1022 \renewcommand*{\verbatimindent}{}
1023 \renewcommand*{\verbatimbreakchar}[1]{}

1024 \DefineVerbatimEnvironment{fboxverbatim}{Verbatim}{frame=single}

```

boxedverbatim is already defined by moreverb. boxedverbatim\* does not appear to work at all, even in a minimal print memoir document.

```

1025 \renewcommand*{\bvbox}{}
1026 \renewcommand*{\bvtopandtail}{}
1027 \renewcommand*{\bvsiides}{}
1028 \renewcommand*{\nobvbox}{}
1029 % \newlength\bvboxsep
1030 \renewcommand*{\bvtoprulehook}{}
1031 \renewcommand*{\bvtopmidhook}{}
1032 \renewcommand*{\bvendrulehook}{}
1033 \renewcommand*{\bvleftsidehook}{}
1034 \renewcommand*{\bvrightsidehook}{}
1035 \renewcommand*{\bvperpagetrue}{}
1036 \renewcommand*{\bvperpagefalse}{}
1037 \renewcommand*{\bvtopofpage}[1]{}
1038 \renewcommand*{\bvendofpage}[1]{}
1039 \renewcommand*{\linenumberfrequency}[1]{}
1040 \renewcommand*{\resetbvlینumber}{}
1041 \renewcommand*{\setbvlینums}[2]{}
1042 \renewcommand*{\linenumberfont}[1]{}
1043 \renewcommand*{\bvnumbersinside}{}
1044 \renewcommand*{\bvnumbersoutside}{}

```

## § 340.16 Cross referencing

```

1045 \renewcommand*{\fref}[1]{\cref{#1}}
1046 \renewcommand*{\tref}[1]{\cref{#1}}
1047 \renewcommand*{\pref}[1]{\cpageref{#1}}
1048 \renewcommand*{\Aref}[1]{\cref{#1}}
1049 \renewcommand*{\Bref}[1]{\cref{#1}}
1050 \renewcommand*{\Pref}[1]{\cref{#1}}
1051 \renewcommand*{\Sref}[1]{\cref{#1}}
1052 \renewcommand*{\figurerefname}{Figure}
1053 \renewcommand*{\tablerefname}{Table}

```

```

1054 \renewcommand*{\pagerefname}{page}
1055 \renewcommand*{\bookrefname}{Book~}
1056 \renewcommand*{\partrefname}{Part~}
1057 \renewcommand*{\chapterrefname}{Chapter~}
1058 \renewcommand*{\sectionrefname}{\S}
1059 \renewcommand*{\appendixrefname}{Appendix~}
1060 \LetLtxMacro\titleref\nameref
1061 \renewcommand*{\headnameref}{\S}
1062 \renewcommand*{\tocnameref}{\S}
1063
1064 \providecounter{LWR@currenttitle}
1065
1066 \renewcommand*{\currenttitle}{%
1067     \addtocounter{LWR@currenttitle}{1}%
1068     \label{currenttitle\arabic{LWR@currenttitle}}%
1069     \nameref{currenttitle\arabic{LWR@currenttitle}}%
1070 }
1071
1072 \renewcommand*{\theTitleReference}[2]{\S}
1073 \renewcommand*{\namerefon}{\S}
1074 \renewcommand*{\namerefoff}{\S}

```

### § 340.17 Back matter

```

1075 \DeclareDocumentCommand{\newblock}{\S}{\S}
1076 %
1077 \renewcommand*{\showindexmarks}{\S}
1078 \renewcommand*{\hideindexmarks}{\S}
1079
1080 \renewcommand*{\xindyindex}{\S}

```

### § 340.18 Miscellaneous

```

1081 \renewcommand*{\changemarks}{\S}
1082 \renewcommand*{\nochangemarks}{\S}
1083 \renewcommand*{\added}[1]{\S}
1084 \renewcommand*{\deleted}[1]{\S}
1085 \renewcommand*{\changed}[1]{\S}
1086
1087 \renewcommand*{\showtrimsoff}{\S}
1088 \renewcommand*{\showtrimson}{\S}
1089 \renewcommand*{\trimXmarks}{\S}
1090 \renewcommand*{\trimLmarks}{\S}
1091 \renewcommand*{\trimFrame}{\S}
1092 \renewcommand*{\trimNone}{\S}
1093 \renewcommand*{\trimmarkscolor}{\S}
1094 \renewcommand*{\trimmarks}{\S}
1095 \renewcommand*{\tmarktl}{\S}
1096 \renewcommand*{\tmarktr}{\S}

```

```
1097 \renewcommand*{\tmarkbr}{}  
1098 \renewcommand*{\tmarkbl}{}  
1099 \renewcommand*{\tmarktm}{}  
1100 \renewcommand*{\tmarkmr}{}  
1101 \renewcommand*{\tmarkbm}{}  
1102 \renewcommand*{\tmarkml}{}  
1103 \renewcommand*{\trimmark}{}  
1104 \renewcommand*{\quarkmarks}{}  
1105 \renewcommand*{\registrationColour}[1]{}  
1106  
1107 \renewcommand*{\leavespergathering}[1]{}  
1108  
1109 \renewcommand*{\noprelistbreak}{}  
1110  
1111 \renewcommand*{\cleartorecto}{}  
1112 \renewcommand*{\cleartoverso}{}  
1113  
1114 \renewenvironment{vplace}[1][ ]{}{}
```

#### § 340.19 **Final patchwork**

```
1115 \newlistof{tableofcontents}{toc}{\contentsname}  
1116 \newlistof{listoffigures}{lof}{\listfigurename}  
1117 \newlistof{listoftables}{lot}{\listtablename}
```



# Change History and Index

## § 340 Change History

v0.10	General: 2016/03/08 Initial version . . . 1	Docs: Table: Float data structures. 390
v0.11	General: 2016/03/11 . . . . . 1	Docs: Trademarks section. . . . . 484
	Added section: Operating-System	Docs: Troubleshooting
	portability. . . . . 146	cross-references. . . . . 136
	Added section: Selecting the	Test Suite: Assigned cleveref name
	operating system. . . . . 93	for Test Float. . . . . 1
	Test Suite: MS-WINDOWS in	Test Suite: Floatrow . . . . . 1
	README.txt . . . . . 1	
	Test Suite: limages and index in	
	README.txt . . . . . 1	
v0.12	\LWR@newhtmlfile: Bugfix: TOC	
	with numbered files. . . . . 275	
	General: 2016/03/14 . . . . . 1	
	Global: Uses \p@(type) in float	
	captions. . . . . 1	
	Test Suite: Sub-figures . . . . . 1	
v0.13	\CaptionSeparator: Fix for newer	
	babel package. . . . . 393	
	\LWR@LwarpStart: \up and \fup . . 292	
	General: 2016/03/24 . . . . . 1	
	Removed package: subfig . . . . . 1	
	Test Suite: Ordinals, Subcaption . . 1	
	tikzpicture: Fix dollar-redefined	
	bug for newer package. . . . . 699	
v0.14	\LWR@htmlsectionfilename: Fix:	
	Links to home page. . . . . 241	
	General: 2016/03/31 . . . . . 1	
	floatrow: Added. . . . . 548	
	Docs: Commands for a successful	
	HTML conversion. . . . . 98	
	Docs: Commands into a warpprint	
	environment. . . . . 95	
	Docs: Newclude limitations. . . . 123	
	Docs: Table: Cross-referencing	
	data structures. . . . . 379	
		Docs: Table: Float data structures. 390
		Docs: Trademarks section. . . . . 484
		Docs: Troubleshooting
		cross-references. . . . . 136
		Test Suite: Assigned cleveref name
		for Test Float. . . . . 1
		Test Suite: Floatrow . . . . . 1
v0.15	General: 2016/04/06 . . . . . 1	
	Added . . . . . 553	
	Ampersand (&): Fixed handling	
	when passed as an argument. . . 329	
	Docs: Added warning icons for	
	items needing special attention. 142	
	Docs: Clarify print/HTML output. 94	
	Docs: Moved the supported	
	features table to the introduction. 54	
	Files: lwarp_formal.css added. . . . 1	
	Fix: steps counter . . . . . 553	
	Fixed & handling. . . . . 551	
	Test Suite: test_suite_formal.css	
	file added. . . . . 1	
v0.16	General: 2016/04/11 . . . . . 1	
	\titlingpage: Improved	
	print-output spacing. . . . . 300	
	xfrac: Adjusted for the use of any	
	font: . . . . . 749	
	Added XeLaTeX, LuaLaTeX	
	support. . . . . 143	
	Docs: Font and UTF-8 support. . . 83	
	Docs: Moved location of	
	\usepackage{lwarp}. . . . . 85	
	Docs: Text not converting. . . . . 136	
	Fix: amsmath options clash . . . . 161	
	Fix: newtxmath compatibility. . . 162	
	Lwarp no longer selects fonts. 83, 158	
	Removed package: suffix . . . . . 1	
	Test Suite: Improved titlingpage. 300	

Test Suite: Lwarp no longer selects fonts. . . . .	1	verse: Supports verse, memoir packages. . . . .	731
Test Suite: Supports XeLaTeX, LuaLaTeX. . . . .	1	minipage: Fix: \linewidth, \textwidth, \textheight inside a minipage. . . . .	456
v0.17		v0.19	
\LWR@htmlsectionfilename: Fix: Links when entire doc is one HTML page. . . . .	241	\HTMLFilename: Docs: Escape filename underscores. . . . .	241
General: 2016/04/14 . . . . .	1	\HomeHTMLFilename: Docs: Escape filename underscores. . . . .	241
mdframed: Added. . . . .	606	\LWR@LwarpStart: Enabled \\ equal to \newline. . . . .	291
Test Suite: Fix: Print-version front-matter page numbers. . . . .	1	\LWR@doubledollar: MATHJAX support. . . . .	415
Test Suite: Mdfamed . . . . .	1	\LWR@filestart: lwarp_mathjax.txt loaded. . . . .	289
v0.18		\LWR@hspace: Fix: \hspace length computations. . . . .	474
\LWR@hspace: \hspace supported. . . . .	474	\LWR@minipagestartpars: Surpresses paragraph tags between minipages. . . . .	473
\LWR@includegraphicsb: Add: svgz file extension. . . . .	570	\LWR@subsingledollar: MATHJAX support. . . . .	416
em, ex, %, px dimensions preserved. . . . .	570	\LateximageFontSizeName: Add: User-adjustable math/lateximage font size. . . . .	434
Fix: \linewidth, \textwidth, \textheight inside a minipage. . . . .	570	\minipagefullwidth: Added: No width tag for the next minipage in HTML. . . . .	456
Improved HTML output linebreaks. . . . .	570	\rule: Added . . . . .	479
\LWR@myshorttoc: Reorganize \HomeHTMLFilename logic. . . . .	397	\warpHTMLonly: Added. . . . .	150
\LWR@newhtmlfile: sideroc after title, improving responsive design. . . . .	274	\warpprintonly: Replaces \rowprintedonly. . . . .	150
\LWR@requesttoc: Reorganize \HomeHTMLFilename logic. . . . .	294	\xfracHTMLfontsize: Added. . . . .	749
\LWR@subhyperref: Improved HTML output linebreaks. . . . .	387	General: 2016/06/08 . . . . .	1
\LWR@subhyperrefclass: Improved HTML output linebreaks. . . . .	387	MATHJAX support added. . . . .	420, 424, 426
\LWR@subinlineimage: Surpress extra space. . . . .	388	multirow: Added optional args. . . . .	623
General: 2016/05/19 . . . . .	1	Avoids MATHJAX. . . . .	415
File: lwarp.css: Improved TOC outline display. . . . .	1	cleveref: Loaded \AtEndPreamble. . . . .	451
Files: lwarp.css and lwarp_formal.css: Improved responsive design. . . . .	1	CSS for table note item. . . . .	698
Microtype disabled during HTML generation . . . . .	158	Docs: Math options. . . . .	85
PDF Unicode input characters. . . . .	144	Docs: Table: Cross-referencing data structures, updated. . . . .	379
Test Suite: Verse package . . . . .	1	File: lwarp.css: tnoteitemheader added. . . . .	1
lateximage: pdfcrop: --hires added. . . . .	438	File: lwarp_mathjax.txt added. . . . .	1
Reorganize \HomeHTMLFilename logic. . . . .	438	Introduction: MATHJAX support mentioned. . . . .	52
Surpress extra space. . . . .	438	Options: mathsvg and mathjax . . . . .	148

Supports colored <code>\rule</code> . . . . .	742	<code>\LWR@subhtmlclass</code> :	
<code>titles: null \pagestyle</code> and		Factored code. . . . .	252
<code>\thispagestyle</code> for HTML. . .	700	<code>\SetHTMLFileNumber</code> : Add: Control	
<code>tikzpicture</code> : Adapts to tikz version. . . . .	699	file numbers. . . . .	241
<code>equation</code> : MATHJAX support. . . . .	422	<code>\cpagerefFor</code> : User-redefinable	
v0.20		word for page references. . . . .	452
<code>\BlockClassSingle</code> : Renamed from		<code>\dotfill</code> : Inserts an ellipsis. . . . .	472
" <code>LWR@htmldivclassline</code> ". . . . .	254	<code>\hfill</code> : Inserts a <code>\quad</code> . . . . .	472
<code>\HTMLDescription</code> : Added		<code>\hrulefill</code> : Inserts a short rule. . . . .	472
<code>\NewHTMLdescription</code> .		<code>\hyperindexref</code> : Print mode	
(Renamed in v0.30.) . . . . .	264	provided in case <code>hyperref</code> not	
<code>\HTMLFilename</code> : No longer escape		used. . . . .	407
underscores. . . . .	241	<code>\pageref</code> : Added. . . . .	386
<code>\HomeHTMLFilename</code> : No longer		<code>\tracinglwarp</code> : Added. . . . .	169
escape underscores. . . . .	241	General: 2017/02/09 . . . . .	1
<code>\InlineClass</code> : Renamed from		afterpage: Added. . . . .	492
" <code>inlineclass</code> ". . . . .	254	alltt: Added. . . . .	494
<code>\LWR@LwarpStart</code> : Fix: math cross		bookmark: Added. . . . .	505
references. . . . .	293	caption and subcaption	
<code>\LWR@closeparagraph</code> : <code>\unskip</code>		supported. . . . .	1
extra spaces. . . . .	258	cleveref and referencing patches:	
No break tags in the start/end of a		Applied <code>\AfterEndPreamble</code> . . . . .	451
tabular. . . . .	258	draftwatermark: Added. . . . .	522
<code>\LWR@endoffline</code> : Fix: <code>\\</code> . . . . .	473	eso-pic: Added. . . . .	529
<code>\LWR@filestart</code> : Adds meta		everypage: Added. . . . .	529
description. . . . .	289	extramarks: Added. . . . .	530
<code>\LWR@hspace</code> : Add: Supports HTML		fancyhdr: Added. . . . .	535
thin breakable space. . . . .	474	hyperref: Additional user macros. . . . .	581
<code>\LWR@htmldivclass</code> : Added		keyfloat: Added. . . . .	588
optional style. . . . .	252	letterspace: User-interface	
<code>\LWR@htmlclass</code> : Added		emulated. . . . .	591
optional style. . . . .	252	listings: Added. . . . .	595
<code>\LWR@htmlsectionfilename</code> :		ltcaption: Added. . . . .	601
<code>HTMLFilename</code> : removed		lwarp-newproject: Added. . . . .	177
additional trailing <code>'</code> , and may be		microtype: User-interface	
empty. . . . .	241	emulated. . . . .	617
Sections called "Index" or "index"		needspace: Added. . . . .	627
have an underscore prepended to		nowidow: Added. . . . .	630
their filenames if no prefix. . . . .	241	placeins: Added. . . . .	647
<code>\LWR@includegraphicsb</code> : Fix:		ragged2e: Added. . . . .	650
<code>\linewidth</code> in a floatrow. . . . .	570	setspace: Improved support. . . . .	665
Fix: Expands filename. . . . .	570	textpos: Added. . . . .	693
<code>\LWR@longtabledatacaptiontag</code> :		titles: Added. . . . .	700
Fix: Pars in captions. . . . .	362	titlesec: Added. . . . .	704
<code>\LWR@section</code> : Combined		titletoc: Added. . . . .	705
higher-level sections together		titling: Improved compatibility. . . . .	707
into files. . . . .	281	tocloft: Added. . . . .	715
<code>\LWR@setOSWindows</code> : Auto-detects		wallpaper: Added. . . . .	735
operating system. . . . .	146	wrapfig: Added. . . . .	736
		Added @, <, > columns. . . . .	323

Added single-expansion data arrays. . . . .	238	lwarpmk: Fix: lwarpmk limages for WINDOWS. . . . .	222
Code factored into independent lwarpmk files. . . . .	484	lwarpmk: Fix: lwarpmk uses lateximages text file instead of shell script. . . . .	222
Docs: Examples for generating HTML file names. . . . .	91	Add: Errors for misplaced packages. . . . .	151
Docs: Improved index. . . . .	1	Docs: Added internet class. . . . .	58
Enhanced titling support. . . . .	299	Docs: Added TeX2page, GladTeX. . . . .	58
File: lwarpmk.css: Minor fixes for validation. . . . .	1	Docs: Installing on WINDOWS. . . . .	64
File: lwarpmk used to compile print, HTML, indexes, and lateximages. . . . .	1	File: lwarpmk_tutorial.txt added. . . . .	69
Fix: \linewidth in a floatrow. . . . .	551	v0.22	
Improved float caption type handling. . . . .	546	\LWR@parseDcolumn: Added tabular D column. . . . .	336
Moved sidebar and example code to test suite. . . . .	1	\LWR@parsebangcolumn: Added tabular ! column. . . . .	334
Page geometry set to 6in wide with large margins. . . . .	159	\LWR@parsetablecols: Unknown table column types become 1. Added tabular D, !, X columns. . . . .	337
Parallel versions of aux files for print/HTML. . . . .	1	\LWR@printmccoldata: Added tabular D, !, and X columns. . . . .	358
Removed reliance on make, grep, gawk. . . . .	1	General: 2017/03/02 . . . . .	1
Tabular: \unskip extra spaces. . . . .	323	abstract: Added. . . . .	486
Test Suite: HTML meta descriptions. . . . .	1	changepage: Added. . . . .	513
verbatim: Added. . . . .	310	dcolumn: Added. . . . .	520
BlockClass: Added optional style. . . . .	253	ftnright: Added. . . . .	562
Renamed from "blockclass". . . . .	253	geometry: Nullified commands. . . . .	563
LWR@nestspan: Fix: Minipages inside a span. . . . .	249	indentfirst: Added. . . . .	587
v0.21		layout: Added. . . . .	590
\LWR@LwarpStart: Changed lateximages to a .txt file. . . . .	291	lscape: Added. . . . .	601
\LWR@filestart: Skip title if not given. . . . .	289	mcaption: Added. . . . .	606
\LWR@newhtmlfile: Skip title if not given. . . . .	274	nameref: Added. . . . .	626
\marginpar: Fixed source listing. . . . .	270	nextpage: Added. . . . .	628
\marginparBlock: Fixed source listing. . . . .	270	parskip: Added. . . . .	645
General: 2017/02/23 . . . . .	1	showkeys: Added. . . . .	667
fontenc: Added. . . . .	556	sidecap: Added. . . . .	668
fontspec: Added. . . . .	556	tabularx: Added. . . . .	687
inputenc: Added. . . . .	588	varioref: Supported. . . . .	105
newclude: Added. . . . .	628	verse: Added. . . . .	730
newunicodechar: Added. . . . .	628	v0.23	
lwarpmk: Fix: lwarpmk again for WINDOWS. . . . .	222	\LWR@parsetablecols: Fix for vert bar column type. . . . .	337
		\LWR@printmccoldata: Fix for vert bar column type. . . . .	358
		General: 2017/03/02 . . . . .	1
		v0.24	
		\LWR@hspace: Add: \hspace \fill converts to 2em . . . . .	474
		\LWR@htmlfileref: Fix: Index links while \tracinglwarpmk. . . . .	382

\hypertocfloat: List of floats responds to lofdepth, lotdepth. . . . .	404	mparhack: Added. . . . .	621
General: 2017/03/15 . . . . .	1	pagenote: Supported as-is. . . . .	643
floatrow: Support for subfig. . . . .	548	sidenotes: Added. . . . .	668
subfig: Added. . . . .	679	Docs: Improved MiKTeX install instructions. . . . .	63, 65
tikz: For tikz v3.0.0 or later, auto-loads tikz babel library if necessary. . . . .	699	Dollar span avoided in a lateximage. . . . .	415
Docs: Filename underscore. . . . .	88, 99	Footnotes now are $\LaTeX$ boxes instead of pagenotes. . . . .	265
No longer preloads subcaption; conflicted with subfig. . . . .	161	lateximage: Labels track page numbers of lateximages. . . . .	438
picture: Fix for inline images. . . . .	454	Print mode now uses a minipage of \linewidth. . . . .	438
tikzpicture: Fix for inline images. . . . .	699	picture: Fix for \makebox in picture. . . . .	454
v0.25		v0.27	
\LWR@loadnever: Added the ability to prevent conflicting packages. . . . .	152	\LWR@footnotetext: Fix for table footnote par tags. . . . .	266
\addcontentsline: Handles theorems. . . . .	396	General: 2017/04/04 . . . . .	1
General: 2016/03/22 . . . . .	1	lettrine: Added. . . . .	591
amsthm: Added. . . . .	494	microtype: Fix with Xe $\LaTeX$ , Lua $\LaTeX$ . . . . .	617
ccaption: Prevented. . . . .	512	soul: Added. . . . .	675
ellipsis: Added. . . . .	523	ulem: Added. . . . .	727
emptypage: Added. . . . .	524	Docs: Installing utilities for MACOS. . . . .	66
framed: Added. . . . .	559	Docs: Limitations of saveboxes. . . . .	100
lips: Added. . . . .	595	Page geometry modified to reduce line overflow. . . . .	159
mdframed: Help avoid hyphenation. . . . .	608	v0.28	
nththeorem: Added. . . . .	630	\@wrindex: Improved indexing. . . . .	407
showidx: Added. . . . .	667	\HTMLAuthor: Added \HTMLauthor. (Renamed in v0.30.) . . . . .	264
theorem: Added. . . . .	694	\LWR@LwarpEnd: If FormatEPUB or FormatWP, no bottom nav. . . . .	294
Basic $\LaTeX$ theorems: improved css. . . . .	311	\LWR@LwarpStart: FormatWordProcessor forces single-file output. . . . .	291
Docs: Adds credits for patched code. . . . .	1	\LWR@filestart: Adds HTML meta author. . . . .	289
Docs: Testing lwarp. . . . .	135	\LWR@forcenewpage: Forces new PDF page before major environments. . . . .	245
Fix: Allows Xe $\LaTeX$ and Lua $\LaTeX$ to preload graphics and graphicx. . . . .	153	\LWR@htmlcomment: Breaks ligatures in HTML comments. . . . .	251
v0.26		\LWR@includegraphicsb: Adapts to graphics syntax. . . . .	570
General: 2017/03/31 . . . . .	1	\LWR@newhtmlfile: If FormatEPUB or FormatWP: skips headers, footers, nav. . . . .	274
lwarp.css: Improved responsive marginpar and marginblock. . . . .	180		
cutwin: Added. . . . .	518		
endnotes: Added. . . . .	525		
floatflt: Added. . . . .	547		
footmisc: Added. . . . .	557		
footnotehyper: Added. . . . .	559		
footnote: Added. . . . .	558		
marginfix: Added. . . . .	605		
marginnote: Added. . . . .	605		

\LWR@parsetablecols: Added L, C, R, J column types. . . . .	337	\HTMLLanguage: Renamed from \MetaLanguage. . . . .	288
\LWR@startref: Removed space. . .	384	\HTMLPageBottom: Renamed from \SetPageBottom. . . . .	262
\chapter: If EPUB, prints footnotes before each section. . . . .	286	\HTMLPageTop: Renamed from \SetPageTop. . . . .	262
\hyperindexref: Improved indexing. . . . .	407	General: 2017/04/29 . . . . .	1
\textup: Fixed span class. . . . .	466	lwarmpm-newproject removed, and combined with lwarpm. . . . .	177
General: 2017/04/14 . . . . .	1	lwarpmk: Add: xdyfile configuration option. . . . .	222
glossaries: Added. . . . .	564	lwarpmk: Fix: xindy and texindy adjusted for pdflatex, xelatex and lua <del>l</del> atex. . . . .	222
graphics: Added. . . . .	565	lwarpmk: Fix: xindy now used for print index generation with latexmk. . . . .	222
tabularx: Fix for optional pos. . .	687	lwarpmk: language now used for both index and glossary generation. . . . .	222
tabulary: Added. . . . .	688	File: lwarpm_html.xdy renamed to lwarpm.xdy. . . . .	218
lwarpmk: Add: printglossary and htmlglossary commands. . . . .	222	Fix: *.css files only written in print mode. . . . .	180
Added boolean FormatEPUB. . . . .	170	Fix: lwarpm.xdy only written in print mode. . . . .	218
Added boolean FormatWP. . . . .	170	Fix: lwarpm_mathjax.txt: Only written in print mode. . . . .	219
Added boolean HTMLDebugComments. . . . .	169	Option lwarpmklang changed to IndexLanguage. . . . .	148
Added boolean HTMLMarkFloats, changed to WPMarkFloats as of v0.42. . . . .	170	Option OSWindows replaces macro \warpOSWindows. . . . .	149
Docs: Modfying lwarpmk and index processing. . . . .	135	Option xdyFilename added. . . . .	148
File: lwarpm_mathjax.txt: Updated CDN repository. . . . .	219	Option latexmk replaces macro \UseLatexmk. . . . .	149
Forced oneside to maintain large right margin. . . . .	159	Options HomeHTMLFilename and HTMLFilename replace macros \HomeHTMLFilename and \HTMLFilename. . . . .	149
v0.29		v0.31	
\LWR@includegraphicsb: Fix: Error when no optional arguments. . .	570	General: 2017/05/15 . . . . .	1
General: 2017/04/15 . . . . .	1	keyfloat: Improved compatibility. . . . .	588
lwarpmk: Add: language option for config files. . . . .	222	v0.32	
Add: lwarpmklang option for lwarpm. . . . .	148	\RequirePackage: Fix: Ignores blanks in package list. . . . .	164
Docs: Using a glossary . . . . .	80	General: 2016/06/09 . . . . .	1
File: *.lwarpmkconf: Add: language option for config files. . . . .	179	glossaries: Prevent error with \glo@name not defined. . . . .	408
File: lwarpmk.conf: Add: language option for config files. . . . .	178		
v0.30			
\CSSFilename: Renamed from \NewCSS. . . . .	263		
\HTMLAuthor: Renamed from \HTMLauthor. . . . .	264		
\HTMLDescription: Renamed from \NewHTMLdescription. . . . .	264		
\HTMLFirstPageTop: Renamed from \SetFirstPageTop. . . . .	262		

lwarpmk: Fix: <code>io.lines()</code> changed to <code>file:lines()</code> due to luatex changes. . . . .	222	\LWR@nullfonts: Improved font control. . . . .	467
v0.33		\LWR@restoreorigformatting: booktabs: Works inside lateximage. . . . .	409
\HTMLAuthor: Fix: Provides empty default author if none given. . .	264	Improved font control. . . . .	409
\LWR@loadbefore: Fix: No \PackageError if already loaded. . . . .	152	\LWR@subhtmlclass: Moved optional argument in front of mandatory. . . . .	252
\LWR@parseatcolumn: Fix: Column alignment with leftmost @. . . .	333	\LWR@tabledatacolumn: booktabs: Works inside lateximage. . . . .	370
\LWR@tabledatasinglecolumn: Fix: Macros in tabular could cause extra data cell. . . . .	346	\fboxBlock: Added. . . . .	462
\LWR@vspace: Add: <code>\vspace</code> nullified. . . . .	476	\makebox: Fix: Handles paren arg. .	460
\StartDefiningTabulars: Add: Avoids error: misplaced alignment tab character &. . . .	329	General: 2017/08/08 . . . . .	1
General: 2017/07/10 . . . . .	1	babel-french: Adds fixed-width HTML spaces to punctuation. . .	247
amsmath: Removed <code>fleqn</code> option. . . . .	162	balance: Added. . . . .	502
fancyhdr: Fix: Optional args for <code>\lhead</code> , etc. . . . .	535	booktabs: Works inside lateximage. . . . .	373, 505
Add: Tabular at and bang columns now have their own HTML columns. . . . .	323	boxedminipage2e: Added. . . . .	506
cleveref: Fix: Loaded <code>\AtEndPreamble</code> . . . . .	451	boxedminipage: Prevented. . . . .	506
Fix: Incorrectly-inline math environments. . . . .	426	crop: Added. . . . .	517
New handling of & to localize catcode changes. . . . .	323	enumerate: Added. . . . .	526
v0.34		enumitem: Added, no longer required. . . . .	526
\@fnsymbol: Text symbols instead of math. . . . .	302	everyshi: Added. . . . .	530
\InlineClass: Moved optional argument in front of mandatory. .	254	fancybox: Added. . . . .	532
\LWR@htmldivclass: Moved optional argument in front of mandatory. . . . .	252	fancyvrb: Added, no longer required. . . . .	537
\LWR@htmlclass: Moved optional argument in front of mandatory. . . . .	252	figcaps: Added. . . . .	542
\LWR@htmlclassline: Moved optional argument in front of mandatory. . . . .	253	filecontents: Required. Patched for morewrites. . . . .	160
\LWR@htmlspanclass: Moved optional argument in front of mandatory. . . . .	250	floatpag: Added. . . . .	548
		flushend: Added. . . . .	554
		fullpage: Added. . . . .	562
		hyperxmp: Added. . . . .	585
		idxlayout: Added. . . . .	586
		marginfit: Added. . . . .	604
		mdframed: Improved <code>mdtheorem</code> patch. . . . .	614
		moreverb: Added. . . . .	619
		paralist: Added. . . . .	644
		pdfscape: Added. . . . .	645
		pdfsync: Added. . . . .	646
		prelim2e: Added. . . . .	648
		rotfloat: Added. . . . .	654
		savetrees: Added. . . . .	655
		shadow: Added. . . . .	667
		syntonly: Added. . . . .	686

titles: No longer required. . . . .	700	super/subscripts in a <code>lateximage</code> . . . . .	409
titleref: Prevented. . . . .	703	<code>\LWR@section</code> : Improved spacing. . .	281
xmpinl: Added. . . . .	752	<code>\LWR@stoppars</code> : Extra HTML source space after paragraphs. . . . .	261
Added. . . . .	745	<code>\fbox</code> : Fix: Uses <code>\fboxrule</code> and <code>\fboxsep</code> . . . . .	461
Docs: Horizontal space limitations. . . . .	1	<code>\framebox</code> : Fix: Handles width and horiz position. . . . .	461
Docs: Misplaced alignment character. . . . .	136	<code>\makebox</code> : Fix: Handles width and horiz position. . . . .	460
File: <code>lwarp_mathjax.txt</code> : Version change. . . . .	219	General: 2017/08/17 . . . . .	1
File: <code>README.txt</code> : updated. . . . .	1	<code>babel-french</code> : Adjustements for French variants, load order, footnotes, ellipses. . . . .	247
Fix: Added the <code>eqnarray</code> environments. . . . .	426	footnote: Extra HTML source space after paragraphs. . . . .	558
Improved font control. . . . .	465	<code>siunitx</code> : Fix for <code>babel-french</code> . . . .	445
Lists refactored to remove <code>enumitem</code> requirement. . . . .	312	<code>siunitx</code> : Improved symbol support.	670
Verbatim refactored to remove <code>fancyvrb</code> requirement. . . . .	308	transparent: Added. . . . .	725
<code>lateximage</code> : Fix: <code>lateximage</code> with <code>minipage</code> , <code>\parbox</code> , <code>\makebox</code> , <code>\fbox</code> , <code>\framebox</code> , <code>\raisebox</code> , <code>\scalebox</code> , <code>\reflectbox</code> . . . .	438	<code>upref</code> : Added. . . . .	730
<code>BlockClass</code> : Moved optional argument in front of mandatory.	253	<code>xcolor</code> : Added <code>\fcolorboxBlock</code> , <code>\colorboxBlock</code> . . . . .	738
<code>fminipage</code> : Added. . . . .	462	<code>xcolor</code> : Fix: Background none in print mode. . . . .	738
<code>LWR@nestspan</code> : Fix: Minipages, <code>BlocksClass</code> , and lists inside a span. . . . .	249	<code>xcolor</code> : Refactored <code>\LWR@colorstyle</code> . . . . .	742
<code>LWR@tabular</code> : <code>booktabs</code> : Works inside <code>lateximage</code> . . . . .	374	<code>xcolor</code> : Uses <code>\fboxrule</code> and <code>\fboxsep</code> . . . . .	738
v0.35		<code>xcolor</code> : <code>\fcolorbox</code> etc. now work inside <code>lateximage</code> . . . . .	738
General: 2017/08/08 . . . . .	1	Docs: Reorganized: Special cases and limitations. . . . .	98
Fix: <code>\textbf</code> and related. . . . .	465	Source: Improved formatting. . . .	1
v0.36		<code>lateximage</code> : Footnotes appear in regular text instead of the <code>lateximage minipage</code> . . . . .	438
<code>\LWR@HTMLsanitize</code> : Fix for <code>babel-french</code> . . . . .	435	<code>LWR@tabular</code> : Fix for <code>babel-french</code> . .	374
<code>\LWR@HTMLsanitizeexpand</code> : Fix for <code>babel-french</code> . . . . .	435	v0.37	
<code>\LWR@closeparagraph</code> : Extra HTML source space after paragraphs.	258	<code>\include</code> : Maintains independent aux files for HTML. . . . .	167
<code>\LWR@currenttextcolor</code> : Fix for <code>\rule</code> when <code>xcolor</code> not loaded.	476	General: 2017/08/19 . . . . .	1
<code>\LWR@footnotetext</code> : Extra HTML source space after paragraphs.	266	$\TeX$ accents: Added. . . . .	176
Force HTML superscripts. . . . .	266	<code>babel-french</code> : Adjustment for load order. . . . .	247
<code>\LWR@nullfonts</code> : Fix: Filenames while using <code>MATHJAX</code> . . . . .	467	color: Prevented. . . . .	515
<code>\LWR@restoreorigformatting</code> : <code>siunitx</code> : Improved		comment: Maintains independent cutfiles for print, HTML. . . . .	150
		<code>siunitx</code> : Improved symbol support.	670
		<code>textcomp</code> : Improved support. . . .	689



lwarpmk: Removes additional	Supports authblk with <div>s of
HTML aux files. . . . . 222	class oneauthor instead of
File handles reorganized. . . . . 166	tabular. . . . . 303, 710
v0.38	\AddSubtitlePublished: Added. . 304
\@seccntformat: Added for	\LWR@domulticolumn: Add: Optional
appendix. . . . . 280	vpos and # rows. . . . . 360
\ForceHTMLPage: Added. . . . . 278	\LWR@restoreorigformatting:
\ForceHTMLTOC: Added. . . . . 278	Appended with \appto instead of
\LWR@section: \part* starts a new	calling various macros. . . . . 409
HTML page, for appendix. . . . . 281	\LWR@tabledatacolumnntag: Don't
Modified spacing, uses	start a data cell if see
\numberline. . . . . 281	\TabularMacro. . . . . 370
\numberline: Added trailing \quad. 402	\ResumeTabular: Added. . . . . 368
\part: Fix with article class. . . . . 286	\TabularMacro: Added. . . . . 368
General: 2017/08/27 . . . . . 1	\multicolumnrow: Added. . . 367, 625
appendix: Added. . . . . 499	\printauthor: Removed minipages. 299
arabicfront: Added. . . . . 499	Supports authblk with <div>s of
caption2: Prevented. . . . . 512	class oneauthor instead of
chappg: Added. . . . . 514	tabular. . . . . 299
color: Forces xcolor as well. . . . . 515	\thanksmarkseries: Removed
fix2col: Added. . . . . 543	minipage footnotes. . . . . 711
fncychap: Added. . . . . 555	General: 2017/09/05 . . . . . 1
grffile: Added. . . . . 577	a4wide: Added. . . . . 485
metalogo: Added. . . . . 617	a4: Added. . . . . 485
nonumonpart: Added. . . . . 629	a5comb: Added. . . . . 485
nopageno: Added. . . . . 629	addlines: Added. . . . . 492
pagenote: Option page disabled. 643	anysize: Added. . . . . 498
realscripts: Added. . . . . 651	authblk: Added. . . . . 501
relsize: Added. . . . . 652	bigdelim: Added. . . . . 503
romanbarpagenumber: Added. . . 654	bigstrut: Added. . . . . 504
romanbar: Added. . . . . 653	chngepage: Prevented. . . . . 513
scalefnt: Added. . . . . 656	ebook: Added. . . . . 523
siunitx: Removed from lwarp core. 670	fullwidth: Added. . . . . 563
textcomp: Removed from lwarp	midpage: Added. . . . . 619
core. . . . . 689	multirow: Add: New optional vpos
tocbibind: Added. . . . . 714	argument. . . . . 623
xltxtra: Added. . . . . 751	multirow: Add: Supports left/right
lwarpmk: Added print1 and	border for bigdelim. . . . . 623
html1 actions. . . . . 222	multirow: Fix: Long text
Added \markboth, \sloppy, etc. 245	argument. . . . . 623
Docs: Enhanced <i>Supported</i>	supertabular: Added. . . . . 685
<i>Features</i> table. . . . . 54	textarea: Added. . . . . 688
Docs: Index, tocbibind. . . . . 108	titling: Improved compatibility. . 707
Docs: Starred sections. . . . . 106	titling: Removed extraneous
v0.39	center environments. . . . . 708
\@maketitle: titling version. . . . . 710	typearea: Added. . . . . 727
Native L <sup>A</sup> T <sub>E</sub> X version. . . . . 303	xtabular: Added. . . . . 752
Removed minipages. . . . . 303, 710	zwpgelayout: Added. . . . . 755
	Docs: Reorganized tabular
	discussion. . . . . 115

Titlepage \published and \subtitle removed. \AddSubtitlePublished restores. ....	304	midfloat: Added. ....	618
titlepage: Clear pending footnotes.	298	multirow: Improved bigdelim borders. ....	623
Removed minipages. ....	298	pnfnote: Added. ....	646
titlingpage: Clear pending footnotes. ....	708	quotchap: Added. ....	649
v0.40		sectsty: Added. ....	665
\@chapcntformat: Added for tocbibind, anonchap. ....	280	stabular: Added. ....	678
\LWR@HTMLhline: Added. ....	373	tabs: Added. ....	687
\LWR@includegraphicsb: Add: Full \graphicspath support. ....	570	textcomp: Additional symbols, improved XeLaTeX and LuaLaTeX support. ....	689
\LWR@nullfonts: Fix: Long arguments for expandable command. ....	467	tocbibind: Improved for \simplechapter. ....	714
\LWR@restoreorigformatting: Improved $\TeX$ logos inside a lateximage. ....	409	xltxtra: Fix for \showhyphens with XeLaTeX. ....	751
Improved symbols inside a lateximage. ....	409	No longer preloads xfrac. ....	162
Nullified \InlineClass, etc. inside a lateximage. ....	409	v0.41	
\LWR@tabledatacolumnntag: Fix for bigdelim: \ldelim, \rdelim. ..	370	\LWR@addcmidruletrim: Add: \cmidrule trims. ....	351
\chapter: Added support for quotchap. ....	286	\LWR@clearmidrules: Add: \cmidrule trims. ....	349
\multicolumnrow: Fix: Adapts to older multirow and xparse. ...	367	\LWR@closetabledatacell: Add: Mute > for \bottomrule. ....	327
\simplechapterdelim: Added for tocbibind, anonchap. ....	280	Fix: At/bang column with \multirow. ....	327
\underline: Added. ....	471	Fix: Cancel < for \multicolumn. ...	327
General: 2017/09/25 ....	1	\LWR@domulticolumn: Add: \cmidrule trims. ....	360
adjmulticol: Added. ....	491	Added vertical rules. ....	361
anonchap: Added. ....	498	\LWR@nullifyNoAutoSpacing: Fix: \NoAutoSpacing in a tabular with babel-french. ....	374
bigdelim: Improved documentation. ....	503	\LWR@parsebarcolumn: Added vertical rules. ....	335
cuted: Added. ....	518	\LWR@printatbang: Add: \cmidrule trims. ....	345
dblfnote: Added. ....	519	Add: Mute at and bang columns for \bottomrule. ....	345
fnpos: Added. ....	556	\LWR@printbartag: Added vertical rules. ....	345
graphics: Moved out of the lwarp core. ....	565	\LWR@subaddcmidruletrim: Added. ...	351
graphics: Restores \includegraphics and \DeclareGraphicsExtensions in a lateximage. ....	565	\LWR@subcmidrule: Add: \cmidrule trims. ....	350
graphicx: Moved out of the lwarp core. ....	577	\LWR@tabledatasinglecolumnntag: Add: \cmidrule trims. ....	346
grffile: Directly supported. ....	577	Add: Mute < for \bottomrule. ...	346
		\LWR@tabularfinishrow: Unfinished tabular rows automatically filled. ....	330

\mcolrowcell: Added for	
\multicolumrow cells. . . . .	372
General: 2017/10/07 . . . . .	1
multirow: Add: \cmidrule trims. . . . .	623
Added vertical rules. . . . .	624
Fix: < spec. . . . .	624
LWR@tabular: Fix: \NoAutoSpacing	
in a tabular with babel-french. . . . .	375
Improved rules. . . . .	376
v0.42	
\@ensuredmath: Improved	
\ensuremath. . . . .	419
\@textsubscript: Added. . . . .	470
\@textsuperscript: Added. . . . .	470
\LWR@HTMLhline: If FormatWP force	
explicit border. . . . .	373
\LWR@HTMLtextstyle: Added. . . . .	465
\LWR@addformatwpalignment: If	
FormatWP add explicit style for	
cell alignment. . . . .	353
\LWR@addrulewidth: If FormatWP	
force explicit border. . . . .	352
\LWR@amsmathbody: Fix: Numbering	
and naming AMS math	
environments. . . . .	437
\LWR@amsmathbodynumbered: Fix:	
Numbering and naming AMS	
math environments. . . . .	438
\LWR@domulticolumn: If FormatWP	
add cell alignment. . . . .	361
\LWR@doubledollar: If FormatWP	
print LaTeX expression. . . . .	415
Improved \ensuremath. . . . .	415
Improved line spacing with	
mathjax. . . . .	415
\LWR@floatbegin: If FormatWP add	
a text frame. . . . .	391
\LWR@floatend: If FormatWP add a	
text frame. . . . .	392
\LWR@hspace: If FormatWP add	
\quads. . . . .	475
\LWR@htmlmathlabel: If FormatWP	
print LaTeX expression. . . . .	425
\LWR@includegraphicsb: Fix:	
Filename expansion. . . . .	570
If FormatWP, use explicit size. . . . .	571
\LWR@remembertag: Fix: Numbering	
and naming AMS math	
environments. . . . .	437
\LWR@restoreorigformatting:	
Improved \ensuremath. . . . .	411
\LWR@rule: If FormatWP add \quads. . . . .	478
\LWR@subaddcmidruletrim: Opt if	
no rule given. . . . .	351
\LWR@subsingledollar: If	
FormatWP print LaTeX	
expression. . . . .	416
\LWR@tabledatasinglecolumn: If	
FormatWP add cell alignment. . . . .	347
\LaTeX: If FormatWP use explicit	
style. . . . .	480
\TeX: If FormatWP use explicit style. . . . .	480
\listoffigures: Added boolean	
WPMarkLOFT. . . . .	400
\listoftables: Added boolean	
WPMarkLOFT. . . . .	400
\marginpar: If FormatWP emulate a	
wrapfig. . . . .	270
\marginparBlock: If FormatWP	
emulate a wrapfig. . . . .	270
\tableofcontents: Added boolean	
WPMarkTOC. . . . .	399
\underline: If FormatWP, use	
explicit styles for \underline,	
etc. . . . .	471
General: 2017/10/30 . . . . .	1
\textbf and related: If FormatWP,	
use explicit styles for \textsc,	
etc. . . . .	465
algorithmicx: If FormatWP add	
\quads. . . . .	493
epigraph: If FormatWP add HTML	
styles. . . . .	527
fancybox: If FormatWP add HTML	
styles. . . . .	532
floatflt: Added width. . . . .	547
includegraphics: Fix: Class key. . . . .	569
keyfloat: If FormatWP add explicit	
HTML style. . . . .	589
moreverb: Simplified formatting of	
listings. . . . .	619
morewrites: Added. . . . .	621
multirow: If FormatWP add cell	
alignment. . . . .	624
overpic: Added. . . . .	642
realscripts: Fix for subscripts in a	
lateximage. . . . .	651
sidenotes: If FormatWP add	
explicit HTML style. . . . .	669

siunitx: Improved \ensuremath. . . . .	670	\LWR@maybeprintpendingfootnotes:	
soul: If FormatWP, add explicit		Added FootnoteDepth. . . . .	269
styles. . . . .	675	\LWR@nullfonts: Fix: Nullify dollar	
textcomp: Improved		inside filenames. . . . .	467
\interrobangdown. . . . .	689	\LWR@parsetablecols: Ignore	
wrapfig: If FormatWP add explicit		spaces in col spec. . . . .	339
HTML style. . . . .	736	\LWR@printmccoltype: Added	
Added boolean WPMarkLOFT. . . . .	171	vertical rules. . . . .	357
Added boolean WPMarkMath. . . . .	171	\LWR@restoreorigformatting: Fix:	
Added boolean		\ref in math or lateximage. . . . .	409
WPMarkMinipages. . . . .	170	Fix: Improved math inside	
Added boolean WPMarkTOC. . . . .	171	lateximages. . . . .	411
Added boolean WPTitleHeading. . . . .	171	\LWR@section: Fix: Expansion in	
Docs: Added support page. . . . .	2	comparison. . . . .	281
Docs: Improper \prevdepth. . . . .	136	Fix: Math in section name. . . . .	283, 285
Docs: Reorganized math		Fix: Nullify fonts inside HTML	
limitations . . . . .	109	comment. . . . .	282
File: lwarp_mathjax.txt:		\TabularMacro: \newcommand	
Updated siunitx script. . . . .	219	instead of \relax to fix	
Fix: Numbering and naming AMS		supertabular and xtab. . . . .	368
math environments. . . . .	436	\href: Made robust. . . . .	388
If FormatWP, shift section		\nameref: Made robust. . . . .	386
headings. . . . .	172	\nolinkurl: Made robust. . . . .	388
tabbing: Added. . . . .	310	\url: Made robust. . . . .	388
lateximage: Fix: Numbering and		General: 2017/11/08 . . . . .	1
naming AMS math		breakurl: Added. . . . .	507
environments. . . . .	439	hyperref: Made robust. . . . .	581–583
center: If FormatWP use explicit		hyperref: \Gauge added. . . . .	584
text-align. . . . .	443	luatodonotes: Added. . . . .	602
minipage: Added boolean		todonotes: Added. . . . .	723
WPMarkMinipages. . . . .	458, 459	LWR@currentautosec: Added. . . . .	281
If FormatWP add a text frame. . . . .	457	Added FootnoteDepth. . . . .	265
eqnarray: Fix: Numbering and		Docs: HTML settings table. . . . .	87
naming AMS math		Docs: Reorganized HTML	
environments. . . . .	427	customization. . . . .	87
If FormatWP print LaTeX			
expression. . . . .	426	v0.44	
equation: If FormatWP print LaTeX		\@currentlabelname: Adjustment	
expression. . . . .	423	for koma-script. . . . .	379
LWR@BlockClassWP: Added to factor		\HTMLTitle: Added. . . . .	264
code. . . . .	254	\LWR@addformatwpalignment: Fix	
LWR@figcaption: If FormatWP forces		for multicolumn alignment if	
italic captions. . . . .	394	FormatWP. . . . .	353
LWR@tabular: If FormatWP force		\LWR@backgroundcolor: Added. . . . .	742
explicit border. . . . .	376	\LWR@filestart: Add \HTMLTitle. . . . .	290
		Fix \HTMLAuthor. . . . .	289
v0.43		\LWR@listitem: Added list and	
\LWR@domulticolumn: Fix for		trivlist. . . . .	315
vertical rules. . . . .	361	\LWR@patchlists: Added list and	
Fix: Multicolumn trim. . . . .	360, 361	trivlist. . . . .	318

\LWR@strresult: Fix:	
\providecommand. ....	325
\addcontentsline: Automatic	
\LWR@newfloatanchor. ....	396
\chapter: Add preamble for	
koma-script. ....	286
\marginparBlock: Added. ....	270
\nopagecolor: xcolor: Fix for	
\nopagecolor. ....	743
\part: Add preamble for	
koma-script. ....	286
\title: Added \thetitle. ....	263
General: 2017/11/22 .....	1
algorithmx: Improved comment	
symbol. ....	493
atbegshi: Added. ....	500
cancel: Added. ....	508
changepage: Additional options.	513
easy-todo: Added. ....	522
fancyref: Added. ....	536
fixmetodonotes: Added. ....	545
fixme: Added. ....	544
fontenc: Allowed after lwarp. ...	556
hang: Added. ....	577
ifoddpage: Added. ....	587
ltxtable: Added. ....	602
luatodonotes: Improved. ....	602
lwarp-patch-komascript: Added. .	757
overpic: Fix: Groups for	
lateximages. ....	642
pdfsync: Fixes. ....	646
preview: Added. ....	648
scrextend: Added. ....	657
scrhack: Added. ....	660
scrlayer-notecolumn: Added. ...	662
scrlayer-scrpage: Added. ....	662
scrlayer: Added. ....	660
section: Added. ....	663
soulpos: Added. ....	677
soulutf8: Added. ....	678
supertabular: Fix for caption. ...	686
tocbasic: Added. ....	712
tocloft: Added \newlistentry. .	720
tocloft: Improved \newlistof. .	720
tocstyle: Added. ....	721
todonotes: Improved. ....	723
todo: Added. ....	722
typearea: Added expert	
commands. ....	727
watermark: Added. ....	736
xcolor: Added	
\LWR@currenttextcolorstyle. 741	
xcolor: Added	
\LWR@findcurrentcolor. ...	741
xcolor: Added	
\LWR@textcurrentcolor. ...	741
xtab: Fix for caption. ....	753
Adjustment for koma-script. 157, 159	
AMS environments: Fix: Groups	
for lateximages. ....	428
If pdfLaTeX, require T1 and UTF8	
encoding. ....	144
picture: overpic: Fix: Groups for	
lateximages. ....	454
tikzpicture: tikz: Fix: Groups for	
lateximages. ....	699
list: Added list and trivlist. .	316
LWR@nestspan: Added list and	
trivlist. ....	249
v0.45	
\@currentHref: Added. ....	387
\@donoparitem: Modified for HTML. 313	
\@item: Modified for HTML. ....	313
\@mklab: Modified for HTML. ....	313
\CSSFilename: Improved filenames	
with underscores. ....	263
\LWR@LwarpStart: Fix: Lateximages	
on incorrect pages with Mathjax. 293	
\LWR@filenamenoblanks: Fix:	
Section names with \ . ....	273
Fix: Section names with	
underscores. ....	273
\LWR@includegraphicsb: Improved	
URLs with underscores. ....	570
\LWR@newautoidanchor: Fix: No	
anchor if frozen autoid. ....	392
\LWR@notmemoirloadafter: Added. 152	
\LWR@printpendingmpfootnotes:	
Added. ....	269
\LWR@startref: Fix: Labels with	
underscores. ....	385
\LWR@subhyperref: Improved URLs	
with underscores. ....	387
\LWR@subhyperrefclass: Improved	
URLs with underscores. ....	387
\LWR@sublabel: Fix: Labels with	
underscores. ....	383
\LWR@tabledatacolumnntag: Fix:	
Empty line between rows. ....	372

<code>\chapter</code> : Add optional heading title for memoir. . . . .	286	Docs: Improved MiKTeX install instructions. . . . .	63
<code>\newpage</code> : Added. . . . .	472	Docs: Moved table so doesn't interfere with install docs. . . . .	62
<code>\nolinkurl</code> : Improved URLs with underscores. . . . .	388	File: <code>lwarp_mathjax.txt</code> : Allow MATHJAX inside tabbing. . . . .	219
<code>\normalmarginpar</code> : Added. . . . .	271	File: <code>lwarp_mathjax.txt</code> : Allow MATHJAX inside verse. . . . .	219
<code>\reversemarginpar</code> : Added. . . . .	271	Fix: Empty sidetoc. . . . .	399
<code>\section</code> : Add optional heading title for memoir. . . . .	287	Improved filenames with underscores. . . . .	581
<code>\tableofcontents</code> : Fix: Empty sidetoc. . . . .	399	Improved URLs with underscores. . . . .	507
Fix: Patch <code>\AtBeginDocument</code> . . . . .	399	Improved: Robust <code>\</code> , <code>\</code> , and <code>\textellipsis</code> commands. . . . .	471
<code>\url</code> : Improved URLs with underscores. . . . .	388	Separate LWR@thisautoidWP for word processor <code>&lt;div&gt;s</code> . . . . .	392
General: 2018/01/14 . . . . .	1	<code>thebibliography</code> : Patched to emphasize titles. . . . .	408
array: Added. . . . .	500	<code>minipage</code> : Fix: Improper <code>\prevdepth</code> . . . . .	456, 459
babel-french: Robust commands. . . . .	247		
backref: Added. . . . .	502	v0.46	
changebar: Added. . . . .	512	<code>\LWR@closeparagraph</code> : Fix: Tabular empty lines. . . . .	259
cite: Added. . . . .	514	<code>\LWR@closeprevious</code> : Fix: Stack unnesting. . . . .	245
continue: Added. . . . .	517	<code>\LWR@forcenewpage</code> : Fix: Improper <code>\prevdepth</code> . . . . .	245
endfloat: Added. . . . .	524	<code>\LWR@lookforpackagename</code> : Fix: Spaces in <code>\usepackage</code> . . . . .	164
epigraph: Support for memoir. . . . .	528	<code>\LWR@providelength</code> : Added. . . . .	145
fancyvrb: Improvements. . . . .	537	<code>\LWRPrintStack</code> : Name changed from <code>\PrintStack</code> . . . . .	244
flafter: Added. . . . .	545	<code>\makebox</code> : Fix: Lateximage in a <code>\makebox</code> . . . . .	460, 461
fltrace: Added. . . . .	554	<code>\popclose</code> : Fix: Stack unnesting. . . . .	238
footnpag: Added. . . . .	559	<code>\pushclose</code> : Fix: Stack unnesting. . . . .	237
fwlw: Added. . . . .	563	General: 2018/01/23 . . . . .	1
hanging: Added. . . . .	579	LWR@tabularpardepth added. . . . .	325
hyperref: Fix for underscores in <code>\hyperref</code> . . . . .	581	amsthm: Adapted to <code>trivlist</code> changes. . . . .	496
lwarp-patch-memoir: Added. . . . .	759	mdframed: Fixes for svg math or lateximage in title. . . . .	610
memhfixc: Added. . . . .	617	mdframed: Fixes for footnotes. . . . .	611
memoir: Added. . . . .	483	ntheorem: Adapted to <code>trivlist</code> changes. . . . .	630
natbib: Added. . . . .	626	theorem: Adapt to <code>trivlist</code> changes. . . . .	696, 697
pagesel: Added. . . . .	644	list: Fix: Stack unnesting. . . . .	316
prettyref: Added. . . . .	648	LWR@tabular: Fix: Tabular empty lines. . . . .	378
subfigure: Added. . . . .	684		
subfig: Fix for subcaption end tag. . . . .	682		
subfig: Fix: Math in subcaptions. . . . .	680		
textfit: Added. . . . .	692		
titleref: Added. . . . .	703		
turnthepage: Added. . . . .	726		
Allows memoir's preloaded packages. . . . .	153		
Docs: xparse warnings. . . . .	124		
Docs: Fix for double hyphens. . . . .	67		
Docs: Improved install instructions. . . . .	67		

v0.47	
\LWR@LwarpStart: Fix for SVG math in \nameref. . . . .	294
\LWR@WPcell: Fix: Line wrap at HTML hyphen. . . . .	353
\LWR@caption@begin: Fix: Argument passed to \LWR@origcaption@begin. . .	394
\LWR@createautosec: Fix: Line wrap at HTML hyphen. . . . .	279
\LWR@domulticolumn: Fix: Line wrap at HTML hyphen. . . . .	360
\LWR@floatbegin: Fix: Line wrap at HTML hyphen. . . . .	391
\LWR@htmlclosecomment: Add \mbox to prevent line breaks. .	250
\LWR@newautoidanchor: Fix: Line wrap at HTML hyphen. . . . .	392
\LWR@printopenlist: Fix: Line wrap at HTML hyphen. . . . .	313
\LWR@startref: Fix: Line wrap at HTML hyphen. . . . .	384
\LWR@sublabel: Fix: Line wrap at HTML hyphen. . . . .	383
\LWR@subsingledollar: Added SVG math image baseline adjust and em sizing. . . . .	416
Fix: Line wrap at HTML hyphen. .	418
\captionlistentry: Fix: Line wrap at HTML hyphen. . . . .	395
\hypertoc: Fix: Line wrap at HTML hyphen. . . . .	403
\hypertocfloat: Fix: Line wrap at HTML hyphen. . . . .	404
\makebox: Fix: Line wrap at HTML hyphen. . . . .	460
General: 2018/01/30 . . . . .	1
adjmulticol: Fix: Line wrap at HTML hyphen. . . . .	491
blowup: Added. . . . .	505
caption: Added. . . . .	509
caption: Also loads lwarp-caption. .	166
changepage: Fix for pagecheck macros. . . . .	513
endheads: Added. . . . .	524
epigraph: Fix: Line wrap at HTML hyphen. . . . .	527
hanging: Fix: Line wrap at HTML hyphen. . . . .	579
hang: Fix: Line wrap at HTML hyphen. . . . .	577
keyfloat: Fix for SVG math in captions. . . . .	588
midpage: Fix: Line wrap at HTML hyphen. . . . .	619
multirow: Fix: Line wrap at HTML hyphen. . . . .	623
multitoc: Added. . . . .	626
ntheorem: Fix: Line wrap at HTML hyphen. . . . .	635
realscripts: Fix: Line wrap at HTML hyphen. . . . .	651
scrextend: Fix: Line wrap at HTML hyphen. . . . .	657
sectionbreak: Added. . . . .	664
sidenotes: Fix for SVG math in captions. . . . .	669
subfig: Fix for SVG math in captions. . . . .	680
subfig: Fix: Support \nameref. . .	679
xurl: Added. . . . .	754
lwarpmk: pdfcrop: Removed hires option for improved crop accuracy. . . . .	222
lateximage: Added CSS style option. .	438
Fix: Line wrap at HTML hyphen. .	441
center: Fix: Line wrap at HTML hyphen. . . . .	443
minipage: Fix: Line wrap at HTML hyphen. . . . .	457
flushleft: Fix: Line wrap at HTML hyphen. . . . .	443
flushright: Fix: Line wrap at HTML hyphen. . . . .	443
enumerate: Fix: Line wrap at HTML hyphen. . . . .	317
itemize: Fix: Line wrap at HTML hyphen. . . . .	317
LWR@BlockClassWP: Fix: Line wrap at HTML hyphen. . . . .	254
v0.48	
\@@@setcpageref: Fix for new v0.21 of cleveref. . . . .	453
\@@@setcref: Fix for new v0.21 of cleveref. . . . .	452
\@@@setcrefrange: Fix for new v0.21 of cleveref. . . . .	452
\@biblabel: Improved bibliography label. . . . .	408

<code>\@item: Honors \makelabel.</code> . . . .	313	<code>\rightline: Added.</code> . . . . .	444
<code>\@maketitle: Fix: Errors with</code>		<code>\thempfootnote: Removed</code>	
<code>IEEEtran class.</code> . . . . .	303	<code>\itshape.</code> . . . . .	268
<code>\LWR@LwarpStart: Adjusted space</code>		General: 2018/02/14 . . . . .	1
<code>around captions.</code> . . . . .	292	<code>acronym: Added.</code> . . . . .	490
<code>\LWR@ProvidesPackageDrop: Fix:</code>		<code>acro: Added.</code> . . . . .	488
<code>Options with braces.</code> . . . . .	165	<code>chapterbib: Added.</code> . . . . .	514
<code>\LWR@addtabularhrulecolor:</code>		<code>colortbl: Added.</code> . . . . .	342, 353, 515
<code>colortbl: Added.</code> . . . . .	354	<code>fancyheadings: Prevented.</code> . . . . .	535
<code>\LWR@addtabularrulecolors:</code>		<code>fancyref: Now directly supported.</code>	536
<code>colortbl: Added.</code> . . . . .	354	<code>hypcap: Added.</code> . . . . .	580
<code>\LWR@closetabledatcell:</code>		<code>hypernat: Added.</code> . . . . .	580
<code>colortbl: Added.</code> . . . . .	328	<code>hyperref: \texorpdfstring now</code>	
<code>\LWR@includegraphicsb: Fix:</code>		<code>uses the T<sub>E</sub>X string.</code> . . . . .	583
<code>Virtual page size limited to a</code>		<code>luatodonotes: Improved</code>	
<code>group.</code> . . . . .	570, 573	<code>\todotoc.</code> . . . . .	603
<code>\LWR@lookforpackagename: Fix:</code>		<code>siunitx: Changes fraction to</code>	
<code>Parsing similar package names.</code>	163	<code>symbol.</code> . . . . .	672
<code>\LWR@newautopagelabel: Fix: TOC,</code>		<code>siunitx: Improved svg math.</code>	670, 672
<code>LOF, LOT links.</code> . . . . .	274	<code>siunitx: Improved color output.</code>	671
<code>\LWR@newhtmlfile: Fix: TOC, LOF,</code>		<code>stfloats: Added.</code> . . . . .	679
<code>LOT links.</code> . . . . .	276	<code>todonotes: Improved</code>	
<code>\LWR@nullfonts: Fix: \newline in</code>		<code>\todotoc.</code> . . . . .	723
<code>title.</code> . . . . .	467	<code>vmargin: Added.</code> . . . . .	732
<code>\LWR@parsedrequirepackagenames:</code>		<code>xfrac: Fix: Added groups around</code>	
<code>Fix: Parsing similar package</code>		<code>super/subscripts to localize</code>	
<code>names.</code> . . . . .	163	<code>LWR@nestspan changes.</code> . . . . .	749
<code>\LWR@parsetablecols: Fix: Ignore</code>		<code>Docs: Converting an existing</code>	
<code>optional tabular column</code>		<code>document.</code> . . . . .	82
<code>arguments.</code> . . . . .	340	<code>Improved font control.</code> . . . .	468, 469
<code>\LWR@restoreorigformatting: Fix:</code>		<code>lateximage: Print mode boxed to</code>	
<code>\mbox in svg math, lateximage,</code>		<code>natural width.</code> . . . . .	442
<code>Tikz.</code> . . . . .	411	<code>abstract: Allow optional name.</code>	306
<code>Fix: Font sizes in svg math,</code>		<code>LWR@tabular: colortbl: Added.</code>	375
<code>lateximage, Tikz.</code> . . . . .	409		
<code>Fix: Spacing in svg math,</code>		<code>\LWR@addtabularcellcolor: xcolor:</code>	
<code>lateximage, Tikz.</code> . . . . .	410	<code>Added tabular row colors.</code> . . . .	355
<code>\LWR@section: Fix: TOC, LOF, LOT</code>		<code>\LWR@domulticolumn: xcolor: Added</code>	
<code>links.</code> . . . . .	285	<code>tabular row colors.</code> . . . . .	361
<code>\LWR@tabledatasinglecolumn: tag:</code>		<code>\LWR@printlength: Fix: Group</code>	
<code>colortbl: Added.</code> . . . . .	347	<code>printlen changes.</code> . . . . .	162
<code>\LWR@textcurrentfont: Added.</code>		<code>\LWR@subsingledollar: Fix:</code>	
<code>Improves font control.</code> . . . . .	469	<code>Incorrect PDF output size</code>	
<code>\bibliography: Fix: \BaseJobname</code>		<code>corrupted HTML.</code> . . . . .	417
<code>for bibliography.</code> . . . . .	408	<code>\affiliation: Fix: Adapts to classes</code>	
<code>\centerline: Added.</code> . . . . .	444	<code>which already provide.</code> . . . . .	298
<code>\l@part: Adapts to classes without</code>		<code>\href: Fix: Adapt to classes.</code> . . . .	388
<code>\part.</code> . . . . .	404	<code>\noalign: Fix: \noalign inside</code>	
<code>\leftline: Added.</code> . . . . .	443	<code>tabular.</code> . . . . .	373
<code>\mbox: Nullified for HTML.</code> . . . . .	460	<code>\url: Fix: Adapt to classes.</code> . . . . .	388



General: 2018/02/19	1	\vspace: Robustify macros.	476
amsmath: Fix: Patches for		General: 2018/03/03	1
\eqref.	162	lwarp.css: Improved svg display	
eso-pic: Fix for		math centering.	180
\AddToShipoutPicture.	529	lwarp_one_limage.txt: Added.	219
figsize: Added.	543	amsmath: Fix: Upright tags for	
fnlineno: Added.	555	svgmath.	162
hypdestopt: Added.	580	axodraw2: Added.	502
hyphenat: Added.	585	bytefield: Added.	507
lineno: Added.	592	dblfloatfix: Added.	519
luacolor: Added.	602	diagbox: Added.	520
pagegrid: Added.	643	epstopdf: Added.	528
pdfrender: Added.	646	listings: Force flexible columns.	595
resizegather: Added.	653	morefloats: Added.	619
vertbars: Added.	732	nonfloat: Added.	629
vwcol: Added.	733	ntheorem: Fix: Not standard nor	
xcolor: Added tabular row		amsthm selected.	637
colors.	342, 747	pbox: Added.	645
Fix: Adapt to classes.	472	phfqit: Added.	647
v0.50		schemata: Added.	656
\@ensuredmath: Fix: Use		siunitx: Fix: Loads xcolor.	670
lateximage even if MathJax.	419	siunitx: Improved svg math alt	
Improved svg math alt tags.	419	tags.	672
\LWR@atbeginverbatim: Improved		siunitx: Improved units.	445, 670, 673
column alignment.	309	xy: Added.	754
\LWR@doubledollar: Improved svg		lwarpmk: Error if	
math alt tags.	416	lateximages.txt does not	
Improved svg math display.	416	exist.	222
\LWR@footnotetext: Robustify		lwarpmk: Error if lwarpmk.conf	
macros.	267	points to lwarp.	222
\LWR@htmlrefsectionfilename:		lwarpmk: Improved error	
Fix: SVG math in a section name.	243	messages.	222
\LWR@newhtmlfile: Fix: SVG math		lwarpmk: MD5 hash avoids	
in a section name.	275	duplicate svg math.	222
\LWR@nullfonts: Fix: \underline		lwarpmk: Multiprocess support	
in sectioning file name.	468	making lateximages.	222
Robustify macros.	468	AMS environments: Improved svg	
\LWR@overline: Added.	471	math display.	428
\LWR@restoreorigformatting: Fix:		Fix: \centering, etc. in svg math,	
\centering, etc. in svg math,		lateximage, Tikz.	175
lateximage, Tikz.	410	Fix: Load fontspec if necessary.	158
\LWR@subsingledollar: Fix: Use		Robustify macros.	469
lateximage even if MathJax.	416	lateximage: Fix: SVG math in a	
Improved svg math alt tags.	416	section name.	441
MD5 hash avoids duplicate svg		MD5 hash avoids duplicate svg	
math.	418	math.	439, 441
\newline: Robustify macros.	472	eqnarray: Improved svg math	
\rule: Robustify macros.	479	display.	427, 428
\textsubscript: Robustify macros.	470	equation: Improved svg math	
\textsuperscript: Robustify		display.	423
macros.	470		

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	
\\$	415
\&	240, 5737
\(	7713
\)	7713
\,	98
\@@@setcpageref	8537
\@@@setcref	8500
\@@@setcrefrange	8514
\@author	296
\@begintheorem	5397
\@biblabel	7454
\@capttype	7122
\@chapcntformat	4716
\@currentHref	7018
\@currentlabelname	6879
\@date	296
\@dlbfloat	7094
\@donoparitem	5423
\@endtheorem	5409
\@ensuredmath	7717
\@float	7094
\@fnsymbol	5202
\@footnotetext	4414
\@include	610
\@item	5436
\@makecaption	7124
\@makefnmark	4385
\@makefnmark	4384
\@maketitle	49, 5225
\@mklab	5417
\@mpfootnotetext	4416
\@nbitem	5503
\@opargbegintheorem	5403
\@rowc@lors	5983
\@rowcolors	5982
\@seccntformat	4714
\@starttoc	7250
\@textsubscript	8955
\@textsuperscript	8951
\@title	296
\@wrglossary	7431
\@wrindex	7422
\\	473
\$	415
\$\$	415
~	98
<b>A</b>	
a4 (package)	485
a4wide (package)	485
a5comb (package)	485
abstract (environment)	5291
abstract (package)	106, 485
\abstractname	89, 5290
accents	277
acro (package)	488
acronym (package)	489
adapting	
class	134
document	82
package	133
\addcontentsline	7198
addlines (package)	492
\AddSubtitlePublished	5253
adjmulticol (package)	491
Adobe (program)	59
\affiliation	5105
afterpackage (package)	159
afterpage (package)	492
algorithmic	
with newfloat, trivfloat	726
algorithmicx (package)	492
algorithmx (package)	120
align (environment)	8003
align* (environment)	8027
alignment tab character &, misplaced	116, 320, 328
alltt (package)	494
\AmS	9169
AMSMath	
split miss-numbered	112, 630
amsmath (package)	161
amsthm (package)	494
\and	296
anonchap (package)	498

anysize (package) .....	498	LWR@emptyatbang .....	324
appendix (package) .....	107, 499	LWR@exitingtabular .....	369
arabicfront (package) .....	499	LWR@freezethisautoid .....	392
array (package) .....	500	LWR@infloatrow .....	569
\arrayrulecolor .....	5992	LWR@intabularmetadata .....	325
AsciiDoc (program) .....	59	LWR@isstartingequation .....	436
AsciiDoctor (program) .....	59	LWR@minipagefullwidth .....	456
Asciidoctor-LaTeX (program) .....	59	LWR@minipagethispar .....	456
atbegshi (package) .....	500	LWR@opttablecol .....	337
\attrib .....	122, 307, 730	LWR@skipatbang .....	324
\attribution .....	5301	LWR@skippingmcolrowcell .....	324
authblk (package) .....	107, 501	LWR@skippingmrowcell .....	324
author		LWR@starredlongtable .....	362
HTML meta tag .....	97, 264	LWR@startedrow .....	323
\author .....	96, 296	LWR@tableparcell .....	324
autosec .....	279	LWR@tabularcelladded .....	324
axodraw2 (package) .....	502	LWR@tabularmutemods .....	369
<b>B</b>		LWR@tracinglwarp .....	169
babel (package) .....	123	LWR@validtablecol .....	337
\backmatter .....	4695	LWR@verbtags .....	308
backref (package) .....	502	mathjax .....	147
balance (package) .....	502	usingOSWindows .....	146
\BaseJobname .....	3786	warpingHTML .....	147
BaseJobname (option) .....	86, 148	warpingprint .....	147
baseline		WPMarkFloats .....	129, 170
tabular .....	337	WPMarkLOFT .....	130, 171
\bfseries .....	8940	WPMarkMath .....	130, 171
\bibliography .....	7447	WPMarkMinipages .....	129, 170
\BibTeX .....	9164	WPMarkTOC .....	129, 171
bigdelim (package) .....	119, 503	WPTitleHeading .....	130, 171
bigstrut (package) .....	504	boxedminipage (package) .....	506
BlockClass (environment) .....	4123	boxedminipage2e (package) .....	506
\BlockClassSingle .....	4134	\boxframe .....	158
blowup (package) .....	505	breakurl (package) .....	507
bookmark (package) .....	505	bugs .....	136
booktabs (package) .....	505	BVerbatim (environment) .....	197
boolean:		bytefield (package) .....	507
CombineHigherDepths .....	89, 271	<b>C</b>	
FileSectionNames .....	89, 241	calc (package) .....	161
FormatEPUB .....	125, 170	Calibre .....	125
FormatWP .....	127, 170	cancel (package) .....	508
HTMLDebugComments .....	169	capt-of (package) .....	396
LWR@amsmultline .....	424	caption (package) .....	120, 166, 396, 509
LWR@copiedsidetoc .....	399	caption2 (package) .....	512
LWR@doingapar .....	256	\caption@begin .....	7179
LWR@doingcmidrule .....	324	\caption@end .....	7179
LWR@doinghline .....	324	\captionlistentry .....	7183
LWR@doingstartpars .....	256	\captionof .....	7212
LWR@doingtbrule .....	324	\CaptionSeparator .....	7123



document		flushright	8327
convert existing	82	fminipage	8766, 8813
\dotfill	8996	gather	7955
\doublerulesepcolor	5994	gather*	7979
draftwatermark (package)	522	itemize	5553
<b>E</b>		lateximage	8209, 8314
easy-todo (package)	522	list	5523
ebook (package)	523	longtable	2
ellipsis (package)	523	LWR@BlockClassWP	4151
\emph	8822	LWR@figcaption	7138
emptypage (package)	524	LWR@nestspan	3987
\end@dlbfloat	7110	LWR@tabular	6746
\end@float	7110	LWR@print@fminipage	8789
\EndDefiningTabulars	5731	math	7737
endfloat (package)	524	minipage	8619
endheads (package)	524	multline	7904
endnotes (package)	107, 525	multline*	7929
\enlargethispage	9065	picture	8600
\enskip	473, 9019	quote	5316
enumerate (environment)	5559	tabbing	5385
enumerate (package)	526	thebibliography	7455
enumitem (package)	526	theindex	7400
environ (package)	161	tikzpicture	7
environment:		titlepage	5122
fminipage	101	titlingpage	14
lateximage	433	verbatim	5375
picture	454	verse	2
titlepage	95	warpall	139
titlingpage	96	warpHTML	89, 140
warpall	94	warpprint	140
warpHTML	91, 94	epigraph (package)	527
warpprint	91, 94	EPS images	113, 446
environments:		epstopdf (package)	113, 115, 446, 528
abstract	5291	Epub	
align	8003	conversion software	125
align*	8027	HTML conversion settings	125, 170
BlockClass	4123	eqnarray (environment)	7850
BVerbatim	197	equation	
center	8319	miss-numbered	112, 630
description	5578	equation (environment)	7785
displaymath	7738	equation* (environment)	7813
enumerate	5559	error messages	136
eqnarray	7850	eso-pic (package)	529
equation	7785	etoolbox (package)	145
equation*	7813	everyhook (package)	160
fcolorminipage	143, 8438	everypage (package)	529
flalign	8051	everyshi (package)	530
flalign*	8075	expl3 (package)	160
flushleft	8335	export	
		to word processor	127



G	
gather (environment)	<u>7955</u>
gather* (environment)	<u>7979</u>
GELLMU (program)	<u>58</u>
generator	
HTML meta tag	<u>289</u>
geometry (package)	<u>159, 563</u>
getttitlestring (package)	<u>160</u>
[Gin]:	
class (key)	<u>569</u>
\Gin@setfile	<u>96</u>
GladTeX (program)	<u>58</u>
glossaries (package)	<u>107, 564</u>
glossary	
language	<u>80</u>
processing	<u>80</u>
glyphtounicode (file)	<u>84</u>
graphics	
EPS files	<u>113, 446</u>
PDF files	<u>113, 445</u>
graphics (package)	<u>113, 565</u>
graphicx (package)	<u>113, 576</u>
grffile (package)	<u>114, 577</u>
H	
hang (package)	<u>577</u>
hanging (package)	<u>579</u>
heading, word processor	<u>130</u>
Hevea (program)	<u>58</u>
\hfill	<u>8994</u>
\HomeHTMLFilename	<u>3788</u>
HomeHTMLFilename (option)	<u>86, 88, 149</u>
horizontal space	<u>98</u>
between minipages	<u>473</u>
\href	<u>7050</u>
\hrulefill	<u>8995</u>
\hskip	<u>98</u>
\hspace	<u>98, 473, 9056</u>
htlatex (program)	<u>58</u>
HTML	
commands for a successful conver-	
sion	<u>98</u>
conversion settings	<u>87</u>
debug comments	<u>169</u>
EPUB	<u>125, 170</u>
word processor	<u>127, 170</u>
filename generation	<u>91</u>
headings	<u>140</u>
in place of SVG images	<u>75, 138</u>
meta tag	
author	<u>97, 264</u>
description	<u>96, 264</u>
generator	<u>289</u>
Title	<u>264</u>
title	<u>97, 263</u>
viewport	<u>290</u>
tabular column conversion	<u>338</u>
\HTMLAuthor	<u>90, 97, 4366</u>
HTMLDebugComments (boolean)	<u>169</u>
\HTMLDescription	<u>90, 96, 4371</u>
\HTMLentity	<u>3769</u>
\HTMLFilename	<u>3787</u>
HTMLFilename (option)	<u>86, 88, 149</u>
\HTMLFirstPageTop	<u>90, 4328</u>
htmlglossary (option) [lwarpmk]	<u>108, 564</u>
\HTMLLanguage	<u>90, 4947</u>
HTMLleftmargini (length)	<u>122, 307, 308, 730</u>
\HTMLPageBottom	<u>90, 297, 4334</u>
\HTMLPageTop	<u>90, 4331</u>
\HTMLTitle	<u>90, 97, 4361</u>
\HTMLunicode	<u>3777</u>
HTMLvleftskip (length)	<u>122, 307, 308, 730</u>
hypcap (package)	<u>580</u>
hypdestopt (package)	<u>580</u>
\hyperindexref	<u>7438</u>
hypernat (package)	<u>580</u>
hyperref (package)	<u>105, 387, 581</u>
\hypertoc	<u>7340</u>
\hypertocfloat	<u>7360</u>
hyperxmp (package)	<u>585</u>
hyphenat (package)	<u>585</u>
I	
icon	
warning	<u>142</u>
idxlayout (package)	<u>586</u>
\if@titlepage	<u>5100</u>
ifoddpage (package)	<u>587</u>
ifplatform (package)	<u>145</u>
images	
EPS	<u>113, 446</u>
graphicx package	<u>445</u>
in strange places	<u>137</u>
PDF	<u>113, 445</u>
showing as HTML	<u>75, 138</u>
Improper \prevdepth.	<u>136</u>
\include	<u>610</u>
\includegraphics	<u>225</u>
indentfirst (package)	<u>587</u>





lotdepth (counter) .....	403	LWR@amsmultline (boolean) .....	424
lscap (package) .....	601	\LWR@atbeginverbatim .....	5343
ltcaption (package) .....	601	\LWR@backgroundcolor .....	48
ltxttable (package) .....	602	\LWR@beginhideamsmath .....	7835
luacolor (package) .....	602	\LWR@blackborderpadding .....	8745
LuaLaTeX		LWR@BlockClassWP (environment) ..	4151
detection .....	143	\LWR@botnavigation .....	3849
file & section names .....	277	\LWR@caption@begin .....	7158
\LuaLaTeX .....	9155	\LWR@caption@end .....	7170
LuaLaTeX (program) [requirement] ..	62	LWR@cellcolordepth (counter) .....	355
\LuaTeX .....	9155	\LWR@cellHTMLcolor .....	5987
luatodonotes (package) .....	123, 602	\LWR@clearmidrules .....	6141
lwarp		\LWR@closeparagraph .....	4232
loading .....	85	\LWR@closeprevious .....	3876
options .....	85	\LWR@closetabledatcell .....	5687
lwarp (package) .....	85	\LWR@cmidrulewidth (length) .....	349
lwarp-patch-komascript (package) ..	756	LWR@colafterspec .....	326
lwarp-patch-memoir (package) .....	758	LWR@colatspec .....	326
lwarp.css (file) .....	93, 180	LWR@colbangspec .....	326
lwarp.xdy		LWR@colbarspec .....	326
customizing .....	97	LWR@colbeforespec .....	326
lwarp.xdy (file) .....	97, 218	\LWR@columnHTMLcolor .....	5985
lwarp_formal.css (file) .....	213	\LWR@convertto .....	60
lwarp_mathjax.txt (file) .....	219	LWR@copiedsidetoc (boolean) .....	399
lwarp_one_limage.cmd (file) .....	219	\LWR@copyfile .....	645
lwarp_sagebrush.css (file) .....	208	\LWR@createautosec .....	4701
lwarp_tutorial.txt (file) .....	69	LWR@currentautosec (counter) .....	281
lwarpmk		\LWR@currentcss .....	4339
customizing .....	135	\LWR@currenttextcolor .....	9068
[lwarpmk]:		\LWR@descitem .....	5565
htmlglossary (option) .....	108, 564	\LWR@docmidrule .....	6172
printglossary (option) .....	108, 564	LWR@doingapar (boolean) .....	256
lwarpmk (option) .....	86, 148	LWR@doingcmidrule (boolean) .....	324
lwarpmk (program) .....	135, 222	LWR@doinghline (boolean) .....	324
lwarpmk.conf (file) .....	178	LWR@doingstartpars (boolean) .....	256
lwarpmk.lua (file) .....	135	LWR@doingtbrule (boolean) .....	324
\LWR@addcmidruletrim .....	6204	\LWR@domulticolumn .....	6417
\LWR@addcmidrulewidth .....	6235	\LWR@doubledollar .....	7606
\LWR@addformatwppalignment .....	6242	LWR@emptyatbang (boolean) .....	324
\LWR@addleftmostbartag .....	6066	\LWR@endhideamsmath .....	7843
\LWR@addmathjax .....	7774	\LWR@endoffline .....	9002
\LWR@addmulticolvertrulecolor ..	6403	\LWR@ensuredoingapar .....	4210
\LWR@addrulewidth .....	6209	\LWR@equationtag .....	8168
\LWR@addtabularcellcolor .....	6306	LWR@exitingtabular (boolean) .....	369
\LWR@addtabularhrulecolor .....	6270	LWR@externalfilecnt (counter) .....	415
\LWR@addtabulararrowcolor .....	6255	LWR@figcaption (environment) .....	7138
\LWR@addtabularrulecolors .....	6282	\LWR@filenamenoblanks .....	4518
\LWR@afterendverbatim .....	5359	\LWR@filestart .....	4954
\LWR@amsmathbody .....	8196	\LWR@findword .....	521
\LWR@amsmathbodynumbered .....	8202	\LWR@floatbegin .....	7073

\LWR@floatend .....	7096	LWR@isstartingequation (boolean) ..	436
\LWR@footnotetext .....	4386	\LWR@itemizeitem .....	5546
\LWR@forceminwidth .....	8734	LWR@latestautopage (counter) .....	394
\LWR@forcenewpage .....	3893	LWR@lateximagedepth (counter) .....	434
LWR@freezethisautoid (boolean) ..	392	\LWR@lateximagedepthref .....	6909
\LWR@futurenonspacelot .....	5633	LWR@lateximagenumber (counter) ..	434
\LWR@FVstyle .....	38	\LWR@lateximagenumberref .....	6912
\LWR@getexparray .....	3755	\LWR@lateximagesfile .....	607
\LWR@getmynexttoken .....	5640	\LWR@lightrulewidth (length) .....	349
\LWR@heavyrulewidth (length) .....	349	LWR@Lipage (counter) .....	434
\LWR@hidelatexequation .....	7760	\LWR@listitem .....	5508
\LWR@hspace .....	9027	\LWR@loadafter .....	150
\LWR@HTMLarrayrulecolor .....	5993	\LWR@loadbefore .....	164
\LWR@htmlblockcomment .....	4073	\LWR@loadnever .....	173
\LWR@htmlblocktag .....	4075	\LWR@longtabledatacaptiontag ...	6467
\LWR@HTMLcellcolor .....	5991	\LWR@lookforpackagename .....	524
\LWR@HTMLcline .....	6731	\LWR@LwarpEnd .....	5076, 9191
\LWR@htmlclosecomment .....	4045	\LWR@LwarpStart .....	5004, 9191
\LWR@HTMLcolumncolor .....	5989	\LWR@maketitlesetup .....	33, 5194
\LWR@htmlcomment .....	4066	\LWR@maybenewtablerow .....	5996
\LWR@htmldivclass .....	4100	\LWR@maybeprintpendingfootnotes	4458
\LWR@htmldivclassend .....	4103	LWR@midrulecounter (counter) .....	349
\LWR@HTMLdoublerulesepcolor ....	5995	LWR@midrules .....	348
\LWR@html element .....	4113	LWR@minipagedepth (counter) .....	455
\LWR@html element class .....	4087	LWR@minipagefullwidth (boolean) ..	456
\LWR@html element class line .....	4092	LWR@minipageheight (length) .....	455
\LWR@html element class line .....	4106	\LWR@minipagestartpars .....	9007
\LWR@html element end .....	4116	\LWR@minipagestoppars .....	9010
LWR@htmlfilenumber (counter) .....	241	LWR@minipagethispar (boolean) ....	456
\LWR@htmlfileref .....	6906	LWR@minipagewidth (length) .....	455
\LWR@HTMLhline .....	6714	\LWR@multicolother .....	6353
\LWR@htmlmathlabel .....	7820	\LWR@multicolpartext .....	6348
\LWR@htmlmulticolumn .....	6456	\LWR@multicolskip .....	6359
\LWR@html opencomment .....	4045	\LWR@multirowborder .....	3
\LWR@htmlrefsectionfilename ....	3831	\LWR@mynexttoken .....	5632
\LWR@HTMLrowcolor .....	5990	\LWR@myshorttoc .....	7222
\LWR@HTMLsanitize .....	8113	\LWR@nameref .....	6903
\LWR@HTMLsanitizeexpand .....	8134	LWR@nestspan (environment) .....	3987
\LWR@htmlsectionfilename .....	3798	\LWR@newautoidanchor .....	7116
\LWR@htmlspan .....	4022	\LWR@newautopagelabel .....	4591
\LWR@htmlspanclass .....	4030	\LWR@newhtmlfile .....	4599
\LWR@htmltag .....	4040	\LWR@newlabel .....	6943
\LWR@htmltagc .....	3974	\LWR@newref .....	6994
\LWR@HTMLtextstyle .....	8817	LWR@nextautoid (counter) .....	393
\LWR@imagesizebox .....	95	LWR@nextautopage (counter) .....	393
\LWR@includegraphicsb .....	100	LWR@nextequation (counter) .....	420
\LWR@indexitem .....	7410, 7414, 7418	\LWR@nohspace .....	9055
LWR@infloatrow (boolean) .....	569	\LWR@notmemoirloadafter .....	161
\LWR@instertatbangcols .....	5679	\LWR@nullfonts .....	8887
LWR@intabularmetadata (boolean) ..	325	\LWR@nullifyNoAutoSpacing .....	6733

\LWR@nulllistfills	5515	\LWR@scalestyle	89
\LWR@opacity	94	\LWR@section	4718
\LWR@openparagraph	4215	\LWR@sectionnumber	4698
LWR@opttablecol (boolean)	337	\LWR@setexparray	3750
\LWR@origcolspec	5673	\LWR@setlatestname	6881
\LWR@originname	239	\LWR@setOSWindows	68
\LWR@originnames	246	\LWR@sidetoc	7317
\LWR@overline	8965	\LWR@singledollar	7710
\LWR@parseaftercolumn	5860	LWR@skipatbang (boolean)	324
\LWR@parseatcolumn	5806	LWR@skippingmcolrowcell (boolean)	324
\LWR@parsebangcolumn	5830	LWR@skippingmrowcell (boolean)	324
\LWR@parsebarcolumn	5869	LWR@spandepth (counter)	256
\LWR@parsebeforecolumn	5851	\LWR@splabel	6915
\LWR@parseDcolumn	5896	LWR@starredlongtable (boolean)	362
\LWR@parsedrequirepackagenames	520	LWR@startedrow (boolean)	323
\LWR@parsenormalcolumn	5880	LWR@startingequation (counter)	436
\LWR@parsepcolumn	5892	\LWR@startingequationtag	8167
\LWR@parsetablecols	5902	\LWR@startnewdepth	4708
\LWR@patchlists	5584	\LWR@startpars	4283
LWR@prevFileDepth (counter)	280	\LWR@startref	6956
LWR@previousautopagelabel (counter)	274	\LWR@stoppars	4301
\LWR@printatbang	6042	\LWR@stripperperiod	6880
\LWR@printbartag	6032	\LWR@strresult	5671
\LWR@printcloselist	5415	\LWR@subaddcmidruletrim	6191
\LWR@printlength	502	\LWR@subaddctabularcellcolor	6300
\LWR@printmccoldata	6362	\LWR@subcmidrule	6157
\LWR@printmccoltype	6327	\LWR@subhtmllementclass	4080
\LWR@printopenlist	5416	\LWR@subhyperref	7023
\LWR@printpendingfootnotes	4448	\LWR@subhyperrefclass	7037
\LWR@printpendingmpfootnotes	4467	\LWR@subinlineimage	7066
\LWR@printthetitle	5146	\LWR@sublabel	6920
\LWR@providelength	57	\LWR@subnewref	6988
\LWR@ProvidesPackageDrop	588	\LWR@subsingledollar	7630
\LWR@ProvidesPackagePass	577	\LWR@subtableofcontents	7240
\LWR@pushoneclose	4704	\LWR@subtabularhtmlcolumns	6534
\LWR@quickfile	604	\LWR@syncmathjax	7740
\LWR@remembertag	8172	LWR@tablecolindex (counter)	326
\LWR@requesttoc	5069	\LWR@tablecolspec	5670
\LWR@requirepackagenames	519	LWR@tablecolspecindex (counter)	326
\LWR@restoreorigaccents	915	LWR@tablecolspecwidth (counter)	325
\LWR@restoreorigformatting	7475	\LWR@tabledatacolumnntag	6611
\LWR@restoreoriglists	5612	\LWR@tabledatasinglecolumnntag	6080
\LWR@restoreorigprintxcolor	8476	LWR@tableparcell (boolean)	324
\LWR@rotboxorigin	238	LWR@tabletotalcols (counter)	326
\LWR@rotstyle	86	LWR@tabletotalcolsnext (counter)	326
\LWR@rowHTMLcolor	5986	LWR@tabular (environment)	6746
\LWR@rule	9069	LWR@tabularcelladded (boolean)	324
\LWR@ruleHTMLcolor	5988	LWR@tabularDepth (counter)	325
\LWR@sanitize	3759	\LWR@tabularendoffline	5788
\LWR@sanitized	3758	\LWR@tabularfinishrow	5749

<code>\LWR@tabularhtmlcolumns</code> . . . . .	6544	<code>\markboth</code> . . . . .	3900
<code>\LWR@tabularleftedge</code> . . . . .	6071	<code>\markright</code> . . . . .	3901
<code>LWR@tabularmutemods</code> (boolean) . . . .	369	markup languages . . . . .	59
<code>LWR@tabularpardepth</code> (counter) . . . .	325	math	
<code>\LWR@tdaddstyle</code> . . . . .	6179	mathjax option . . . . .	148
<code>\LWR@tdendstyles</code> . . . . .	6185	mathsvg option . . . . .	148
<code>\LWR@tdstartstyles</code> . . . . .	6178	font size — SVG . . . . .	110, 413, 433
<code>LWR@tempheight</code> (length) . . . . .	474	in section heading . . . . .	75, 98, 138
<code>LWR@tempraise</code> (length) . . . . .	474	labels — valid characters . . . . .	104
<code>LWR@tempwidth</code> (length) . . . . .	474	MathJax summary . . . . .	111, 413
<code>\LWR@textcurrentfont</code> . . . . .	8930	showing as HTML . . . . .	75, 138
<code>LWR@thisautoid</code> (counter) . . . . .	392	SVG summary . . . . .	110, 413
<code>LWR@thisautoidWP</code> (counter) . . . . .	392	word processor conversion . . . . .	130
<code>\LWR@thiscmidrulewidth</code> (length) . . .	349	math (environment) . . . . .	7737
<code>\LWR@thiscolspec</code> . . . . .	6079	MathJax	
<code>\LWR@thisfilename</code> . . . . .	4516	mathjax option . . . . .	148
<code>\LWR@thisnewfilename</code> . . . . .	4517	custom macros . . . . .	112, 414
<code>\LWR@titlingmaketitle</code> . . . . .	76, 5245	subequations . . . . .	111, 414
<code>\LWR@topnavigation</code> . . . . .	3846	summary . . . . .	111, 413
<code>\LWR@traceinfo</code> . . . . .	673	tagged equations . . . . .	111, 413
<code>LWR@tracinglwarp</code> (boolean) . . . . .	169	MathJax (program) . . . . .	111, 413
<code>LWR@trimlrules</code> . . . . .	348	MathJax (program) [requirement] . . . .	62
<code>LWR@trimrrules</code> . . . . .	348	mathjax (boolean) . . . . .	147
<code>LWR@validtablecol</code> (boolean) . . . . .	337	mathjax (option) . . . . .	85, 88, 148
<code>LWR@verbtags</code> (boolean) . . . . .	308	mathsvg (option) . . . . .	85, 88, 148
<code>\LWR@vspace</code> . . . . .	9057	<code>\mbox</code> . . . . .	8708
<code>\LWR@WPcell</code> . . . . .	6238	mcaption (package) . . . . .	606
<code>\LWR@xcolorrowHTMLcolor</code> . . . . .	5984	<code>\mcolrowcell</code> . . . . .	6693
<code>LWRprint@fminipage</code> (environment) . .	8789	mdframed (package) . . . . .	104, 606
<code>\LWRPrintStack</code> . . . . .	3854	<code>\mdseries</code> . . . . .	8939
<code>\LWRsetnextfloat</code> . . . . .	7132	memhfixc (package) . . . . .	617
<code>\LyX</code> . . . . .	9172	memoir	
<b>M</b>			
Mac OS (program) . . . . .	93, 146	options clash . . . . .	122
Madcap (program) . . . . .	59	verse . . . . .	731
<code>\mainmatter</code> . . . . .	4688	memoir (class) . . . . .	121, 122, 731
<code>\makebox</code> . . . . .	8709	meta tag, HTML	
makeidx (package) . . . . .	108, 161	author . . . . .	97, 264
<code>\MakeIndex</code> . . . . .	9164	description . . . . .	96, 264
<code>\makelabel</code> . . . . .	312	generator . . . . .	289
<code>\maketitle</code> . . . . .	40, 95, 5206	Title . . . . .	264
margin		title . . . . .	97, 263
numbers . . . . .	142	viewport . . . . .	290
tags . . . . .	142	metalogo (package) . . . . .	617
marginfit (package) . . . . .	604	microtype (package) . . . . .	84, 158, 617
marginfix (package) . . . . .	605	midfloat (package) . . . . .	618
marginnote (package) . . . . .	605	midpage (package) . . . . .	619
<code>\marginpar</code> . . . . .	100, 270, 4479	<code>\MiKTeX</code> . . . . .	9171
<code>\marginparBlock</code> . . .	100, 270, 4490, 4507	minipage	
		framed . . . . .	101
		horizontal space between . . . . .	473



acro	488	cuted	518
acronym	489	cutwin	518
addlines	492	dblfloatfix	519
adjmulticol	491	dblfnote	519
afterpackage	159	dcolumn	520
afterpage	492	diagbox	520
algorithmicx	492	draftwatermark	522
algorithmx	120	easy-todo	522
alltt	494	ebook	523
amsmath	161	ellipsis	523
amsthm	494	emptypage	524
anonchap	498	endfloat	524
anysize	498	endheads	524
appendix	107, 499	endnotes	107, 525
arabicfront	499	enumerate	526
array	500	enumitem	526
atbegshi	500	environ	161
authblk	107, 501	epigraph	527
axodraw2	502	epstopdf	113, 115, 446, 528
babel	123	eso-pic	529
backref	502	etoolbox	145
balance	502	everyhook	160
bigdelim	119, 503	everypage	529
bigstrut	504	everyshi	530
blowup	505	expl3	160
bookmark	505	extramarks	530
booktabs	505	fancybox	102, 531
boxedminipage	506	fancyhdr	535
boxedminipage2e	506	fancyheadings	535
breakurl	507	fancyref	536
bytefield	507	fancyvrb	537
calc	161	figcaps	542
cancel	508	figsize	543
capt-of	396	filecontents	160
caption	120, 166, 396, 509	fix2col	543
caption2	512	fixme	124, 543
ccaption	512	fixmetodonotes	545
changebar	512	flafter	545
changepage	513	float	120, 546
chappg	514	floatflt	547
chapterbib	514	floatpag	548
chnpage	513	floatrow	120, 548
cite	514	fltrace	554
cleveref	105, 451	flushend	554
cmap	84	fncychap	555
color	114, 515	fnlineno	555
colortbl	515	fnpos	556
comment	150	fontenc	84, 556
continue	517	fontspec	83, 556
crop	517	footmisc	557

footnote	558	mcaption	606
footnotehyper	559	mdframed	104, 606
footnpag	559	memhfixc	617
framed	559	metalogo	617
ftnright	562	microtype	84, 158, 617
fullpage	562	midfloat	618
fullwidth	563	midpage	619
fwlw	563	morefloats	619
geometry	159, 563	moreverb	619
getttitlestring	160	morewrites	621
glossaries	107, 564	mparhack	621
graphics	113, 565	multicol	621
graphicx	113, 576	multirow	623
grffile	114, 577	multitoc	626
hang	577	nameref	626
hanging	579	natbib	626
hypcap	580	needspace	627
hypdestopt	580	newclude	123, 628
hypernat	580	newfloat	161
hyperref	105, 387, 581	newtxmath	113
hyperxmp	585	newunicodechar	84, 628
hyphenat	585	nextpage	628
idxlayout	586	nicefrac	113
ifoddpage	587	nonfloat	629
ifplatform	145	nonumonpart	629
indentfirst	587	nopageno	629
inputenc	84, 588	nowidow	630
keyfloat	121, 588	ntheorem	112, 630
kvoptions	147	overpic	115, 642
layout	590	pagegrid	643
letltxmacro	145	pagenote	107, 643
letterspace	591	pagesel	644
lettrine	591	paralist	644
lineno	592	parskip	645
lips	595	pbox	645
listings	595	pdfscape	645
lmodern	83	pdfrender	646
longtable	118, 599	pdfsync	646
lscape	601	pfnote	646
ltcaption	601	phfqit	647
ltxtable	602	placeins	647
luacolor	602	prelim2e	648
luatodonotes	123, 602	prettyref	648
lwrap	85	preview	648
lwrap-patch-komascript	756	printlen	162
lwrap-patch-memoir	758	quotchap	649
makeidx	108, 161	ragged2e	650
marginfit	604	realscripts	651
marginfix	605	refcount	161
marginnote	605	relsize	99, 652

resizegather	653	tocloft	107, 715
romanbar	653	tocstyle	721
romanbarpagenumber	654	todo	722
rotating	654	todonotes	123, 723
rotfloat	654	transparent	725
savetrees	655	trivfloat	120, 725
scaleft	656	turnthepage	726
schemata	656	typearea	727
scrextend	657	ulem	727
scrhack	660	units	113
scrlayer	660	upref	730
scrlayer-notecolumn	662	url	105
scrlayer-scrpage	662	varioref	105
section	663	varwidth	162
sectionbreak	664	verse	122, 730, 731
sectsty	665	vertbars	732
setspace	665	vmargin	732
shadow	667	vwcol	733
showidx	667	wallpaper	735
showkeys	667	watermark	736
sidecap	668	wrapfig	736
sidenotes	668	xcolor	114, 447, 738
siunitx	112, 444, 670	xfrac	748
soul	675	xifthen	160
soulpos	677	xltextra	751
soulutf8	678	xmpincl	752
stabular	678	xparse	124, 159
stfloats	679	xpatch	145
subcaption	120	xstring	161
subfig	120, 679	xtab	119, 752
subfigure	684	xurl	754
supertabular	119, 685	xy	754
syntonly	686	zref	161
tabls	687	zwpagelayout	755
tabularx	687	page	
tabulary	688	inaccessible	88, 89
textarea	688	\pagebreak	9061
textcomp	84, 99, 689	\pagecolor	69
textfit	692	pagegrid (package)	643
textpos	693	pagenote (package)	107, 643
theorem	694	\pagenumbering	3906
threeparttable	698	\pageref	7001
tikz	699	\pagerefPageFor	7000
titleps	700	pagesel (package)	644
titleref	703	\pagestyle	3898
titlesec	704	Pandoc (program)	59
titletoc	705	\par	
titling	107, 707	hooks	260
tocbasic	712	\paragraph	4929
tocbibind	108, 109, 713	paralist (package)	644





realscripts (package) .....	651	section (package) .....	663
\ref .....	6994	sectionbreak (package) .....	664
refcount (package) .....	161	sectsty (package) .....	665
\reflectbox .....	292	\SetHTMLFileNumber .....	3789
relsize (package) .....	99, 652	setspace (package) .....	665
[requirement]:		settings	
LuaLaTeX (program) .....	62	css project-specific .....	93
MathJax (program) .....	62	CSS selection .....	92
pdfcrop (program) .....	62	HTML conversion .....	87
pdfLaTeX (program) .....	62	lwrap package options .....	85
pdfseparate (program) .....	62, 67	selecting output .....	94
pdftocairo (program) .....	62, 67	title page .....	95
pdftotext (program) .....	62, 67	\sffamily .....	8942
perl (program) .....	68	\sfrac .....	748
XeLaTeX (program) .....	62	shadow (package) .....	667
\RequirePackage .....	552	showidx (package) .....	667
\resizebox .....	297	showkeys (package) .....	667
resizegather (package) .....	653	sidecap (package) .....	668
\ResumeTabular .....	6598	sidenotes (package) .....	668
\reversemarginpar .....	4503	SideTOCDepth (counter) .....	88, 401
\rightline .....	8345	\sidetocname .....	7314
\rmfamily .....	8941	\simplechapterdelim .....	4715
romanbar (package) .....	653	siunitx	
romanbarpagenumber (package) .....	654	with TeXMaths .....	130
\rotatebox .....	258	siunitx (package) .....	112, 444, 670
rotating (package) .....	654	\sloppy .....	3904
rotfloat (package) .....	654	soul (package) .....	675
\rowcolor .....	5990	soulpos (package) .....	677
\rownum .....	5981	soulutf8 (package) .....	678
\rule .....	9117	\sp .....	8948
<b>S</b>			
sample_project.css (file) .....	93, 217	space	
savetrees (package) .....	655	horizontal .....	98
\sb .....	8949	between minipages .....	473
\scalebox .....	277	split	
scaleft (package) .....	656	miss-numbered .....	112, 630
schemata (package) .....	656	stabular (package) .....	678
scrextend (package) .....	657	stack depths .....	140
scrhack (package) .....	660	\StartDefiningTabulars .....	5727
scrlayer (package) .....	660	stfloats (package) .....	679
scrlayer-notecolumn (package) .....	662	subcaption (package) .....	120
scrlayer-scrpage (package) .....	662	subequations	
\scshape .....	8946	MathJax .....	111, 414
section		subfig (package) .....	120, 679
depths .....	140	subfigure (package) .....	684
heading, word processor .....	130	\subparagraph .....	4937
math in heading .....	75, 98, 138	\subsection .....	4912
missing .....	88, 89	\subsubsection .....	4920
\section .....	4899	subtitle .....	304
		supertabular (package) .....	119, 685





xstring (package) .....	161	
xtab (package) .....	119, 752	
xurl (package) .....	754	
xy (package) .....	754	
		<b>Z</b>
		zref (package) .....
		161
		zwpagelayout (package) .....
		755